NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF A PROTOTYPE DETAILING MANAGEMENT SYSTEM FOR THE CIVIL ENGINEER CORPS

by

Brian Louis Weinstein

September 2002

Thesis Advisor:	Magdi N. Kamel
Co-Advisor:	Dale M. Courtney

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Development of a Prototype Detailing Management System for the Civil Engineer Corps	**5. FUNDING NUMBERS**
6. AUTHOR(S) Brian Louis Weinstein	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

The Civil Engineer Corps (CEC) is a small group of naval officers totaling approximately 1250 officers. Being a small community, the CEC attempts to capture the location and other relevant information of every CEC Officer at a given time and publish that information in the Chief of Civil Engineers Directory (P-1). The CEC detailers also produce and work off a Staffing Plan to help them manage the movement of officers. The current process of developing the Staffing Plan and P-1 is complex and arduous.

This thesis examines existing systems at the CEC detail shop and explores the possibilities of developing a Detailing Management System. The intent is to develop a prototype application that links with existing systems to assist the detailers in the execution of their tasks including the streamlined production of the P-1 and Staffing Plan.

A prototyping methodology is used to accomplish these goals. The methodology includes the following phases: Problem Definition, Requirements Analysis, Design, Build/Revise prototype, and Finalize application. The build phase is iterated through several end-user reviews to obtain the best possible product to support their needs.

The application supports the immediate requirements of streamlining the development of the Staffing Plan and P-1 while providing additional detailing management capabilities.

14. SUBJECT TERMS Civil Engineer Corps, Detailing, Personnel Management, Database, Legacy Systems, Integration	**15. NUMBER OF PAGES** 187
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

DEVELOPMENT OF A PROTOTYPE DETAILING MANAGEMENT SYSTEM FOR THE CIVIL ENGINEER CORPS

Brian L. Weinstein
Lieutenant, United States Navy
B.S., University of Colorado - Boulder, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2002

Author: Brian Louis Weinstein

Approved by: Magdi N. Kamel
Thesis Advisor

Dale M. Courtney
Co-Advisor

Dan C. Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Civil Engineer Corps (CEC) is a small group of naval officers totaling approximately 1250 officers. Being a small community, the CEC attempts to capture the location and other relevant information of every CEC Officer at a given time and publish that information in the Chief of Civil Engineers Directory (P-1). The CEC detailers also produce and work off a Staffing Plan to help them manage the movement of officers. The current process of developing the Staffing Plan and P-1 is complex and arduous.

This thesis examines existing systems at the CEC detail shop and explores the possibilities of developing a Detailing Management System. The intent is to develop a prototype application that links with existing systems to assist the detailers in the execution of their tasks including the streamlined production of the P-1 and Staffing Plan.

A prototyping methodology is used to accomplish these goals. The methodology includes the following phases: Problem Definition, Requirements Analysis, Design, Build/Revise prototype, and Finalize application. The build phase is iterated through several end-user reviews to obtain the best possible product to support their needs.

The completion of this thesis demonstrates that the concept of a detailing management program that meets the needs of the users is valid and must be addressed in future development of production systems. The application supports the immediate requirements of streamlining the development of the Staffing Plan and P-1 while providing additional detailing management capabilities.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS

CEC	Civil Engineer Corps
P-1	Chief of Civil Engineers Directory
PRD	Projected Rotation Date
PEP	Personnel Exchange Program
BSC	Billet Sequence Code
PERS 4413	Civil Engineer Corps Detail Shop
GUI	Graphical User Interface
NAVPERSCOM	Navy Personnel Command
NMCI	Navy/Marine Corps Intranet
PE	Professional Engineer
EIT	Engineer in Training
SIHRS	Single Integrated Human Resource Strategy
DIMHRS	Defense Integrated Military Human Resources System
OAIS	Officer Assignment Information System
OPINS	Officer Personnel Information System
ODIS	On-Line Distribution System
ASCII	American Standard Code for Information Interchange
EMPRS	Electronic Military Personnel Record System
VBA	Visual Basic for Applications
SDLC	System Development Life Cycle
NPDB	Navy Personnel Database
NAVFAC	Naval Facilities Engineering Command
NITC	NAVFAC Information Technology Center
BUPERS	Bureau of Naval Personnel
UIC	Unit Identification Code
SCW	Seabee Combat Warfare
SWO	Surface Warfare Officer

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my terrific wife who supported and advised me through the entire process. I could not have completed this without her support.

I would also like to thank LT Neil Rader for his tireless assistance and persistence in the many projects and coursework during my stay at the Naval Postgraduate School. Finally, I would also like to thank my advisors, Dr. Magdi Kamel and Dale Courtney, for their guidance in the development of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION & BACKGROUND

A. AREA OF RESEARCH

The Civil Engineer Corps (CEC) is small group of naval officers accounting for about 2.3% of the total officer strength in the United States Navy, or approximately 1250 officers. CEC duty assignments are generally focused in three main areas: Public Works, Contracting, or Construction Battalions (Seabees). These duty stations can be located anywhere in the world. Being a small community, the CEC attempts to capture the location of every CEC Officer at a given time and publish that information in the Chief of Civil Engineers Directory (P-1). The P-1 contains information on where all officers in the CEC are located, what their billet position is, and when they reported to that particular duty station. It also has the address and phone numbers of each command (activity) where the CEC has personnel stationed. Many CEC Officers use the P-1 to keep in contact with their colleagues and to search for openings for their next duty station, based on their projected rotation date (PRD). This publication is printed annually and approximately 2000 copies are distributed throughout the CEC at a cost of around $12,000. The current process of developing the P-1 is a complex and arduous one, which is described in Chapter 2.

A new, user-friendly relational database system will greatly facilitate the production of the P-1, Staffing Plan, and assist the CEC detailers in the accomplishment of their duties by giving them ready access to information displayed in a logical manner. A system which is either linked to the legacy data application or routinely imports the data via an automated process will decrease the burden on the staff to produce the P-1 and Staffing Plan and allow them to focus on their primary mission which is to place CEC officers in billets matching the needs of the navy, individual, and the individual's career.

B. RESEARCH ISSUES

The main objective of this thesis is to examine the possibility of integrating a new, user-friendly system with the current legacy personnel system. The research focuses on developing an appropriate data model for the prototype, taking into account the current systems and the requirements of the CEC Detail Shop (PERS 4413). Essential to the process of developing the data model is the determination of which development strategy to use: top-down or bottom-up. (Kroenke, 2000 pp. 41-42) Top-down development is a global approach to system development. It starts with a study of an organization's strategic goals, the methods by which those goals can be accomplished, the information that is required to meet these goals, and the systems necessary to provide that information. A bottom-up approach works in the reverse order. It begins with the need for a specific system and builds up from there. Requirements are drawn up based on existing systems and reports as well as end-user input to form the basis for the design of the project. Bottom-up development was chosen because it is faster and smaller in scope than a top-down approach, which can many times lead to analysis paralysis.

Another critical issue to consider is which design methodology should be used for the system. Quick development and limited initial interaction with the end users of the system play a key role in the choice of design methodology. The methodologies considered were a classic waterfall approach, a prototype approach, and a spiral approach. Prototype approach was used as it is utilizes a complete requirements analysis up front and iterative design and build phases.

A final issue that is critical to the thesis is the integration of the legacy system into the new model. The issue regarding how the new system will interact with the legacy system is considered. Options include linking directly to the system, a one-time migration of the data from the old system to the new system, or an import routine that is done each time the user wants to update the new system. The legacy system situation at PERS 4413 is very dynamic and the solution needs to be flexible to meet the users' needs for the future. A standalone system that is maintained from a one-time import could be ideal if the users are diligent in maintaining the information in parallel systems. A more in-depth analysis and discussion on this issue is discussed in Chapter IV.

C. SCOPE

The primary objective of this thesis is to provide PERS 4413 with a user-friendly management tool that will link with or import from the legacy personnel system to assist in the production of the P-1 and Staffing Plan along with other detailing functions. To accomplish this goal, the work will be split up into two parts. The first part is the development of the relational data model and database application that captures all the entities currently maintained in the PERS 4413 legacy systems that relate to the CEC personnel management and the production of the P-1. Information contained in the database will include, but not be limited to, personnel data, billet information, and activity information. The remaining portion of the thesis will focus on how the new system will interact with the legacy application and its data. The ease of data transfer and integration for the user is a key part of the success of the project. PERS 4413 intends to use the new application primarily for the production of the P-1 and Staffing Plan. Any additional functionality added to prototype the Detailing Management System is for academic purposes and is intended to demonstrate the increased capability of a relational database application vice legacy stovepipe systems. This thesis will not incorporate the use of web technologies to manipulate or display information. The data maintained in this project contains sensitive Privacy Act information. Therefore, security is built into the application to allow only authorized personnel access to the information.

D. METHODOLOGY

Much of the success of any information systems project depends on the appropriate selection and execution of a development methodology. In order to select a proper methodology, a review is provided below for three popular development methodologies. A methodology is an organized, documented set of procedures and guidelines for one or more phases of the systems development life cycle (SDLC) that may or may not incorporate the use of several development tools and techniques. The SDLC is a logical process by which developers and end-users build information systems to solve business problems or address specific needs. (Whitten, 1998, pp. 72-73) The three methodologies that were considered are the Waterfall, Prototype, and Spiral approach. These methodologies are by no means an all-inclusive list of possibilities, but are three of the most popular in the industry today. Each one of these methodologies will be evaluated on their strengths, weaknesses, and implied risks. For this thesis, I chose to follow the prototyping methodology. A more detailed explanation of each of these methodologies and the reason for selecting the prototype is located in Chapter II.

E. ORGANIZATION

This thesis will be organized in the following manner:

- Chapter II further defines the problem, identifies stakeholders, defines the scope, and examines the feasibility of the project.
- Chapter III, the requirements analysis, is the natural follow on to the problem definition and defines what the end-users require to perform their job. The data requirements and application requirements are examined thoroughly. Any required reports or existing reports being used by PERS 4413 will be collected and analyzed to complete the model definition.
- Chapter IV gives an overview of standard design philosophies adhered to in this thesis. Database design issues as well as application design issues are discussed in depth.
- Chapter V discusses the actual development of the application with specific attention to structure and the graphical user interface (GUI). Prototyping is discussed as well as the training and implementation plans.
- Chapter VI addresses what to do after implementation. It discusses how the system is to be maintained, including both data and application maintenance as well as scalability. Human resource requirements and ongoing training are also addressed.
- Chapter VII summarizes the project and reviews lessons learned, in addition to recommending areas of future study.

F. BENEFITS

One of the Navy's greatest resources is its personnel. To maximize the effectiveness of the CEC community, the CEC detailers need an efficient, user-friendly tool to help plan and program personnel movements. This research will have an immediate impact on the PERS 4413 personnel by enabling them to quickly access information that was previously inaccessible without a tremendous effort. It also provides insight on the issues faced when dealing with legacy stovepipe applications, helping the Navy to perhaps avoid the same mistakes in the future.

II. INITIATION & PLANNING

A. METHODOLOGY SELECTION

As discussed in Chapter I, methodology selection can play a very important role into how a project will evolve and ultimately how much success it will have. The three methodologies considered for this thesis are described below.

1. Waterfall Model

The waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development, that once completed are not revisited. In other words, once a phase of development is completed, the development proceeds to the next phase and there is no turning back. A graphical representation of the waterfall approach can be seen in Figure 1.

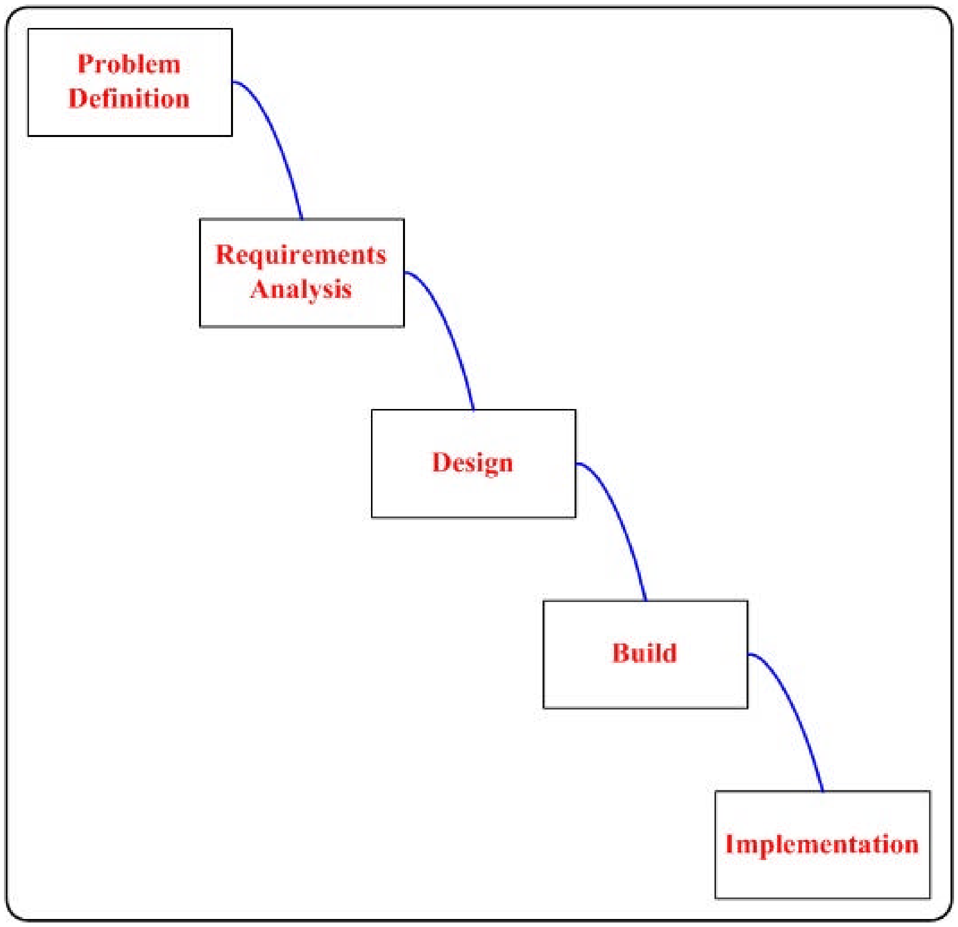


Figure 1. Waterfall Methodology

a. ***Strengths***

- Each step is verified and validated prior to moving to the next.
- Clear definition of requirements up front help to focus the project on necessary elements.

b. ***Weaknesses***

- Changes in requirements during development can be very costly.
- There is no working product prior to completion; therefore, problems in any of the phases will not surface until the end of the program.
- It does not allow for much reflection or revision.

c. ***Implied Risks***

- If the requirements are not well defined, the project could be in jeopardy.
- Changes during development could greatly increase costs and time.

2. Prototyping Model

The prototyping model is a methodology in which a prototype is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users. A graphical representation of the prototype methodology can be seen in Figure 2.

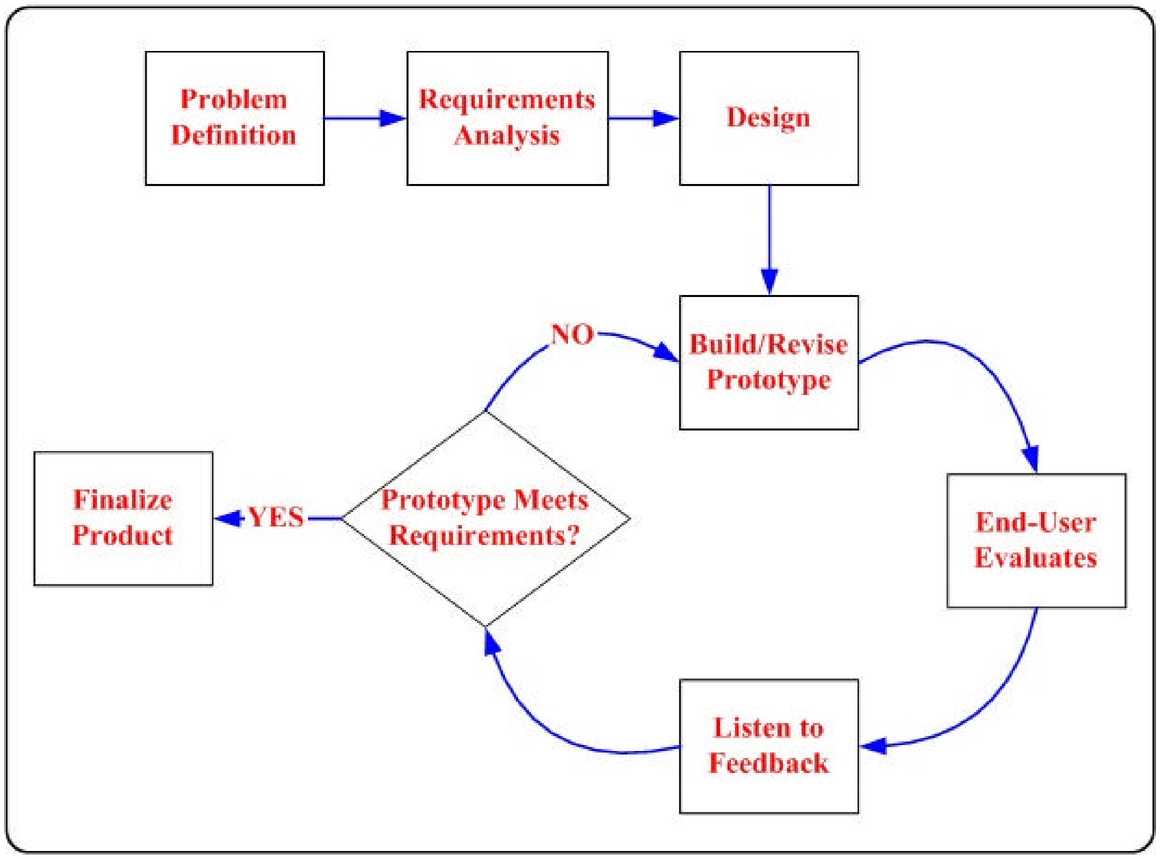


Figure 2. Prototype Methodology

a. ***Strengths***

- Gives early capability in the project.
- Build and test as you go – allows the team to test the product modules with existing infrastructure and make changes as you go.
- Encourages and requires active end-user participation.
- Gives the end-users an opportunity to provide direction for the application design, which provides them with a sense of ownership.
- Users will be more responsive and helpful if they have something to touch and feel as opposed to a diagram to look at.
- Can increase creativity because it allows for quicker user feedback, which can lead to better solutions.

b. ***Weaknesses***

- If the requirements are not properly defined, the project timeline could be delayed and costs could increase.
- Product modules could be split up incorrectly based on invalid assumptions and assessments.
- Can encourage ill-advised shortcuts through the life cycle.

- Often leads to a premature commitment to a design.
- Susceptible to "feature creep" where the scope and complexity of the system can expand beyond original plans quickly.
- Can also reduce creativity in certain areas. A user and designer who view a prototype report can become entrenched in that design for the report, unable to see other options.

c. Implied Risks

- Too much testing can slow the project down. Clear testing milestones need to be set.
- All functions and requirements must be defined and understood up front to prevent a never-ending cycle.

3. Spiral Model

The spiral model is a development methodology which combines the features of the prototype model and the waterfall model. The spiral model is generally favored for large, expensive, and complicated projects. A graphical representation of the spiral methodology can be seen in Figure 3.

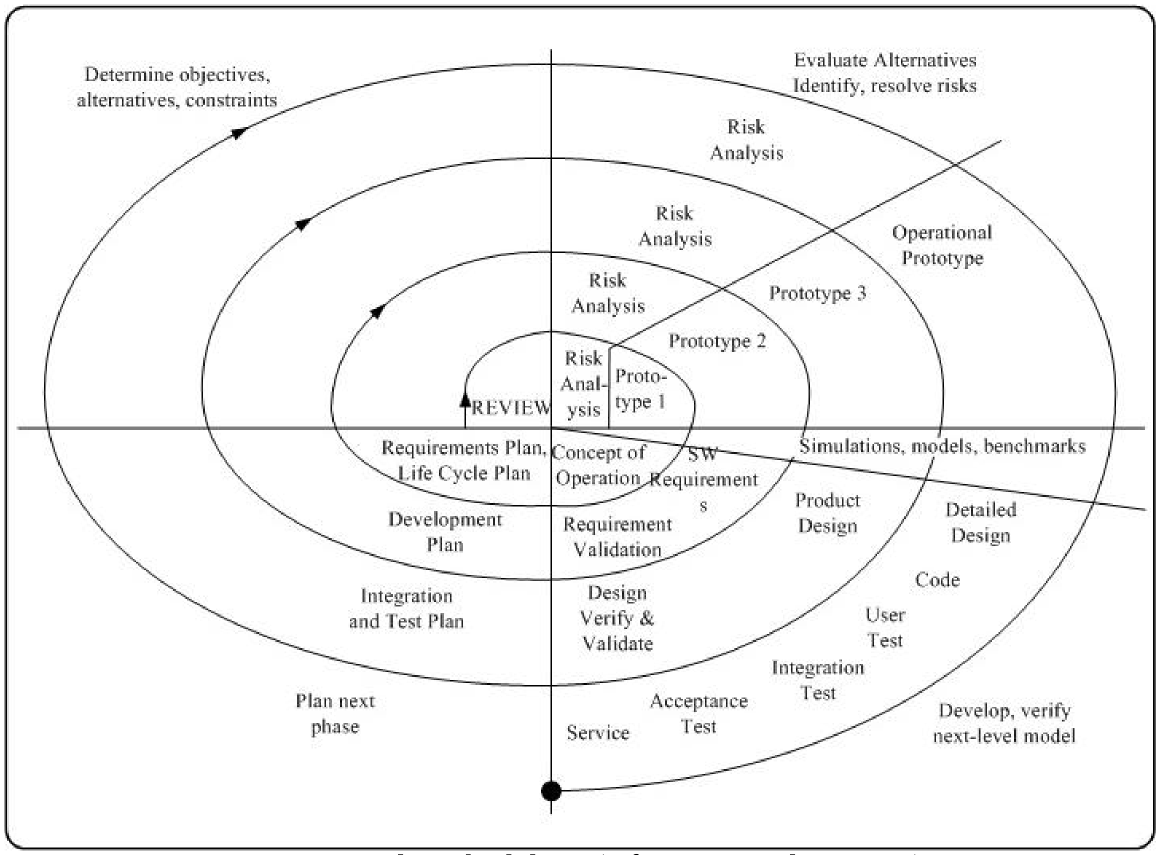


Figure 3. Spiral Methodology (After: Osmundson 2002)

a. ***Strengths***

- Brings early capability to the project.
- Any unforeseen or additional requirements are more easily and cheaply incorporated.

b. ***Weaknesses***

- Difficult to set and stick to a budget when requirements and scope are continually developed as the project progresses.
- When does the spiral end? This type of project has the potential for continual development with no clear end.

c. ***Implied Risks***

- Involves gathering many individuals together at multiple times. This alone could be a major obstacle.
- Obtaining user feedback (not upper management) is vital to the success of the project.

In addition to the methodologies described above, there are numerous others that are being used in all facets of information technology development. Some methodologies are a blend of others and new ones are created continually because every project has specific needs that one particular methodology may not meet.

For this thesis, I chose to utilize the prototyping approach for several reasons:

- The requirements analysis is done in one visit, saving time and travel costs
- Getting all the requirements in one trip is unlikely – multiple prototypes will help flush out the remaining requirements.
- The build phase is iterated and reviewed by the end-user to achieve the best possible product.

B. IDENTIFY STAKEHOLDERS

Seeking out and identifying key stakeholders in any project is critical to its success. Stakeholders can be anyone that has a vested interest in the existing systems being analyzed, the new system being proposed, or anyone who controls the purse strings of the budget that will be used for the project. These stakeholders can range from the Commanding Officer who wants to see an improved system for any number of reasons to the end-user who interacts with the system on a daily basis. Other stakeholders could include the information systems support staff, chief financial officers, or any other interested party that may have influence on the project.

LT Neil Rader and I conducted an initial discovery trip in June 2001 for the purpose of reviewing existing systems, identifying stakeholders, and determining feasibility. At the time, the scope of the project was envisioned to be a much larger enterprise that would have included full internet capability along with the database application. A complete copy of the trip report for this visit can be found in Appendix B. During this trip, several stakeholders were identified and interviewed, producing the following information.

1. CEC Detailers

a. PERS 4413 Head Detailer/Community Manager

The head detailer is responsible for the overall management of the CEC community and specifically handles all the personnel detailing for those officers who hold the rank of Captain and above. He also is the final approving authority for budget spending on this project. He was not available for specific interviews during the discovery trip as it was his first week in that position, but did endorse the conceptual idea of the project to a certain extent. His main concern was that the project would result in something easier for the Community Management Analyst to use in the preparation of the P-1 and Staffing Plan. Little to no interest was expressed in a detailing management tool for the rest of the detailers to use in addition to the other systems they were using at the time.

b. PERS 4413A LCDR Detailer

The LCDR detailer is typically a CDR and handles all the detailing needs for CEC personnel that hold the rank of LCDR. He works closely with the head detailer and also with the LT detailer to fill any billet gaps with junior or senior officers. During his interview, he expressed that his main concern was streamlining the production of the P-1 and Staffing Plan. Anything beyond that he was not concerned with at all. He routinely pulled data down from the Officer Data Information System (ODIS) to manage the personnel under his responsibility. He would download the data and import it into an MS Excel spreadsheet and manipulate it from there (see example in Appendix C). He would sort the spreadsheet by rotation date to see who was due to change duty stations and then added about 7 fields that he used to plug names in of the people he was thinking about placing in that billet. He would then juggle these names around to find the best fit based on balancing the needs of the Navy, career needs of the individual, and desires of the individual. He would also sort these lists in other manners such as from the senior LCDR to most junior LCDR in the promotion zone to get an idea on whether he should wait to issue orders depending on if the officer was selected for promotion or not. He also examined the billet list sorted by type of position. In addition to these spreadsheets, he also logged every phone conversation and email correspondence that he had with a member using MS Outlook's Journal feature.

c. PERS 4413E LT/CWO/Grad School Detailer

The LT detailer is also in charge of sending CEC personnel to graduate school and detailing the Chief Warrant Officers to their respective billets. Similar to the LCDR Detailer, he works closely with the detailers above and below him to fill any billet gaps. During his interview, he was most receptive to a tool that would consolidate all the information that he spends hours collating just to get a simple report. He was most interested in being able to sort a list of personnel by a multitude of factors such as year group, rank, promotion status, etc. Another useful tool that he would like to see is the capability to show only those billets that require a certain qualification, such as Seabee Combat Warfare pin. Other nice to have reports were a duty history for each member, a billet history, and the ability to filter billets by type such as Public Works, ROICC, staff, or Seabees, etc. He did most of his detailing off the Staffing Plan, but also used a spreadsheet system similar that was slightly different from the LCDR detailer. His spreadsheet (example shown in Appendix C) contained information on the member and could be sorted in various manners. He also added several fields on the end for his own use in placing the member's preferences and any other additional comments he may have on the particular member.

d. PERS 4413C LTJG/ENS Detailer

The LTJG/ENS detailer is typically a LT and handles all the assignments for the junior officers of the CEC. He is responsible for each officer's first duty assignment in the CEC and also handles the Naval Mobile Construction Battalion (NMCB) detailing. The interview with the LTJG/ENS detailer revealed that in this office, like most other offices in the military, the junior man gets to do a lot of the grunt work for the other senior members of the office. In addition to performing his own data pulls from the system, he also did data pulls for the other detailers as well. Again, like the other detailers he used a spreadsheet to keep track of the personnel and billets under his responsibility. His spreadsheet was similar to the others with respect to the data elements contained (see Appendix C), but his method of using it was slightly different. He relied mostly on color-coding of information. Purple rows were for students at the Civil Engineer Corps Officer School (CECOS), red rows indicated proposed orders, orange were approved, awaiting orders, and blue were reserved for those personnel to whom he had guaranteed a particular duty station. Additionally, he kept a paper log of phone conversations that he had with any service members. He also filled in when needed to write orders when the Community Management Analyst was not available to do them.

e. PERS 4413S CEC Community Management Analyst

The CEC Community Management Analyst is the heart and soul of the organization. He is a civilian worker that provides continuity amongst the constant rotation of officers in the detail shop. He is also the information hub where all the corporate knowledge resides. He provides assistance to the detailers regarding all aspects of the business and is the resident expert in the use of the information systems applications provided for the detailers. He is responsible for the production of the P-1 and the Staffing Plan and the only one who knows how to use the current system to produce the plans. He is also the chief order writer for the detailers. During this discovery trip, most of our time was spent with him as he walked us through the process of creating the P-1 and Staffing Plan. As mentioned in Chapter I, this process is a long and complex one that involves numerous steps and requires considerable corporate knowledge of the CEC community. The following steps are a synopsis of the current process for producing the P-1 and the Staffing Plan:

- Two separate downloads of delimited text files for personnel (bodies) and job positions (billets). See Appendix D for a sample of these files.
- Each of these files is then imported into a Microsoft Excel spreadsheet. See Appendix E for a sample of these spreadsheets.
- The spreadsheet is examined and manipulated by hand to correct for any data anomalies that include, but are not limited to:

- Personnel that do not belong in the database, but have been incorrectly identified as belonging to the CEC.
- Reserve billets that are not included in the P-1.
- PEP billets that are not used.
- Billet sequence code (BSC) errors.
- Prioritization of personnel qualifications.
- Modification of billet titles and rank descriptions.
- Once all modifications have been made, the Excel file is imported into a Microsoft Access database for further validation and manipulation. The database checks for:
- Billets without any matching activities
- Duplicate names in the personnel report
- Missing names in the personnel report
- Billets not showing up in the billets report
- Once all of these validations are complete, the P-1 is ready to be generated. A button is pushed and approximately 100 macros are sent into action performing magic behind the curtain to produce the P-1.
- The entire process takes about 8 hours to accomplish assuming there are not any data problems or computer crashes.

After interviewing all of the detailers, it became clear that each detailer operated using the same data, but in a slightly different matter. There seem to be similar business models for each detailer, but each varies slightly from the other. Further analysis would be required to standardize the detailers in a business process that works for all of them. Once that is done, the application could be enhanced to meet their needs.

2. NAVPERSCOM Information Systems Personnel

NAVPERSCOM is currently undertaking many IT related projects including, but not limited to, NMCI, legacy systems upgrades, and an upgrade to the EMPRS system. There is a local development project underway (Single Integrated Human Resource Strategy - SIHRS) which is supposed to combine all the stovepipe systems into one relational model for use by NAVPERSCOM. The systems described above are all planned to be included in this local project along with other systems not mentioned. There is currently no timeframe for the execution of this project and it is understood to be just in the conceptual phase, which leads to the estimation that the successful completion of this project is at least 5 years away. There is also a project underway at NAVPERSCOM to try to make these management systems entirely web based. DOD is also examining the possibility of developing the Defense Integrated Human Resource System (DIMHRS), which is intended to be a personnel management application for the entire military. It is not known whether these two projects will be integrated, superceded, or separated. Given the number and magnitude of the projects that NAVPERSCOM is undertaking, they expressed great concern in allowing another system to be created to support PERS 4413 outside the realm of their oversight.

C. REVIEW OF CURRENT SYSTEMS

1. Database Systems Background

Prior to discussing the current systems, it is important to note the differences between file-processing systems and database systems. File processing systems predated modern relational database systems. While they are a great improvement over manual record-keeping systems, they have several disadvantages (Kroenke, pp 11-13):

- Data are separated and isolated. For example, in the case of the NAVPERSCOM systems there are two files. One file is for billets and the other is for bodies. These two files are completely separate from each other and it takes programming or manual manipulation to figure out what person (body) is attached to what billet.
- Data are often duplicated. In this same case, both the billets and the bodies files contain the same information on UIC, BSC, Billet Rank, etc. In fact, matching the UIC and BSC for each record in both the files is the only way to figure out who is attached to what billet.
- Application programs are dependent on file formats. Most file-processing systems are programmed to access the data in a certain format. If the format changes for one system, then all other systems using that file must be reprogrammed to match that change. For example, if the BSC field in the billets file were to change from five digits to seven digits, all programs using that file would have to be changed to match the new format.
- Files are often incompatible with one another. The billets file comes from OAIS and the bodies file comes from OPINS. The OAIS application cannot read the bodies file and the OPINS application cannot read the billets file. A separate application (ODIS) was created to combine the two files and present the information in one application. (Kroenke, pp 11-13)

Relational database systems were developed to overcome the limitations of file-processing systems. Databases are essentially a collection of interrelated files. The records in each file must allow for relationships to the records in other files. These systems have great advantages over file-processing systems:

- Data are integrated. All the application data is stored in a single database. Each table (or file) in the database is uniquely identified by a field called the primary key. Tables are related to each other by placing one table's primary key into another table. This key is called a foreign key.
- Reduced data duplication. In the example given above, instead of the billet information being stored with both the person and the billet, the UIC and BSC would be stored only in with the billet. Because data are stored in only one place, data integrity problems are less common.
- Program/Data Independence – Database processing reduces the dependency of programs on file formats. The formats are stored along with the data inside the database and are accessed by the Database Management System (DBMS), not the application program. (Kroenke, pp 14-15)

The main criticism of relational databases has traditionally been performance. Relational databases require more computer processing power than its predecessors. This issue has virtually disappeared with the great advancements in processing power of the modern computer.

2. Current Systems at NAVPERSCOM

Based on the initial discovery trip, a basic understanding of the existing systems being used was achieved. The diagram below in Figure 4 shows the current system architecture.

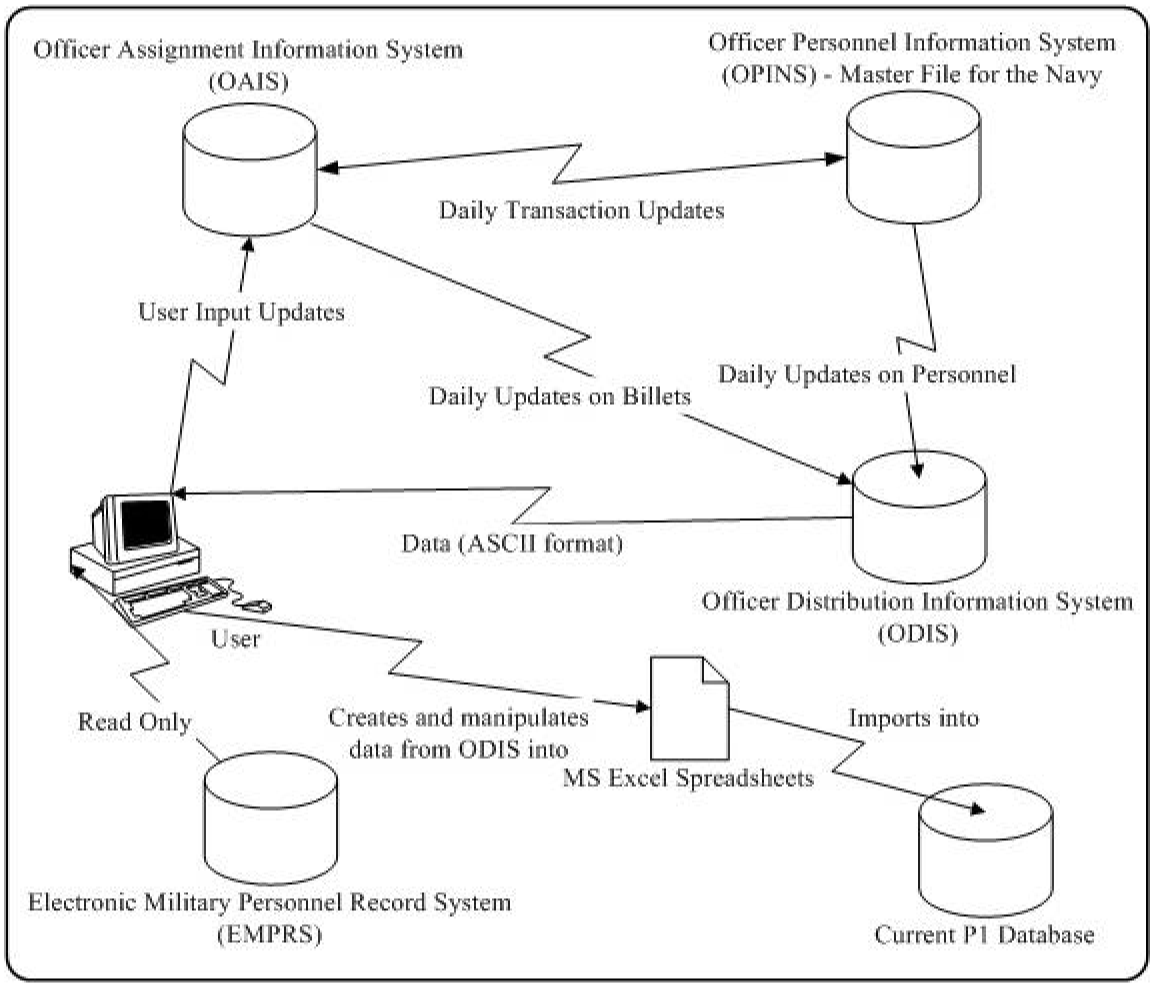

Officer Assignment Information System (OAIS)
Officer Personnel Information System (OPINS) - Master File for the Navy
Daily Transaction Updates
User Input Updates
Daily Updates on Billets
Daily Updates on Personnel
Data (ASCII format)
Officer Distribution Information System (ODIS)
User
Read Only
Creates and manipulates data from ODIS into
MS Excel Spreadsheets
Imports into
Electronic Military Personnel Record System (EMPRS)
Current P1 Database

Figure 4. Current System Architecture

The master file for the Navy (OPINS) is maintained on a large mainframe located in Cleveland, OH. The locally maintained mainframe (OAIS) handles the day-to-day operations for personnel management. The personnel in the detail shop make any changes concerning personnel or billets in this system. OAIS and OPINS perform nightly transaction updates to synchronize the information. When a staff member needs to extract information to create reports or spreadsheets to perform their duties, this data is taken from the On-Line Distribution System (ODIS) in ASCII format. The ASCII data is downloaded in a delimited format and imported into Microsoft Excel for manipulation and reporting. After manipulation in Excel, the data is then imported into the current P-1 Database to produce the Staffing Plan and the P-1. The Electronic Military Personnel Record System (EMPRS) is a separate application that the detailers use to view their constituents' digital records during the detailing process. The four systems represented here are all stovepipe mainframe applications that do not relate to each other except in a nightly transaction-oriented process for synchronization of data.

One year after this initial visit, it was discovered that there is a relational database model that contains a large amount of the data from these systems at NAVPERSCOM. This database is called the Navy Personnel Database (NPDB). This development sparked a change in the data migration plan. Instead of creating a data import routine that would import the ASCII files, a read-only view of the required data elements could be created that to link to the new database. This view can be custom created to match the requirements of the data model and save considerable time in the development of the application. This link will reduce the complexity of operation for the end-user and thus save the detailers time as well. As time continued on in the development of this thesis, it became apparent that an approval to link with this data would come in time to complete the thesis. Development quickly switched back to the original plan of importing the data from the delimited text files and further manipulating it from there. Future plans for this application should ensure a link is made to the existing production systems to enhance the usability of the application and the accuracy of the data.

3. Define Current Problems and Constraints

a. Existing Access Database

The current database essentially a collection of spreadsheets and macros that miraculously, although inefficiently, produce the P-1 and Staffing Plan. The P-1 is only updated twice a year because the task is so arduous. It is published to the Civil Engineer Corps secure server for viewing on the Internet.

The existing Microsoft Access database that is being used for the creation of the P-1 and Staffing Plan contains 43 tables, none of which has primary keys or established relationships. As explained previously, this type of data can be very dangerous because it allows for the duplication of data and potential data integrity problems. If two tables contain a member's name and location but differ in the location, which table is the correct one? This is the inherent problem with stovepipe systems that do not relate to each other.

The application also contains 105 queries and 100 macros. A macro is a saved sequence of commands or keyboard strokes that can be stored and then recalled with a single command or keyboard stroke. In this case, the macros execute a combination of queries and other macros to create relationships and match the billets to bodies, etc. The daisy chain of commands that are executed when producing the P-1 and Staffing Plan are so numerous and complex that even a skilled database administrator and programmer would have a hard time following. This setup makes it very difficult to troubleshoot any problems that may occur during the generation of the P-1 and Staffing Plan. As a general rule, macros should be reserved for either capturing keystrokes or carrying out a series of actions when opening an Access database. In the place of macros, VBA should be used because of its many advantages:

- VBA can trap and handle errors where macros cannot.
- VBA executes faster than macros, making the database performance more efficient.
- VBA is more portable than macros. If you want to move a form into another database, the VBA code automatically goes with it. With macros, you would have to find the specific macros that you needed to take as well.
- VBA gives you more flexibility and programmatic control where macros can only carry out a limited set of instructions.

b. Data Cleansing Problems

As with most transitions to a new system, data cleansing is always an issue. As previously outlined, there are several issues with the existing data set that must be addressed each time the P-1 and Staffing Plan are created. Appendix F is the page from the original user's manual for the current system and outlines what particular data elements must be modified or deleted prior to creating the plans. Some of this data is simply erroneous data in the legacy system that could be fixed by requesting the changes through proper channels. In the interview with the Community Management Analyst, he indicated that past attempts to request these changes were ineffective. He now just lives with the current anomalies and fixes them when he does the reports. Much of these corrections can be automated as they follow simple rules, but there are some that can only be done manually. For example, any personnel that come through the system that have a rank of Commander or above and a designator of 5105 are automatically deleted because they are reservists who are not tracked by this office. Below the rank of Commander, there are many active duty CEC Officers who still have a designator of 5105 and the only way to know which ones belong in the system is through corporate knowledge. The Community Management Analyst knows all the active CEC Officers and can pick out the ones that do not belong. This is something that cannot be automated and must be set up for the Community Management Analyst to check using traditional manual methods. The new database automates the data cleansing wherever possible and has several forms for the manual cleansing that is necessary. Even after this manual cleansing, there is no guarantee that the data will be correct. The crux of the problem is that PERS 4413 is trying to apply a different business model to the legacy systems. These systems do not support the way the CEC detailers want to do business. Because of this, they have to create workarounds to get the job done. Chapter VI addresses in detail how the challenge of importing and cleansing the data as well as relating the data to the proper entities was accomplished.

4. Define Future Objectives

The objectives of this database application are two-tiered. The first tier will meet the requirements of the CEC Detail Shop by providing a quick, easy to use application that imports the legacy system data and produces the P-1 and Staffing Plan. This is the only deliverable that they are looking for. The second tier will meet the thesis requirements by expanding database application features to include a prototype detailing management system. In this application, the user will be able to move personnel from one duty station to another. They will also be able to add, update, and delete all the background data as well as personnel and billet information. Additional useful reports will be created as well as areas to put a service-member’s duty preferences.

5. Project Scope Definition

The project scope will include the following:

- Data import feature to bring in the two delimited files from ODIS
- Automated data cleansing routines to take care of known data problems
- Manual review and manipulation of imported data

- Data management tools to maintain background information such as qualification codes, designator codes, etc
- Easy to use forms for updating activity, billet, and member information
- P-1 and Staffing Plan reports
- Other reports deemed useful

The project scope will not include the following:

- P-1 and Staffing Plan reports in the *exact* format that they currently exist in
- Real-time link to existing production systems
- Web interface

D. FEASIBILITY

1. Hardware and Software Requirements

The hardware and software requirements for this system are in line with current IT-21 and industry standards. The application will be developed using Microsoft Access 2002 but saved in the Microsoft Access 2000 format. Since this application has only one user, there is not a need for a centralized server to host the application. The host computer that the application will reside on should be at least a Pentium III, 700 MHz computer with a minimum memory requirement of 128 MB and 1 GB of free hard drive space. Computers that do not meet these standards may experience degradation in performance of the database application resulting in frustration and possibly application errors.

2. Cost Benefit Analysis

The costs for this project are broken out into three options:

- Do nothing
- Hire a contractor
- Utilize a NPS student

A rudimentary cost analysis is presented in Table 1 below:

	Do Nothing	Hire Contractor	NPS Student
Hardware	$0.00	$0.00	$0.00
Software	$0.00	$0.00	$0.00
Labor (320 hours @ $200/hr)	$0.00	$64,000.00	$0.00
Travel	$0.00	$0.00	$3,000.00
Reference Materials	$0.00	$0.00	$400.00
Totals	**$0.00**	**$64,000.00**	**$3,400.00**
Rank =	1	3	2

Intangible Factors	**PERS 4413**	**PERS 4413S**	**PERS 4413A**	**PERS 4413E**	**PERS 4413C**	**Average**
Security	8	6	8	7	7	7.20
Compatibility	8	9	8	9	10	8.80
Reliability	9	9	7	8	9	8.40
Scalability	4	4	5	7	5	5.00
Affordability	7	6	6	6	7	6.40
Ease of Use	8	10	8	9	8	8.60

Wtd pts = Average * Points

Technical Evaluation	**Do Nothing**		**Hire Contractor**		**NPS Student**	
	Points	**Wtd Pts**	**Points**	**Wtd Pts**	**Points**	**Wtd Pts**
Security	2	14.40	9	64.80	7	50.40
Compatibility	7	61.60	8	70.40	8	70.40
Reliability	3	25.20	9	75.60	7	58.80
Scalability	4	20.00	8	40.00	8	40.00
Affordability	10	64.00	4	25.60	8	51.20
Ease of Use	4	34.40	8	68.80	7	60.20
Total Score	**219.60**		**345.20**		**331.00**	

Table 1. Cost Analysis For Project

Based on a cost only analysis, the best option is to do nothing and keep the current system. A cost analysis should not be done solely based on cost. It should also include intangible factors as shown above (Cook 2001). The idea is to interview the primary stakeholders and find out what their priorities are in a new application with respect to each of the categories. The scores for these factors are averaged and multiplied by the technical evaluation scores for each category. The technical evaluation scores are obtained by evaluating each option with respect to each of the evaluation criteria.

This simple comparison of estimated costs reveals that the two cheapest alternatives are to do nothing and fund a NPS student to do the work, however the best overall evaluation score was obtained by the contractor. The NPS student option was a close second and because it was considerably cheaper to do fiscally as well as politically it is the winning choice in this analysis.

III. REQUIREMENTS ANALYSIS

A. DATA REQUIREMENTS (ERD)

Based upon the initial discovery trip, subsequent conversations with the PERS 4413 detail shop, use of existing systems, and reports a data model was developed to capture all the required entities for the system. The compilation of this data into structured entities with relationships and business rules is known as the database schema. The database schema is the foundation on which the database and the applications are built. (Kroenke 2000, p 30) After sifting through the available information, LT Rader and I held a brainstorming session to determine a potential list of data elements that may be required. The result of this session is depicted below in Figure 5.

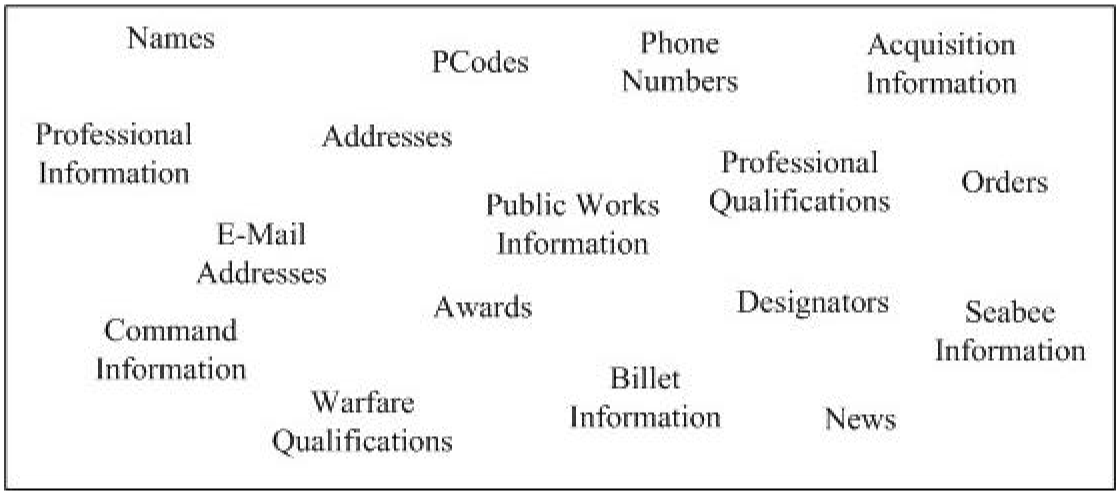


Figure 5. Brainstorming Session Results

Based on these results and further discussion, a list of primary entities was determined for this application and is displayed below in Table 2. This list does not include any lookup tables or intersection tables which will be discussed later on in this chapter.

MEMBER	PREFERENCES
BILLET	ACTIVITY
QUALIFICATION	

Table 2. Primary Entities

Using this list of entities, an Entity Relationship Diagram (ERD) was created to outline the database schema and is shown below in Figure 6.

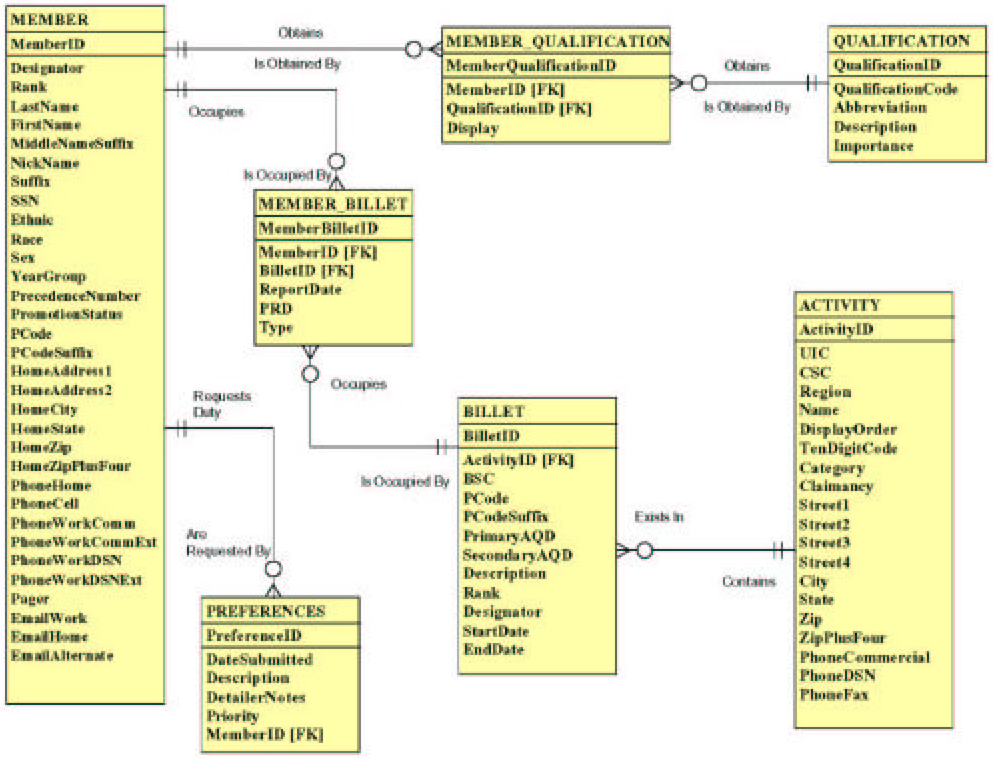


Figure 6. Entity Relationship Diagram

The key elements of an ERD are entities, attributes, identifiers, and relationships. Using a grammatical analogy, entities can be defined as the nouns, or something that can be identified in the users' environment that they want to track. Attributes are the adjectives, or properties of the nouns that describe the entity's characteristics. Relationships are the verbs that describe how the entities interact with each other. While a detailed description of each entity and their attributes can be found located in the database schema located in Appendix G, the ERD above provides a quick and easy to read view of the database schema. When a relationship between entities exists that is a many to many relationship, a special table must be created to capture all the records for those entities. This table is called an intersection table and it contains the keys of the two tables in the relationship. The combination of those keys can make up the primary key, or unique identifier, of the table. An alternate method of uniquely identifying an intersection table is to create a separate auto-numbered field that becomes the primary key for the table. This field is called a surrogate key and has advantages in that when variables need to be passed to identify which record needs to be displayed or updated, you only have to pass the one variable vice the combination of keys that would have made up the primary key.

The focal point for this ERD is the MEMBER table. An instance of the MEMBER table represents one CEC service member. That officer is described by the many attributes shown above in the ERD. Each member can have from zero to many duty preferences. Members relay their personal duty preferences and priority to their respective detailer prior to negotiating orders. These preferences are logged by the detailers and stored in the PREFERENCES entity. Throughout the course of their career, a member can obtain from zero to many qualifications. Qualifications include, but are not limited to Professional Engineer Certification, Registered Architect, and Seabee Combat Warfare. All of the possible qualifications are stored in the QUALIFICATION entity. Qualifications have codes, abbreviations, descriptions, and importance (sorting order for display). A single qualification can be obtained by zero to many members. Because the MEMBER to QUALIFICATION relationship is many to many, an intersection table was created to capture all the records of each service member's qualifications. This intersection table, MEMBER_QUALIFICATION, contains an additional field called display which is a yes/no field designed to allow the user to select whether or not that particular qualification should be displayed in the P-1 or Staffing Plan. For example, members may obtain their Engineer in Training (EIT) certification and then later on obtain their Professional Engineers (PE) certification. Once a member has their PE certification, there is no need to display the EIT.

Members are assigned to billets represented by the BILLET entity. Each member is usually assigned to a billet, but in some rare cases a member may not be assigned to a billet. A billet does not have to have a member and a member can be assigned to many billets during his or her career. The relationship between MEMBER and BILLET is also a many to many relationship. The intersection table MEMBER_BILLET was created to capture all instances of members and billets. A member in a billet will have a report date, and a projected rotation date (PRD). Additionally, there is a field in this table called Type. Upon reviewing my notes from the requirements analysis trip and also the existing data that is to be imported, I came across some additional fields that are tied to a member (UltimateUIC and UltimateBSC). These fields describe an ultimate duty station to which the member is headed and are different than the UIC and BSC information on the member. Also included are the ultimate estimated reporting date (ULTEDA) and ultimate projected rotation date (ULTPRD). These fields will be populated when a member has received orders to another duty station, but has not yet left the current duty station. PERS 4413 is interested in not only where the member is, but where their ultimate destination will be. The existing ERD did not have any means to capture this information. Two options presented themselves to resolve this issue. The first option was to add another intersection table specifically for ultimate destination information that would essentially contain all the same data as the MEMBER_BILLET table, except it would hold only that information on ultimate destinations. This idea was quickly discarded because it greatly increased the complexity level of the data model. The second idea was to add an additional field into the MEMBER_BILLET table that would be a required field and toggle between Current, Ultimate, and Previous. A billet is a billet regardless of whether it is a current station or ultimate destination for a member. One member's ultimate UIC and BSC may be the same as another member's current UIC and BSC. The additional field accounts for the case where a member has a current billet and an ultimate billet without adding too much complexity to the model and enables the application to track the complete duty station history of a member.

Billets are associated with an activity represented by the ACTIVITY entity. An activity can have many billets, but a billet can only be associated with one activity. An activity is not required to have a billet, but a billet must be associated with an activity.

A trained eye can quickly look at this ERD and see that its design is not completely normalized. Normalization of data refers to the way data attributes are grouped to form stable, flexible, and adaptive entities. Normalization is the procedure that is used to simplify entities, eliminate redundancy, and build flexibility and adaptability in the data model. (Whitten 1998, pg. 337) This database has been de-normalized intentionally. Normalization and the reasons for de-normalization are discussed in detail in Chapter IV.

Finally, there are several lookup tables that were created to build a user-friendly interface and to ensure the user inputs the correct values for designator, P-Code, and rank. These tables are not related to any of the tables, but are called upon to show a valid list of values for their respective fields in the database application.

B. APPLICATION REQUIREMENTS (DFD)

The application must support complete data management ranging from maintaining the background data such as P-Codes, designators, and qualifications through the management of a member as the officers progress through the course of their careers. In order to assign members to billet positions, the application must support the creation of new activities and billets as well as functionality to update and delete them as well. The Data Flow Diagram (DFD) describes the system processes that support this functionality and how it interacts with the data in the database.

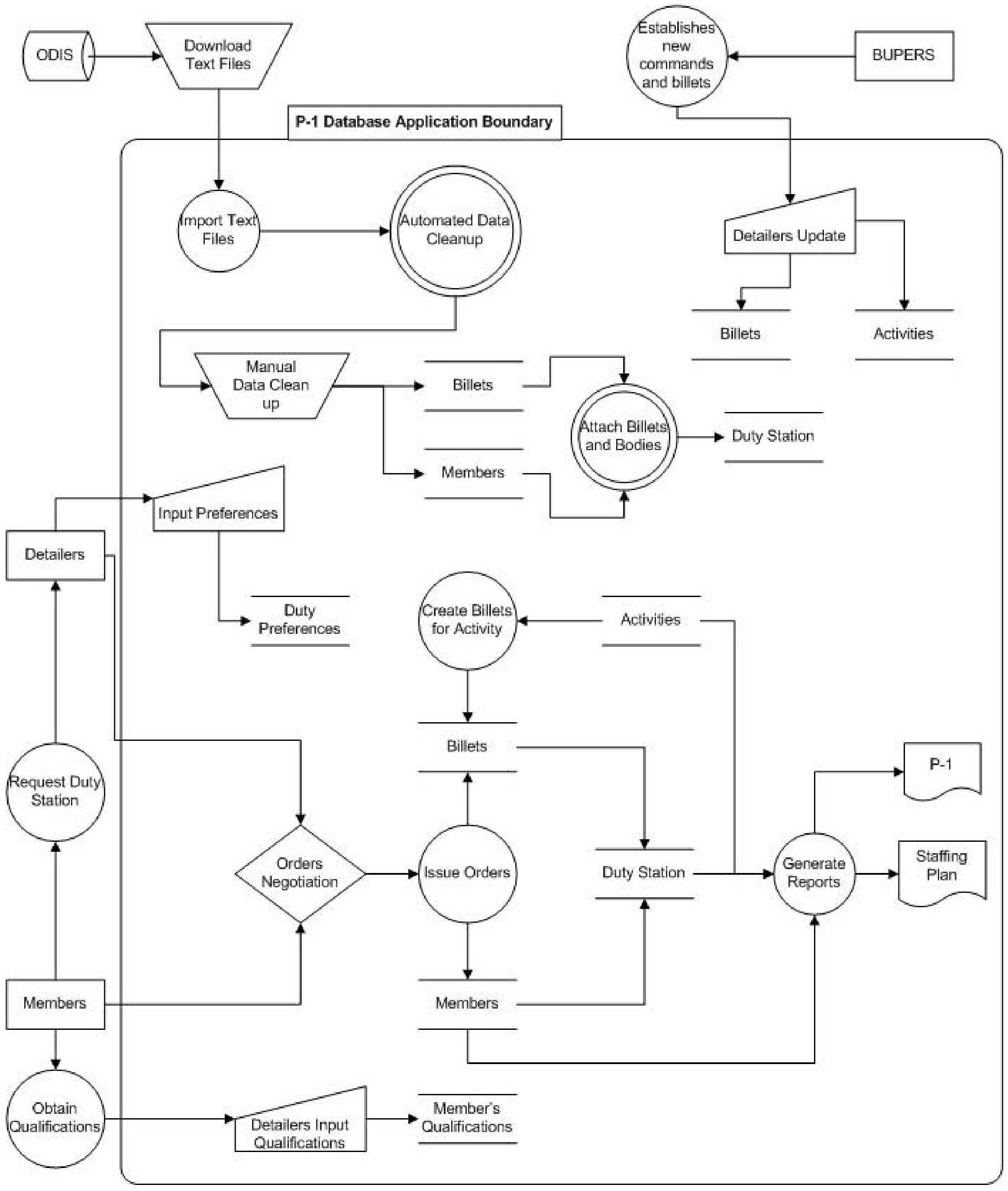


Figure 7. Data Flow Diagram

C. IDENTIFY REPORTS

As discussed in previous chapters, there are two main reports that the CEC detailers use in the daily execution of their duties. These two reports are the P-1 and the Staffing Plan. The P-1 is used primarily to provide information to the CEC community whereas the Staffing Plan is the tool that is used to help the detailers with the assignment of members and the overall balance of the Civil Engineer Corps. The P-1 provides information on where all the CEC officers are located at a given point in time as well as their billet positions, qualifications, activity phone numbers, addresses, and the officer's reporting date. Representative samples of the P-1 and the Staffing Plan are discussed and presented in Chapter V. The Staffing Plan is very similar to the P-1, but shows the following information as well:

- Officers who are separating (retiring or resigning)
- Pending gains (projected new officers joining the CEC)
- Exception personnel and billets (billets or personnel that are not assigned and are pending a decision)
- Projected rotation dates for the officers
- Ultimate duty station information (Activity and estimated report date)

Additional reports that are not required, but may be beneficial include the following:

- Member reports filtered by rank and projected rotation date
- Billet reports filtered by qualification requirements
- Member reports filtered by qualifications obtained
- Members in NMCB activities sorted by year group and PRDs
- Report on member's duty history
- Report on billet's history (who was assigned and when)
- Billet report filtering by type (Public Works, ROICC, Seabee, etc)

These additional reports will make it easier for the detailers to get an overall picture of where there constituents are, what they need for their career, and what billets the CEC needs to fill.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DESIGN

A. DESIGN PHILOSOPHY

Any application is useless if the end-users do not find it easy to use and helpful in the execution of their jobs. If the application is more of a burden than help, resourceful users will find another way to work and leave the application behind. With that in mind, the tried and true philosophy of keeping it simple is used. The database design and application design are modeled with the goal of providing an easy to use interface that quickly produces the information required based on available data. A poorly designed database can make designing the application extremely difficult, if not impossible. Applications with flawed designs could experience performance problems, expandability issues, and data integrity problems. (Blackburn, et. al 2000, pg. 23) A comfortable balance between an expert technical design, user-friendly interface, and reporting capabilities is obtained by de-normalizing the database design to accommodate for ease of use while not sacrificing performance or the reliability of the data.

B. DATABASE DESIGN

A good database design should have the following characteristics (Whitten 1998, pp. 407-408):

- The data model is simple. Data attributes that describe an entity should describe only that entity.
- The data model is essentially non-redundant. Each attribute, other than foreign keys, describes at most one entity.
- The data model should be flexible and adaptable to future needs. The database structure should be capable of being extended or modified without significant impact to current programs.

In order to achieve these basic goals in database design, a specific technique called normalization is utilized.

1. Normalization

A fully normalized database is said to be in domain/key normal form (DK/NF). This is the highest level of normalization that will prevent all modification anomalies. To understand this discussion, a proper definition of terms used is presented below in Table 3: (Kroenke 2000, pp 113-126)

Term	**Definition**	**Example**
Relation	Two-dimensional table. Each row holds data that pertain to some thing or a portion of some thing. Each column (field) contains data regarding an attribute.	Table: MEMBER; Each row represents a service member and each column is a different field describing the member (Name, SSN, Sex, etc)
Functional dependency	A relationship between or among attributes. Given the value of one attribute, the value of another attribute can be looked up.	Given a particular activity (ActivityID in table ACTIVITY), you can look up what billet(s) are in that activity. *Billet is functionally dependent on Activity*.
Candidate key	A group of two or more attributes that uniquely identifies a row.	In the table MEMBER, MemberID and SSN are both keys because they uniquely identify a particular member.
Primary key	The candidate key chosen to be the main key, which identifies the row.	MemberID is the primary key in the MEMBER table
Modification anomalies	Changing data in one table causes an undesirable change in another attribute.	In the table MEMBER, the attribute Designator describes what designator each member has. If the designator does not come from a predefined list (table), a deletion of one member could result in the deletion of that particular designator if the member was the only one to hold that designator.
Constraint	Any rule governing the static values of an attribute. Includes edit rules, intrarelation and interrelation constraints, functional dependencies, and multi-value dependencies.	SSN in table MEMBER can only have 9 characters. A member can have many qualifications and a qualification can be obtained by many members.
Domain	A description of an attribute's allowed values (both physical and logical).	In the table MEMBER, the domain for the attribute Sex is (M, F). Logical domain is member's gender.
Transitive dependency	A table's key determines one attribute and that attribute in turn determines another attribute.	Consider Member, Rank, and Paygrade. The member determines the rank and the rank determines the paygrade, thus the member indirectly determines paygrade. (If a=b and b=c, then a=c)
Multi-value dependency	A condition in a relation with three or more attributes in which independent attributes appear to have relationship they do not have.	Consider a member with (MemberID, PCode, and Qualification). A member can have multiple values of PCode and Qualification. PCode and Qualification have no relationship, but appear to in the relation.

Table 3. Database Design Definitions

While DK/NF is the highest form of normalization, many database designs do not meet this criteria and are thus in other forms ranging from First Normal Form (1NF) to

Fifth Normal Form (5NF). Each of these forms is summarized below (Kroenke 2000, pp. 120-126):

- First Normal Form (1NF) – Any table of data that meets the definition of a relation.
- Second Normal Form (2NF) – All non-key attributes are dependent on all of the keys. An example of this would be if the MEMBER table contained the field BilletTitle. BilletTitle is a non-key attribute that is not dependent on MemberID. To get this table into 2NF, a separate table, BILLET, needs to be created and the field BilletTitle needs to be moved to this table.
- Third Normal Form (3NF) – There are no transitive dependencies.
- Boyce–Codd Normal Form (BCNF) – Every determinant is a candidate key.
- Fourth Normal Form (4NF) – It is in BCNF and there are no multi-valued dependencies.
- Fifth Normal Form (5NF) – Relations that can be divided into sub-relations, but then cannot be reconstructed. This is an obscure concept that has no clear meaning and is included here for completeness only.
- Domain/Key Normal Form (DK/NF) – All constraints on relations are logical consequences of domains and keys.

The Entity Relationship Diagram was introduced in Chapter III. (Figure 6) This diagram was the result of several iterations during the design process. The initial design was closer to DK/NF than the final version. For optimization reasons including ease of use and ease of reporting, the original design was de-normalized to get to the database design its final state. The original ERD is shown below in Figure 8.

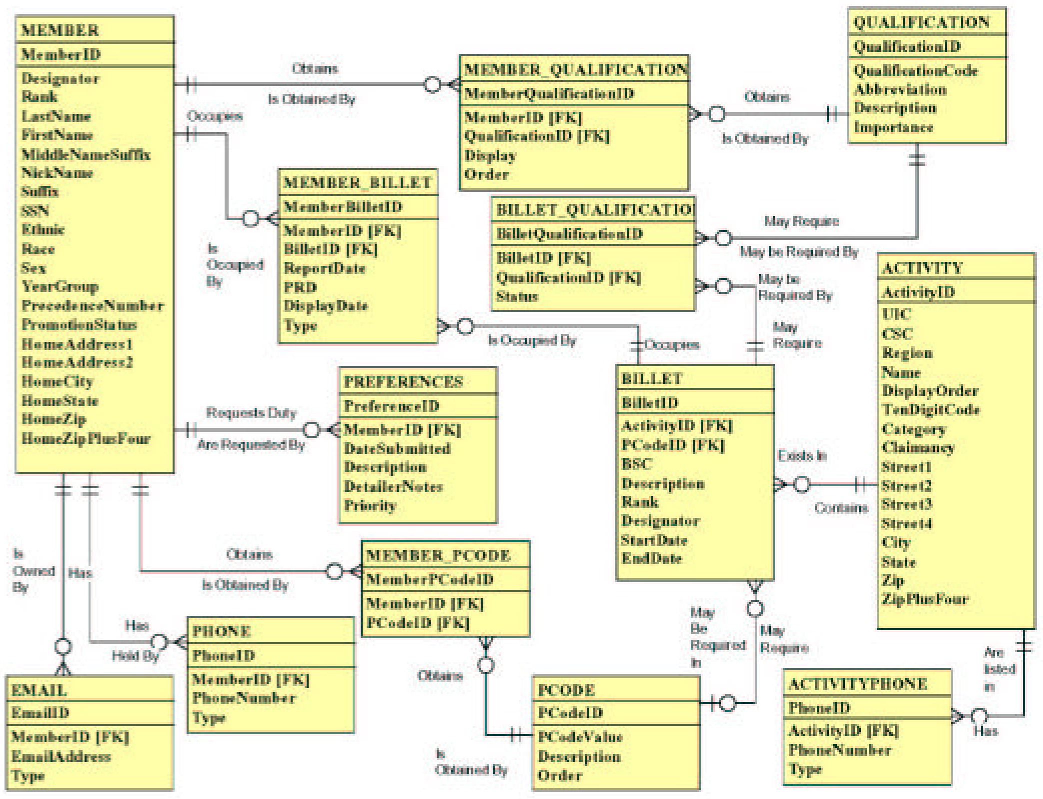


Figure 8. Original Entity Relationship Diagram

As shown above, the original ERD was much more complex than the final version. Several tables were removed in favor of de-normalization. While normalization avoids modification anomalies, sometimes its value is not worth the price and ease of use. The following discussion details the reasons for de-normalizing each table.

- EMAIL – In reality, a member can have any number of email addresses, but do we really want to track all of them? This was discussed with the detailers and they were only concerned with the primary and possibly a secondary address. With that in mind, the joins in the model were simplified by adding three email address fields to the table MEMBER (Work, Home, and Alternate).
- PHONE – It is reasonable to assume that a member will only have a finite quantity of phone numbers. Based on this assumption, the PHONE table was deleted and fields were added to the MEMBER table for phone numbers (Home, Work-DSN, Work-Commercial, Cell, and Pager)
- ACTIVITYPHONE – The same argument made above for the member's phone numbers can be made for the activity's phone numbers. Phone

number fields were added to ACTIVITY and the table ACTIVITYPHONE was dropped.

- PCODE – A member can possibly obtain more than one subspecialty code (PCode), but the data in the current production systems only support one for each member. Since the CEC detail shop did not have an interest in tracking anything other than the main subspecialty code for the member, the P-Code field was added to the MEMBER table and the relationship was dropped in favor of a lookup table.
- BILLET_QUALIFICATION – A billet has a maximum of two qualifications (primary and secondary). Because of this finite number, the many-to-many relationship between BILLET and QUALIFICATION seemed unnecessary. To keep with the design philosophy of simplification, the intersection table was dropped and two fields were added to BILLET (PrimaryAQD and SecondaryAQD). These fields are still validated in the database by looking up their values from QUALIFICATION and not allowing any entries not in the table that do not exist in the table QUALIFICATION.

The process of de-normalization produces an effect that is worth noting. While the database is easier to use and the relationships are less complicated, there is a potential for data anomalies in the future. For example, if information for P-Codes were to change in the Navy as they recently have, a large data update would be required for both the MEMBER and BILLET tables. An example of this would be the recent change in the Information Technology P-Code from 0089 to 6201. Because the data in the MEMBER and BILLET tables are not related to the PCODE table via a foreign key, each member and each billet that has a P-Code of 0089 would have to be changed to 6201. This can be done through automated routines, but requires specialized skill and can be dangerous to execute. If the data model were normalized, the only change in the database would be to change 0089 to 6201 in the PCODE table and the update would be done for the entire database. The exact same case can also be made for the designator field in MEMBER and BILLET.

2. Data Model Verification

Data model verification is a process in which the ER model is verified against end user views and required transactions, security, and business-imposed data requirements and constraints. During this process each entity, its attributes, allowable values, and any business rules, which apply to any processing transactions, are evaluated for the following reasons (Dr Xinping SHI):

- Attributes may lead to a revision of the entities themselves.
- Attributes provide clues about the relationships as they are defined by the primary and foreign keys.

An example of this analysis is the PCode field in MEMBER and BILLET. A subspecialty code is a combination of a four-digit number and one character letter. The number defines the area of expertise and the letter defines the level of education and experience. Originally, I had designed this to be one field that was a variable character field with a length of five. After evaluating how the end user would be interfacing with the system, I found that this method was a bit cumbersome. For each subspecialty code, nine possible character codes define the level of education and experience. Since there are 108 different codes for the subspecialty rating, the number of records in the PCode table would have multiplied by nine yielding 972 records to sort through when searching for the correct value. The solution was to split the field into two parts, the first being the subspecialty code itself, and the second being the one character suffix that describes the level of education and experience. This makes looking up the values of the PCode much easier for the user and still maintains the data integrity by not allowing any values other than those in the list.

Another issue that was discovered during this process was how to deal with the name of a member. The member name data that is imported comes in one field with the last name, first name, middle name (or initial), and suffix (Jr., Sr., etc) separated by a space. Because each member's name can vary, it was impossible to split each part into separate fields. The last name and first name split easily, but when it came to split the middle name and suffix into their respective fields a problem surfaced. Some members have two middle names. For example, if I were to take the name field and split it four times into LastName, FirstName, MiddleName, and Suffix it would work most of the time for standard names. If a member had a name like Weinstein Brian Louis Wiley Jr., then the middle name would become Louis and the suffix would become Wiley and Jr. would have no place to go. The solution was to separate the first and last names of the members and keep the middle names and suffixes together in one field MiddleNameSuffix.

3. Logical Design

The logical design of the database was accomplished using the CASE (Computer-Aided Systems Engineering) tool Visible Analyst. The entities, relationships, attributes, data types, domains (allowable values), and default values were created in Visible Analyst. Visible Analyst maintains a repository, where all specifications associated with the database design are kept. The repository functions interactively with the diagramming and rules to create an organized database of project information for the project automatically as it is developed. The project repository is automatically updated and maintained during all work sessions, and it can be accessed at any time to manually add, delete, edit, or review entries. From the repository, Visible Analyst allows the designer to generate the database schema and export it to the Access database. The Access database is created with all the tables, fields, relationships, default values, and referential integrity settings from Visible Analyst. This can be an invaluable tool while working on the design of the database to quickly change the design and export it to Access for validation and verification. Visible Analyst also includes analysis tools to identify syntax errors (unnamed entities and relationships) as well as potential normalization errors. During the normalization analysis, Visible Analyst checks for and warns of many-to-many relationships, optional-to-optional relationships, and mandatory one-to-one relationships in both directions.

As mentioned previously, the ERD for this project changed several times during the database design process. I made all the changes in Visible Analyst to ensure consistency and then export the database schema directly into Microsoft Access. Visible Analyst would drop the existing tables, add the new tables, and recreate the relationships in the database. The only downside to working in this manner is that any data in an existing table that is dropped by Visible Analyst and replaced with a new table is lost. There are two ways I used to work around this inconvenience. The first is to keep a copy of the database prior to updating it and then import the data that is dropped from the new one. The second method is to export only those tables, which have changed in Visible Analyst. For example, if I made modifications to the MEMBER and BILLET table but left the ACTIVITY table alone, I would exclude the ACTIVITY table in Visible Analyst from the export procedure. The MEMBER and BILLET tables would then be updated and the ACTIVITY table would remain intact with all of its data.

The result of the logical design can be partially seen in the ERD, but to gain the full picture of each entity and its attributes, the data dictionary is needed. The data dictionary breaks down each entity and shows each attribute's data type, length, allowed values, and whether it is a required field. This data dictionary can be found in Appendix G.

4. Physical Design

While logical design is primarily concerned with normalization, the physical design relates mostly to performance. Optimizing the physical performance of the Database Management System (DBMS) can be a challenging task for large-scale production systems. Since this project is a prototype using Microsoft Access, the concern for the physical design of the system is not as great as it would be for a large-scale enterprise system. In reality, there may be only one user of the system and the application will most likely be run off his hard drive. If the other detailers desire to use the system, then a slightly different physical implementation will be necessary. In this case the database would be split into two parts, the first part being the actual data tables, queries, reports, macros, and modules and the second part being the user interface forms. The first portion would be installed on a shared network drive only accessible by the CEC detailers and each detailer would have a client installed on their local machine that would have their own personal copy of the front end and link to the centralized database. This would allow all to share the same data while minimizing any maintenance upgrades to the application logic keeping all the queries, macros, and modules in one central location. This scalability issue is discussed more in Chapter VI.

C. APPLICATION DESIGN

In the original concept of the project, consideration was given to a web-based design as well as a client-server design. During the requirements analysis trip, it became clear that there was no support for a web-based detailing management system for the CEC detailers. The IT staff would not support any new applications that would require their support (hardware or software) without a sufficient review. Time constraints for the project dictated another solution. A small Access database application run by a few users to help them in their job would not be a concern for the IT staff. The final decision was to design the application for the one user who was likely to use it, with the option of splitting it up into a client-server type model. The product chosen for the prototype application was Microsoft Access due to its versatility, cost, and the fact that it is a very robust database system for smaller applications like this one.

1. Hardware & Software

The database application is designed to run on PC-compatible machines meeting a minimum of the following requirements:

- Processor speed should be at least Pentium III, 700 MHz or greater
- Hard drive space should have at least 1 GB free
- Computer should have a minimum of 128 MB of memory

To run the application, the user will require Microsoft Access 2000 running on a Windows platform. The preferred operating system is Windows XP Professional, as it is the most stable of all the Microsoft Windows platforms and is the industry standard for office computers. The database application will run on Windows 2000 and should run on earlier versions of Windows (9X and NT) but has not been tested on these operating systems.

2. Migration from Legacy Systems

In an ideal situation, the new application would take over the functionality of the legacy system, and the legacy system would be dropped when the migration was complete. In this particular case, the application is not a replacement for the current production systems but rather a prototype for what could be used as a detailing management system. The benefit of this prototype is that it has functionality that can be used in concert with the legacy systems to help the CEC detailers execute their duties. A true migration from the current legacy systems requires work and analysis far beyond the scope of this thesis. This issue is addressed in Chapter VII.

3. Security

The security of the application is important for two reasons: 1) to protect the code and 2) to protect the data. Microsoft Access has a built in security system that allows you to add simple security such as adding a database password, but it also allows very complex security schemes including encryption and security groups. Since this project contains sensitive Privacy Act data and has complex forms and queries, I chose to secure it using user-level security instead of a simple database password.

a. Database Password

To protect against unauthorized personnel accessing the database, a simple database password can be set. This protects the database with the password set but still allows users to use Access and other databases on the machine or network. Microsoft Access encrypts the password so that the database password is secured and cannot be accessed by reading the database file directly. (Blackburn 2000, pg. 549) This password only restricts opening the database. Once the database is open, the user has full access to all its objects unless further security is set.

b. User-Level Security

By default, all users have rights to all objects in the database. Microsoft Access allows the administrator to add user level security by assigning permissions for database objects (forms, modules, reports, etc) to either specific users or groups of users. Access comes with two default groups—Users and Admin. The Admin group has rights to administer the database, can perform any action on any database object, and can create users and groups. By default, all users have rights to all objects in the database. Fortunately, Access comes with a User-Level Security Wizard that allows the administrator to set up security users and groups in a quick, easy to use interface. For this project, I chose to create one additional group—Full Data Users. I set the permissions for this group to allow them full permissions to edit data but not alter the design of any database objects. Following good security practices, I also disabled any permissions for the default user group as well as user-level permissions for Admin and User. I created accounts for each of the detailers and placed them in the Full Data User group assigning them user names and passwords. I also created an account for myself as the administrator of the database. Currently I am the only one with full rights to the database objects, but after proper training PERS4413S (Community Management Analyst) will be given escalated rights so he can properly manage the database.

Although the Access database password and user-level passwords is supposedly encrypted and cannot be hacked into, there are several programs available that recover lost passwords for Microsoft Access databases and other Microsoft Office products. In theory, a malicious user could use one of these programs to gain the database password and gain access to the system. This security scheme does protect the database application and data from the average user and is sufficient for the prototype application.

D. DESIGN SUMMARY

Although the database design is not an optimal model, extensive effort was undertaken to ensure that the model would support the users' requirements as well as maintain data integrity. The original model was de-normalized in several areas to accomplish this objective. Data integrity for these areas was strengthened by providing allowable values for entry with lookup tables. The resulting data model supports the users need for a comfortable interface while minimizing the amount of de-normalization. Due to the nature of the data contained in the application, a user-level security system was put in place to protect the data from being viewed by unauthorized personnel and to protect the application from accidental tampering.

THIS PAGE INTENTIONALLY LEFT BLANK

V. DEVELOPMENT & IMPLEMENTATION

A. DATA IMPLEMENTATION

Data implementation involves the physical construction of the database based off the logical design. The generation of the database from the CASE tool Visible Analyst was discussed in Chapter IV. Implementation of the data requires decisions on appropriate field type values and lengths, default values, input masks (to force correct data entry), and visual formatting (for example, making a date display in a particular manner). An example of this are the phone fields located in the MEMBER table and the ACTIVITY table. If the application is only going to deal with phone numbers in the United States, then it would make sense to allow just a ten digit number field that is formatted for standard phone numbers that users are accustomed to. In this case, many phone numbers are international with varying digits and sometimes character codes. The decision to leave the phone fields as a variable character field capable of holding 30 characters was made based on the variety of phone numbers in the data.

A final important area involved with the data implementation is the creation of indexes. Indexes improve the performance of any queries that are performed against a field and are best kept for fields against which the user will be doing many queries, sorting, or grouping. Indexes should only be used where deemed necessary because whenever a new row is inserted into a table, a new entry is made into every index in that table. Indexes add to the size of the database file, reduce concurrency (the ability of more than one user to modify a page at the same time) in multi-user applications, and decrease performance when you update data in fields that are indexed or when you add or delete records.

Microsoft Access has a feature which produces a report on selected tables and other objects in the database. This report is a comprehensive list of each table, its attributes, and detailed information on each field. This is often referred to as the database schema or sometimes a data dictionary. The complete database schema for this project can be found in Appendix G.

B. APPLICATION STRUCTURE DESIGN

Traditional Access applications have been a single machine application with only one user. As times changed and networks became more prevalent, the need for shared applications arose and Microsoft provided further advanced versions of Access. The requirement for users to share data spurred the development of client/server applications. Today, there are several different methods to structure your database application. The three methods I will discuss are single-tier applications, two-tier applications, and three-tier applications. The following terms are commonly used when referring to application structures and their implementation:

- Presentation layer – Contains the graphical and visual elements of the application and is often referred to as the Graphical User Interface (GUI).
- Application logic layer – Contains the business rules associated with an application.
- Data layer – Contains the application data, usually in a database.
- Client – The local computer used by the end user.
- Server – Centralized server that has shared access across many users.

1. Single-Tier Applications

A single-tier application is one where all the layers described above reside as one unit on one machine. This type of application is relatively easy to build and implement initially. The downside to this application structure is that there is no sharing of data across users. It is an excellent structure for a small application designed for just one user.

2. Two-Tier Applications

A two-tier application is an application where the presentation layer resides on the client and the data is located on the server. The application logic may reside on either the client or the server. When it resides on the client, the application structure is typically called a "fat" (sometimes called thick or rich) client. Conversely, when the application logic resides on the server with the data, it is called a "thin" client. The separation into two tiers allows the sharing of data, consistency of usage, and simplified report generation. Two-tier fat client implementations can be difficult to maintain because whenever there is any change in the application logic, the change must be applied to each client using the system. Fat client implementations also require the clients' machines to be sufficiently powerful to run both the application layer and the presentation layer. Thin client applications are easier to maintain, as all the application logic is located in one place. The trade-off with this type of implementation is that it puts a larger strain on the server to support multiple requests to the application logic layer.

3. Three-tier Applications

As the name implies, three-tier applications are broken into three separate parts. The presentation layer resides on the client, the application layer is located on a business (or application) server, and the data layer resides on a database server. This model can produce the most efficient and flexible applications. However, three-tier applications are also the most complex to build. Microsoft Access cannot implement a three-tier application fully by itself, but can play a role in the whole system by being part of the presentation or data layer. (Blackburn 2000, pg. 19)

4. Structure Selection

For this thesis, I chose to use a thin two-tier application structure. For the size of the application and its expected use, this structure offers the greatest efficiency and flexibility while also simplifying the maintenance of the application. Any further changes to the application logic can be accomplished in one place, on the server, instead of making changes to each client's computer. At the time of this writing, each user at the detail shop is still working on outdated, less powerful machines. Shifting the processing load to the more powerful server will increase the performance of the application.

C. GRAPHICAL USER INTERFACE DESIGN (GUI) – USER MANUAL

For the user, a graphical user interface (GUI) is the application. The GUI is how the end-users accomplish their tasks, and great care must go into designing comfortable, easy to use forms for the add, update, and delete requirements of the application. All forms were designed for a user screen resolution of 1024 x 768. Extensive effort was put into the design of each form to display the information in a logical manner and minimize the number of mouse clicks required for the user. At the same time, too much information on one form can create confusion for the user, so a careful balance must be maintained. In designing the interface, I attempted to adhere to three main principals:

- A form should serve as the main interface to the information in the database including all data operations like adding or deleting records.

- A form should link all of the database information components together. For example, preferences and members only share a numerical relationship as far as Access is concerned, but on the member form the end-user can add, edit, update, and delete all the preference information for that member.
- The forms should provide for an intuitive navigation through the program. The switchboard, which controls all of the interfaces to the application, can be used as well as a menu system on each form.

For this database application, I chose to use a standard Access switchboard to control how the user interacted with the database. The switchboard presents a logical flow of sequences for the user, organized by type of action. A screen shot of the main switchboard is shown below.

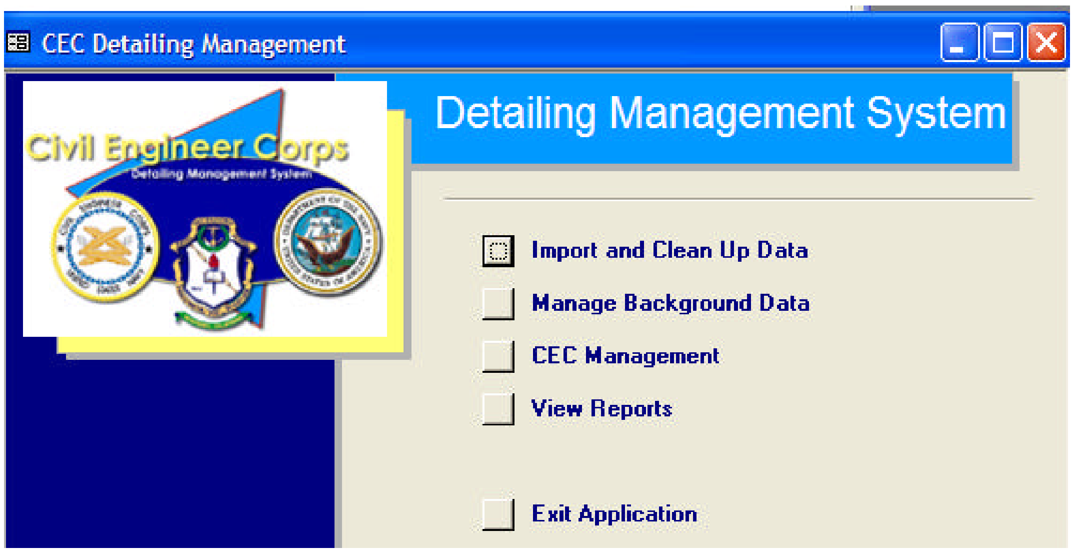


Figure 9. Main Switchboard

The main switchboard leads the user to four different areas—data import/cleanup, CEC member management, background data management, and report viewing. Each of these switchboards controls the user's access to specific areas of the application.

1. Data Import and Cleanup

If the user has not already imported the data, there is a four-step process that is outlined in the Data Import and Cleanup Switchboard shown below in Figure 10.

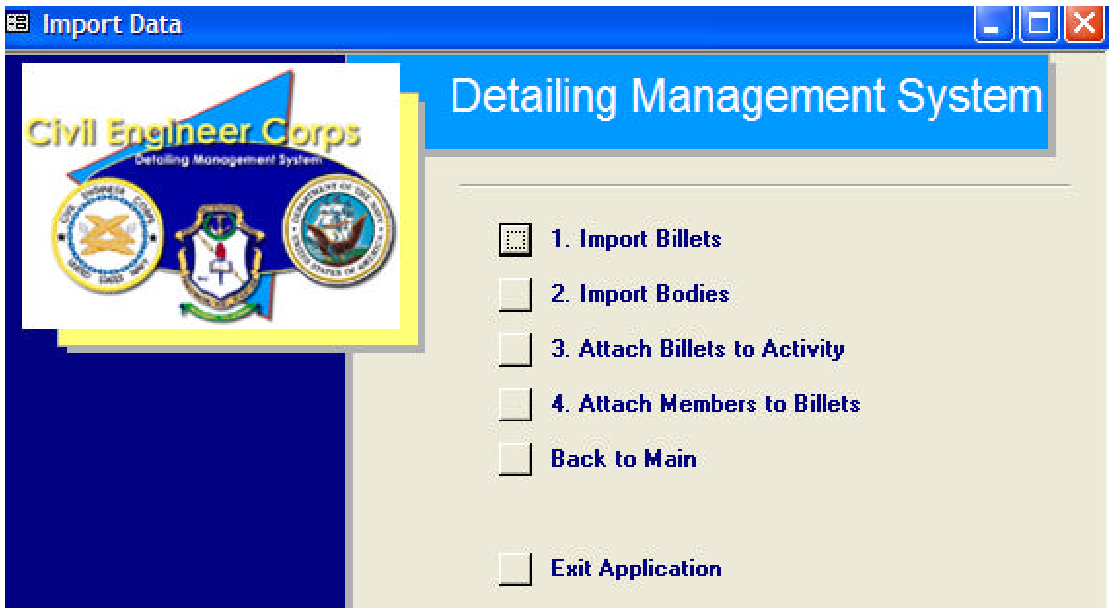
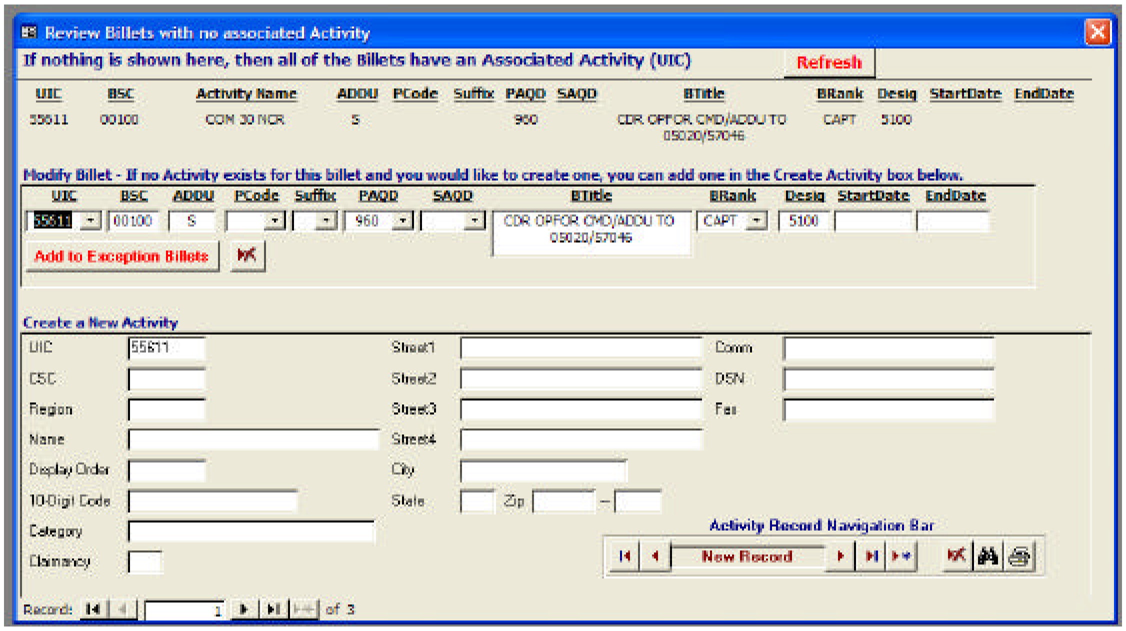

Import Data
Civil Engineer Corps
Detailing Management System
Detailing Management System
1. Import Billets
2. Import Bodies
3. Attach Billets to Activity
4. Attach Members to Billets
Back to Main
Exit Application

Figure 10. Data Import & Cleanup Switchboard

The first step is to import the billets from the delimited text file. This process is execute by a single macro which brings in the data, cleans it up according to predefined rules, and then presents the billet review form shown below in Figure 11. This review form presents the user with a list of imported billets that could not be associated with an existing activity. The user is presented with the option to modify the billet, delete the billet, create a new activity for the billet, or add the billet to the exception activity list, which will allow the user to keep the billet and make a decision on it later. If the user wants to add this billet to an existing activity, the UIC field is a dropdown list that shows all available activities and will not accept any entries that are not already in the list.


Review Billets with no associated Activity

If nothing is shown here, then all of the Billets have an Associated Activity (UIC)

UIC	BSC	Activity Name	ADDU	PCode	Suffix	PAQD	SAQD	BTitle	BRank	Desig	StartDate	EndDate
55611	00100	COM 30 NCR	5			950		CDR OPFOR CMD/ADDU TO 05020/57046	CAPT	5100		

Modify Billet - If no Activity exists for this billet and you would like to create one, you can add one in the Create Activity box below.

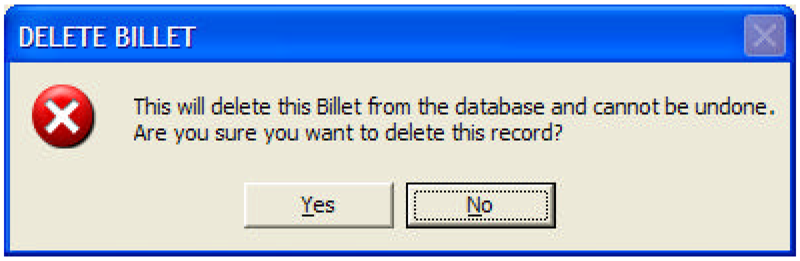
UIC	BSC	ADDU	PCode	Suffix	PAQD	SAQD	BTitle	BRank	Desig	StartDate	EndDate
55611	00100	S			960		CDR OPFOR CMD/ADDU TO 05020/57046	CAPT	5100		

Create a New Activity

UIC	55611	Street1		Comm	
CSC		Street2		DSN	
Region		Street3		Fax	
Name		Street4			
Display Order		City			
10-Digit Code		State	Zip –		
Category					
Claimancy					

Figure 11. Billet Review Form

If the user selects to delete the billet, a warning message is displayed as shown below in Figure 12. This helps prevent the user from accidentally deleting a record. Throughout the entire application, a warning message similar to this is displayed whenever a record is selected to be deleted.


DELETE BILLET

This will delete this Billet from the database and cannot be undone.
Are you sure you want to delete this record?

Figure 12. Delete Warning Message

After completing the billet import and review process, the user will import the delimited bodies file and be presented with a more complex member review form shown below in Figure 13. Only the tabs and column headings are shown in order to protect the CEC members' personal information. See Appendix # for a view of all the forms. This form presents the user seven different views of the imported personnel data. These views are filtered according to the rules established by the Community Management Analyst and are designed to help him sort through the over 1400 members that are imported quickly and efficiently. A summary of the rules is presented below:

- ACC – Checks members where ACC is not equal to 100. Regular CEC members have an ACC equal to 100.
- USNR – Checks members who have a designator of 5105 (USNR) and a rank of LCDR or above. Sometimes a few reserve officers are imported and need to be deleted. The Community Management Analyst is the only one who knows which personnel belong.
- Ultimate EDA – Checks a member's ultimate report date and shows only those members whose report date is prior to the current date.
- Pending Gains – Checks members where Activity = "PNDING NAVY GAIN." These could be members in Officer Candidate School or another program. PERS 4413S will make the decision on what to do with these members (include them, delete them, or reassign them).
- No Orders Loss – Checks members where ultimate activity = "NO ORDERS LOSS." Sometimes the production systems lose particular data for a member regarding their current orders, etc. These records are manually corrected by PERS 4413S.
- Verify Members (filtered) – Displays members where their designators are not like 510x, 653x, 753x. This shows all non-CEC type members.
- All Members – This is the catch all review for the Community Management Analyst. From this screen, he can review all imported member data.
- Members Without Matching Activity – After cleaning up the member data, this screen will show those remaining members who do not have a matching activity (UIC). The user can modify the member's activity (UIC) and billet (BSC), delete the member, move the ultimate duty station information to the current duty station information, or add a new activity and billet for that member.
- Members Without Matching Billets – Once all members have been correctly associated with an activity, this form checks to see if their current billet information matches a billet listed for that activity. The user can fix whatever information is necessary and then, when satisfied, click the button to add the remaining missing billets to the billet table.

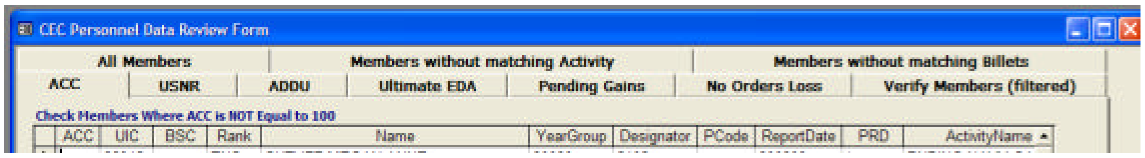

CEC Personnel Data Review Form
All Members
Members without matching Activity
Members without matching Billets
ACC
USNR
ADDU
Ultimate EDA
Pending Gains
No Orders Loss
Verify Members (filtered)
Check Members Where ACC is NOT Equal to 100
ACC
UIC
BSC
Rank
Name
YearGroup
Designator
PCode
ReportDate
PRD
ActivityName

Figure 13. Member Data Cleanup Form

2. CEC Member Management

The heart of the GUI is located in the CEC member management section. The user works off the member management switchboard shown below in Figure 14.

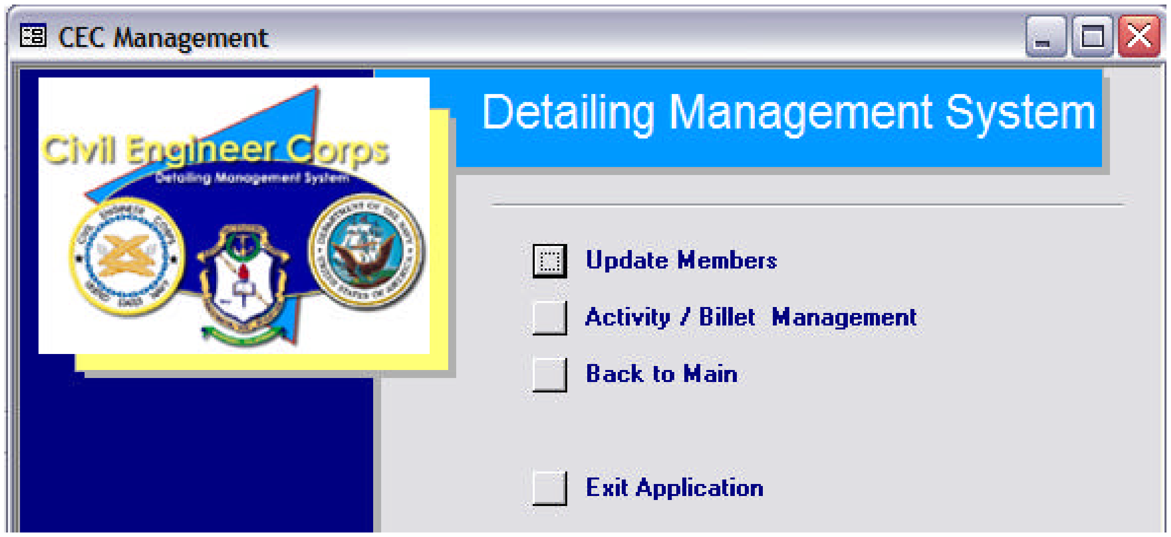
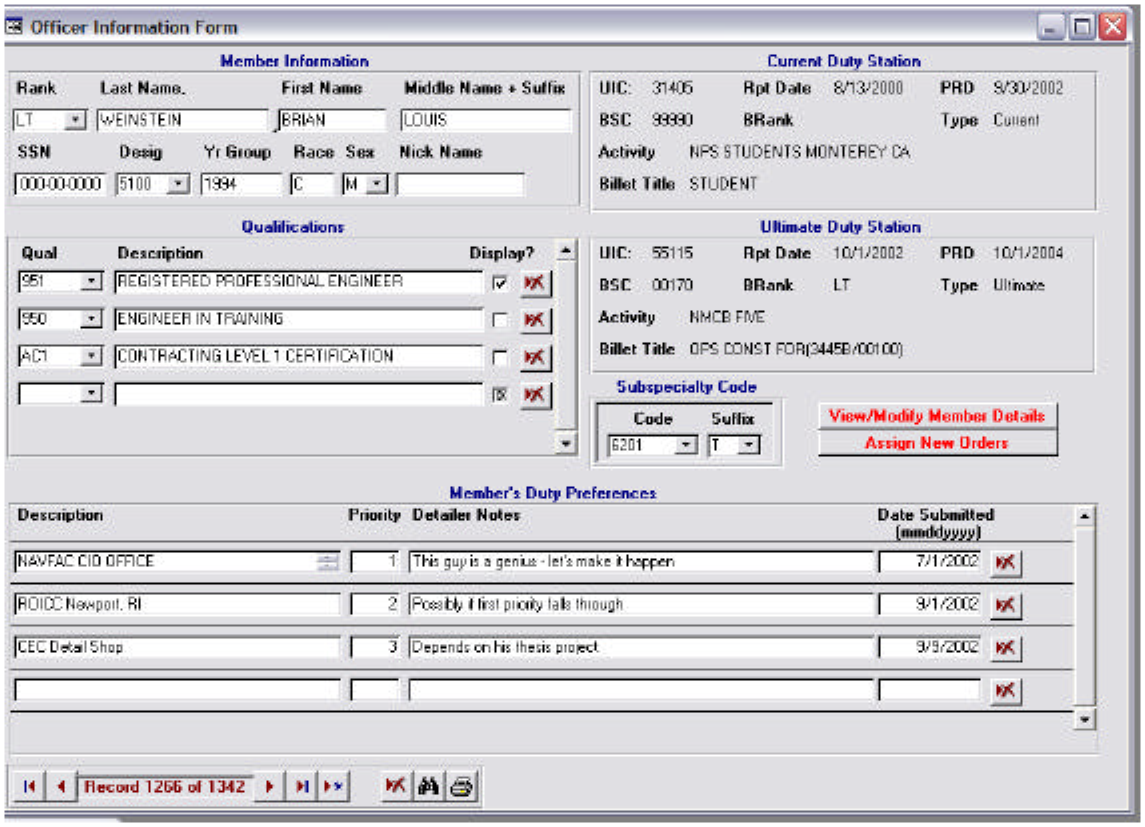

CEC Management
Civil Engineer Corps
Detailing Management System
Detailing Management System
Update Members
Activity / Billet Management
Back to Main
Exit Application

Figure 14. CEC Member Management Switchboard

This switchboard supports two main actions—Update Members and Update Activities and Billets. The user is able to navigate through the application to add, update, and delete billets, activities, and members. These forms also provide functionality for assigning new orders to a member or modifying existing ones.

a. Member Update Form

The member update form is shown below in Figure 15.


Officer Information Form

Member Information

Rank	Last Name	First Name	Middle Name + Suffix
LT	WEINSTEIN	BRIAN	LOUIS

SSN	Desig	Yr Group	Race	Sex	Nick Name
000-00-0000	5100	1994	C	M	

Current Duty Station

UIC:	31405	Rpt Date	8/13/2000	PRD	9/30/2002
BSC	99990	BRank		Type	Current
Activity	NPS STUDENTS MONTEREY CA				
Billet Title	STUDENT				

Qualifications

Qual	Description	Display?
951	REGISTERED PROFESSIONAL ENGINEER	☑
950	ENGINEER IN TRAINING	☐
AC1	CONTRACTING LEVEL 1 CERTIFICATION	☐
		☒

Ultimate Duty Station

UIC:	55115	Rpt Date	10/1/2002	PRD	10/1/2004
BSC	00170	BRank	LT	Type	Ultimate
Activity	NMCB FIVE				
Billet Title	OPS CONST FOR(3445B/00100)				

Subspecialty Code

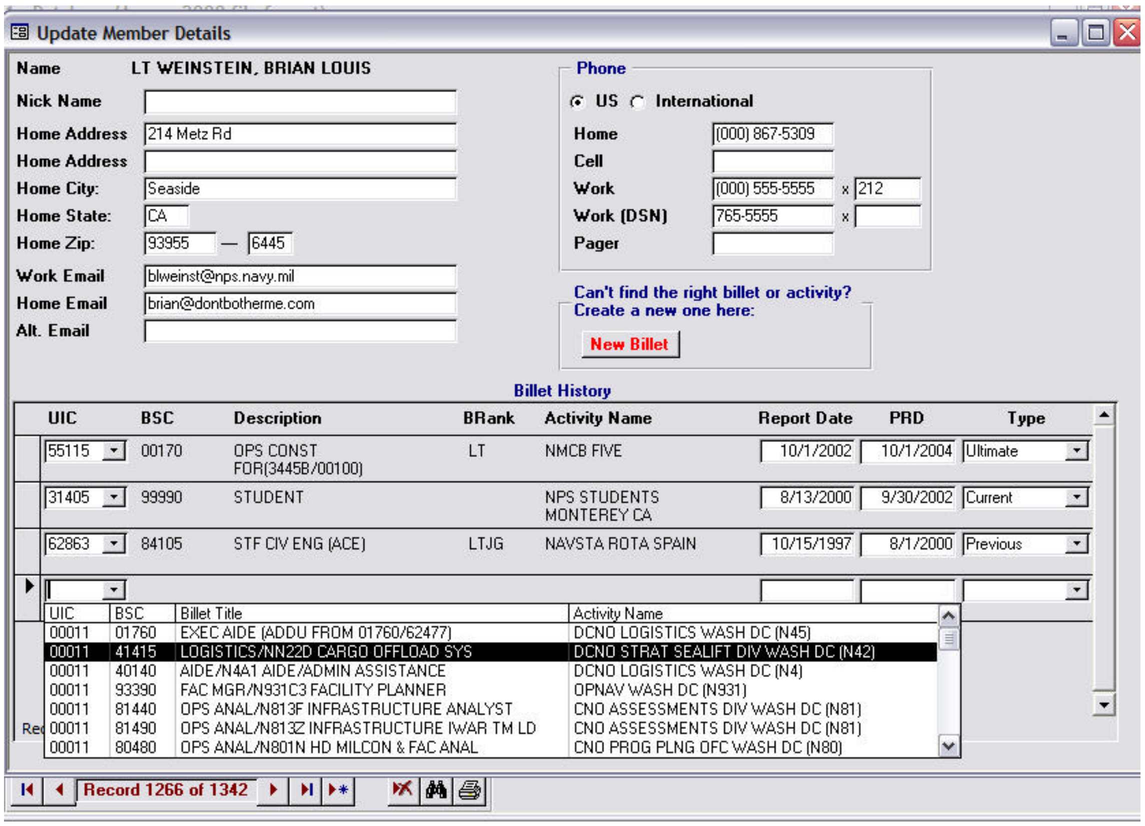
Code	Suffix
6201	T

Member's Duty Preferences

Description	Priority	Detailer Notes	Date Submitted (mmddyyyy)
NAVFAC CID OFFICE	1	This guy is a genius - let's make it happen	7/1/2002
ROICC Newport, RI	2	Possibly if first priority falls through	9/1/2002
CEC Detail Shop	3	Depends on his thesis project	9/9/2002

Figure 15. Member Update Form

From this form, the user can modify a member's name, social security number, designator, subspecialty code (P-Code), year group, race, sex, and nickname. Also located on this form are all the professional qualifications the member has obtained as well as the member's duty preferences. Other information that is listed, but not editable on the screen, is the member's current and ultimate duty stations. Additional personal information as well as the member's billet history is located on the next screen (Figure 16) and accessed by pushing the View/Modify Member Details button.


Update Member Details

Name	LT WEINSTEIN, BRIAN LOUIS
Nick Name	
Home Address	214 Metz Rd
Home Address	
Home City:	Seaside
Home State:	CA
Home Zip:	93955 — 6445
Work Email	blweinst@nps.navy.mil
Home Email	brian@dontbotherme.com
Alt. Email	

Phone

[✓ US] International

Home	(000) 867-5309	
Cell		
Work	(000) 555-5555	x 212
Work (DSN)	765-5555	x
Pager		

Can't find the right billet or activity? Create a new one here:

New Billet

Billet History

UIC	BSC	Description	BRank	Activity Name	Report Date	PRD	Type
55115	00170	OPS CONST FOR(3445B/00100)	LT	NMCB FIVE	10/1/2002	10/1/2004	Ultimate
31405	99990	STUDENT		NPS STUDENTS MONTEREY CA	8/13/2000	9/30/2002	Current
62863	84105	STF CIV ENG (ACE)	LTJG	NAVSTA ROTA SPAIN	10/15/1997	8/1/2000	Previous

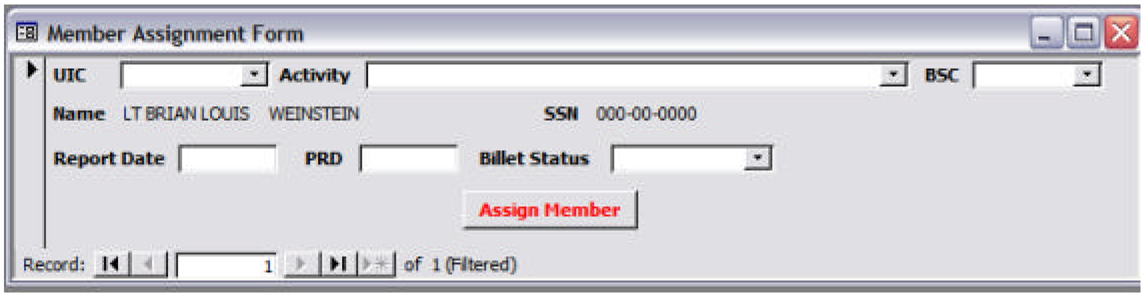
UIC	BSC	Billet Title	Activity Name
00011	01760	EXEC AIDE (ADDU FROM 01760/62477)	DCNO LOGISTICS WASH DC (N45)
00011	41415	LOGISTICS/NN22D CARGO OFFLOAD SYS	DCNO STRAT SEALIFT DIV WASH DC (N42)
00011	40140	AIDE/N4A1 AIDE/ADMIN ASSISTANCE	DCNO LOGISTICS WASH DC (N4)
00011	93390	FAC MGR/N931C3 FACILITY PLANNER	OPNAV WASH DC (N931)
00011	81440	OPS ANAL/N813F INFRASTRUCTURE ANALYST	CNO ASSESSMENTS DIV WASH DC (N81)
00011	81490	OPS ANAL/N813Z INFRASTRUCTURE IWAR TM LD	CNO ASSESSMENTS DIV WASH DC (N81)
00011	80480	OPS ANAL/N801N HD MILCON & FAC ANAL	CNO PROG PLNG OFC WASH DC (N80)

Record 1266 of 1342

Figure 16. Additional Member Details Form

Member information including address, phone numbers, and email can be input or modified on this screen. Additionally, the user can input new duty stations for members (issue orders), delete a duty station, or modify the existing ones. The user selects the UIC drop-down combination box to find the right combination of UIC and BSC for the member. Values for the UIC can also be typed into the combination box to narrow down the search for the correct billet. Billet and activity information cannot be modified from this screen. If the desired billet assignment does not exist, the user must create the billet and/or activity from the provided command button on the form.

An alternative to assigning the member a new billet in Figure 16 is to click the Assign New Orders button from Figure 15. This brings up a small window, shown below in Figure 17, specific to the member on the main form. Selecting the UIC for the new orders will automatically filter the activity list to show only those in that UIC. Once the activity is selected, only billets that are attached to that activity will appear on the BSC drop-down menu. The user completes the form by filling in the appropriate dates and billet status and clicks the Assign Member button to complete the process.


Member Assignment Form

UIC		Activity		BSC	
Name	LT BRIAN LOUIS WEINSTEIN			SSN	000-00-0000
Report Date		PRD		Billet Status	

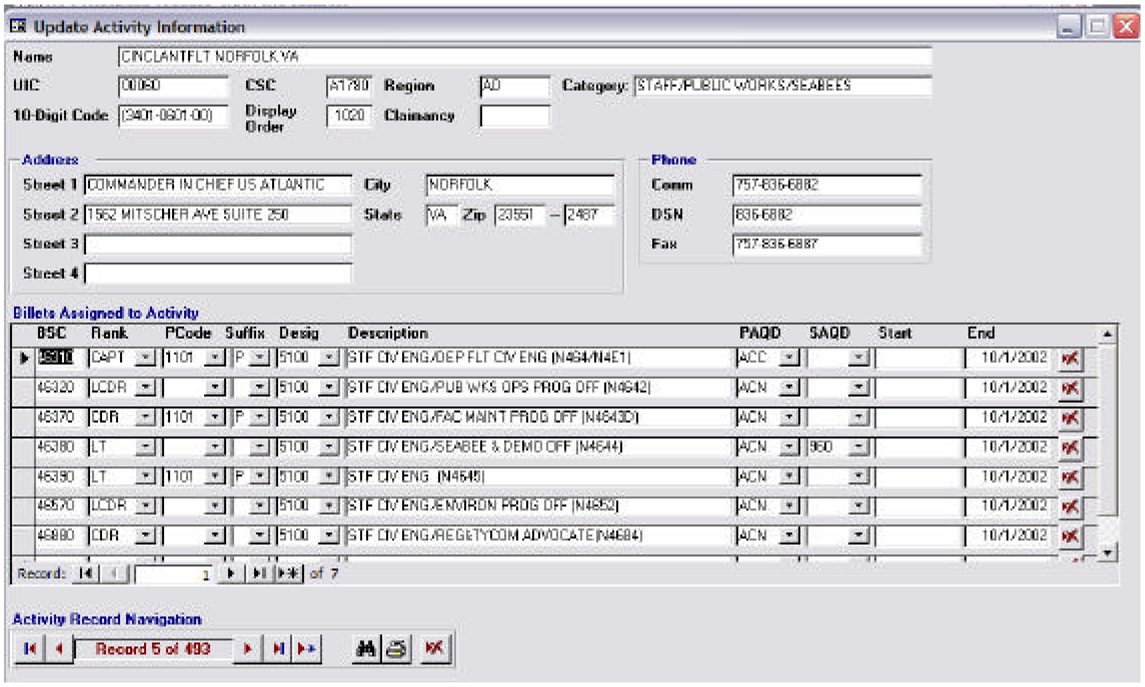
Assign Member

Record: 1 of 1 (Filtered)

Figure 17. New Orders Assignment Form

b. Activity / Billet Update Form

Activity information is not imported from any existing production systems. The ACTIVITY table was populated from the old P-1 database and must be maintained by the Community Management Analyst to ensure that the billets and bodies match up with the correct activity. Activity data is maintained using the Activity Management form shown below in Figure 18.


Update Activity Information

Name	CINCLANTFLT NORFOLK VA						
UIC	00060	CSC	A1780	Region	AD	Category:	STAFF/PUBLIC WORKS/SEABEES
10-Digit Code	(3401-0601-00)	Display Order	1020	Claimancy			

Address

Street 1	COMMANDER IN CHIEF US ATLANTIC	City	NORFOLK
Street 2	1562 MITSCHER AVE SUITE 250	State	VA Zip 23551 – 2487
Street 3			
Street 4			

Phone

Comm	757-836-6882
DSN	836-6882
Fax	757-836-6887

Billets Assigned to Activity

BSC	Rank	PCode	Suffix	Desig	Description	PAQD	SAQD	Start	End
46310	CAPT	1101	P	5100	STF CIV ENG/DEP FLT CIV ENG (N464/N4E1)	ACC			10/1/2002
46320	LCDR			5100	STF CIV ENG/PUB WKS OPS PROG OFF (N4642)	ACN			10/1/2002
46370	CDR	1101	P	5100	STF CIV ENG/FAC MAINT PROG OFF (N4643D)	ACN			10/1/2002
46380	LT			5100	STF CIV ENG/SEABEE & DEMO OFF (N4644)	ACN	960		10/1/2002
46390	LT	1101	P	5100	STF CIV ENG (N4645)	ACN			10/1/2002
46570	LCDR			5100	STF CIV ENG/ENVIRON PROG OFF (N4652)	ACN			10/1/2002
46880	CDR			5100	STF CIV ENG/REGETYCOM ADVOCATE (N4684)	ACN			10/1/2002

Record: 1 of 7

Activity Record Navigation

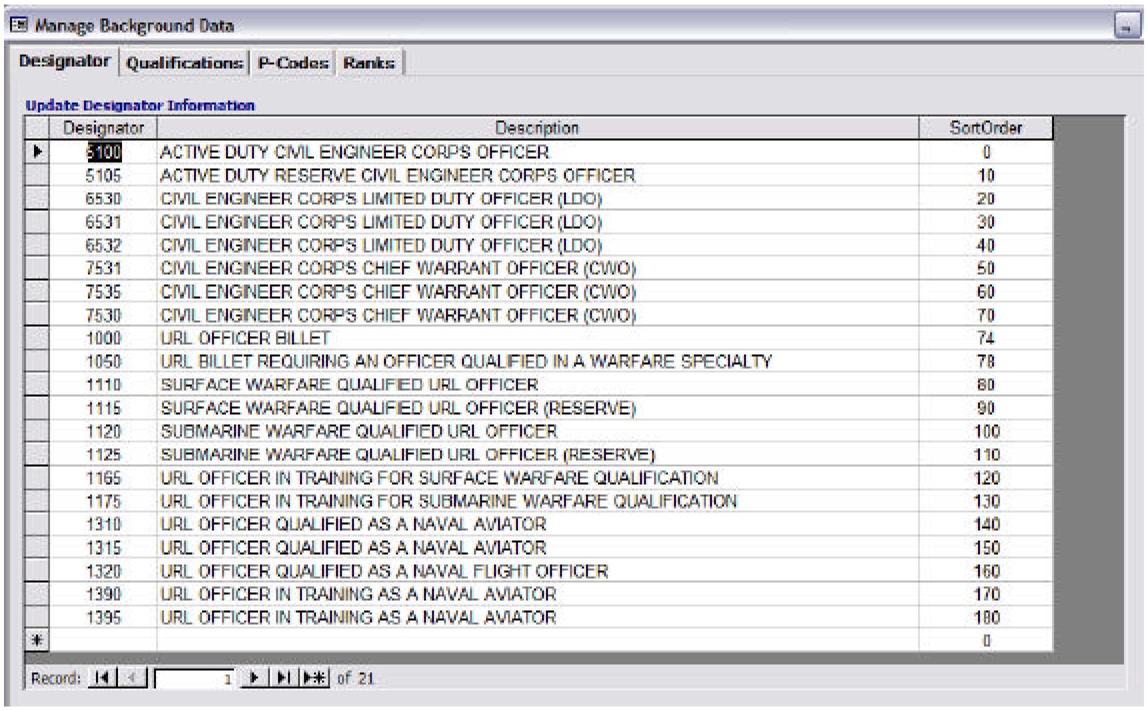
Record 5 of 493

Figure 18. Activity / Billet Management Form

Since billets cannot exist without an activity, it logically follows that the billets should be maintained on the same form as the activity. This makes it easier for the user to search for a billet by first selecting the activity to which it belongs.

3. Background Data Management

Background data is the supporting information in a database for the main entities. It can be in the form of related tables or lookup tables. This data serves two purposes. The first is to ensure data consistency throughout the database. For example, there are several different ways to notate a member's rank. A Lieutenant could be listed as LT, Lt., Lieutenant, O3, O-3, etc. By creating a lookup table of allowed values, the application now limits the end-user to one standardized format. The second purpose is to make data entry easier for the user. With a lookup table, a user can select an allowed value from a drop-down list instead of manually typing in the entry. The problem with having lookup tables is that over time, the allowed values could change. For example, P-Codes are listed in a lookup table and it is conceivable that a new P-Code will be created, another deleted, and others modified. To support these changes, a user must be able to modify the background data. An example of the background data management form is shown below in Figure 19. This form allows the user to modify the four main background data lookup tables.



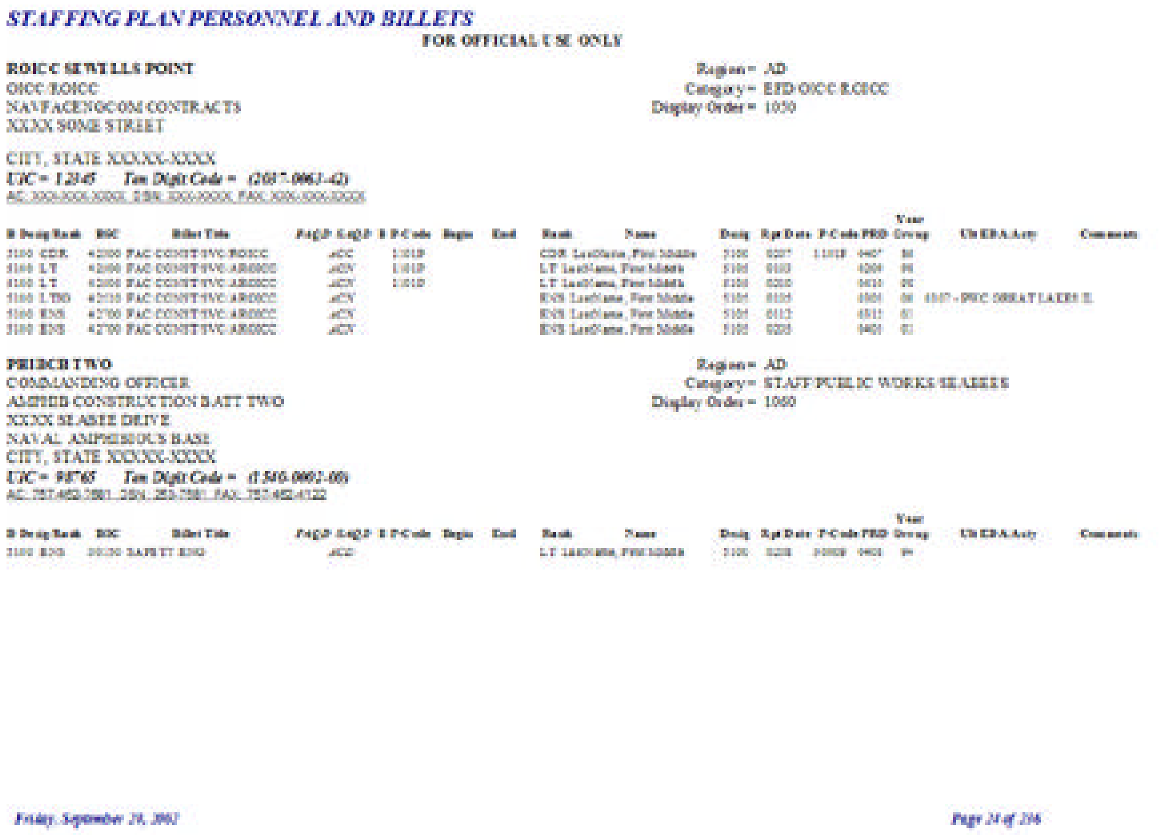
Designator	Description	SortOrder
5100	ACTIVE DUTY CIVIL ENGINEER CORPS OFFICER	0
5105	ACTIVE DUTY RESERVE CIVIL ENGINEER CORPS OFFICER	10
6530	CIVIL ENGINEER CORPS LIMITED DUTY OFFICER (LDO)	20
6531	CIVIL ENGINEER CORPS LIMITED DUTY OFFICER (LDO)	30
6532	CIVIL ENGINEER CORPS LIMITED DUTY OFFICER (LDO)	40
7531	CIVIL ENGINEER CORPS CHIEF WARRANT OFFICER (CWO)	50
7535	CIVIL ENGINEER CORPS CHIEF WARRANT OFFICER (CWO)	60
7530	CIVIL ENGINEER CORPS CHIEF WARRANT OFFICER (CWO)	70
1000	URL OFFICER BILLET	74
1050	URL BILLET REQUIRING AN OFFICER QUALIFIED IN A WARFARE SPECIALTY	78
1110	SURFACE WARFARE QUALIFIED URL OFFICER	80
1115	SURFACE WARFARE QUALIFIED URL OFFICER (RESERVE)	90
1120	SUBMARINE WARFARE QUALIFIED URL OFFICER	100
1125	SUBMARINE WARFARE QUALIFIED URL OFFICER (RESERVE)	110
1165	URL OFFICER IN TRAINING FOR SURFACE WARFARE QUALIFICATION	120
1175	URL OFFICER IN TRAINING FOR SUBMARINE WARFARE QUALIFICATION	130
1310	URL OFFICER QUALIFIED AS A NAVAL AVIATOR	140
1315	URL OFFICER QUALIFIED AS A NAVAL AVIATOR	150
1320	URL OFFICER QUALIFIED AS A NAVAL FLIGHT OFFICER	160
1390	URL OFFICER IN TRAINING AS A NAVAL AVIATOR	170
1395	URL OFFICER IN TRAINING AS A NAVAL AVIATOR	180
		0

Record: 1 of 21

Figure 19. Background Data Management Form

4. Report Viewing

The two main reports for the detailers are the Staffing Plan and the P-1. The Staffing Plan is used to help the detailers determine where CEC members are stationed and where they will be going for their next duty station at a projected date. Typically, the Staffing Plan is generated looking three months in advance, taking the personnel who have ultimate duty stations with reporting dates that fall into that window and placing them in their new duty station for the purposes of the report. The P-1 is generated in the same manner with a projection date applied to ensure the data is valid for several months after the report is published. A sample of the Staffing Plan is shown below in Figure 20. Note that the actual data has been masked to protect the member's privacy and security.



STAFFING PLAN PERSONNEL AND BILLETS

FOR OFFICIAL USE ONLY

ROICC SEWELLS POINT — Region = AD
OICC ROICC — Category = EFD OICC ROICC
NAVFACENGCOM CONTRACTS — Display Order = 1050
XXXX SOME STREET

CITY, STATE XXXXX-XXXX
UIC = 12345 Ten Digit Code = (2017-0063-42)
AC: XXX-XXX-XXXX DSN: XXX-XXXX FAX: XXX-XXX-XXXX

B Desig Rank	BSC	Billet Title	FAQD	SAQD	E P-Code	Begin	End	Rank	Name	Desig	Rpt Date	P-Code	PRD	Year Group	Ult EDA Acty	Comments
5100 CDR	42000	FAC CONST SVC ROICC		ACC	1101P			CDR LastName, First Middle	5100	0207	1101B	0407	86		
5100 LT	42000	FAC CONST SVC AROICC		ACN	1101P			LT LastName, First Middle	5100	0103		0209	98		
5100 LT	42000	FAC CONST SVC AROICC		ACN	1101D			LT LastName, First Middle	5100	0203		0610	00		
5100 LTJG	42010	FAC CONST SVC AROICC		ACN				ENS LastName, First Middle	5105	0105		0501	00	4107 - PWC GREAT LAKES IL	
5100 ENS	42700	FAC CONST SVC AROICC		ACN				ENS LastName, First Middle	5105	0112		0511	01		
5100 ENS	42700	FAC CONST SVC AROICC		ACN				ENS LastName, First Middle	5105	0205		0405	01		

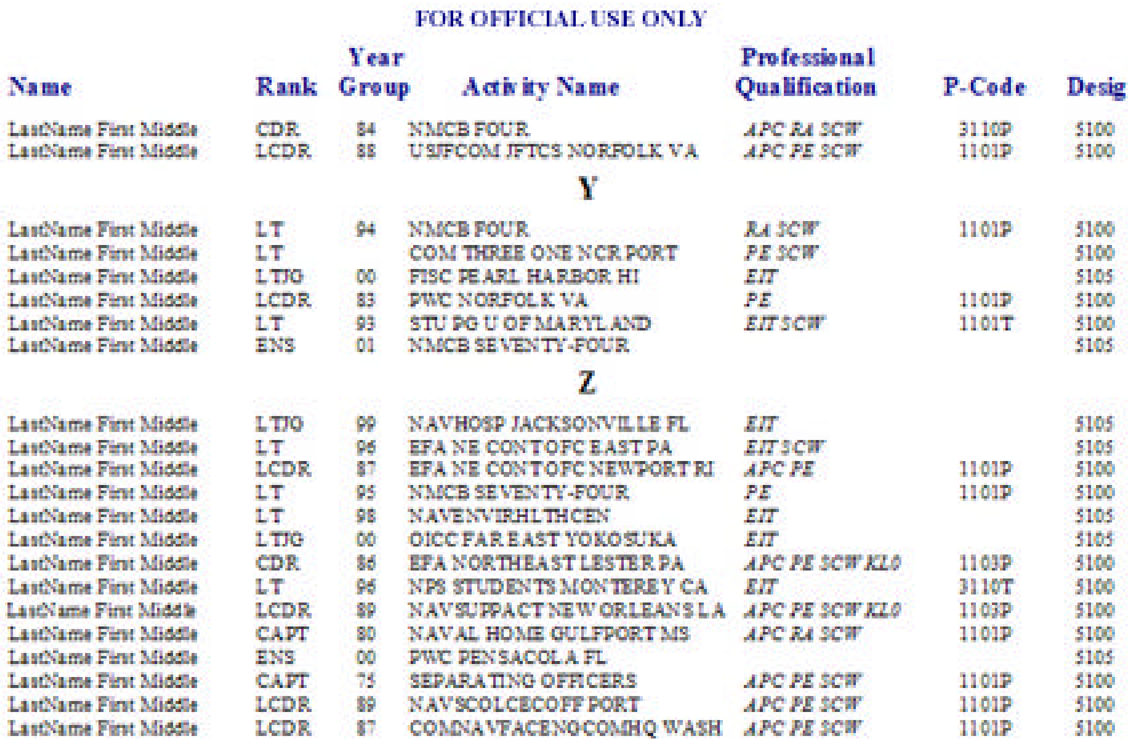
PHIBCB TWO — Region = AD
COMMANDING OFFICER — Category = STAFF PUBLIC WORKS SEABEES
AMPHIB CONSTRUCTION BATT TWO — Display Order = 1060
XXXX SEABEE DRIVE
NAVAL AMPHIBIOUS BASE
CITY, STATE XXXXX-XXXX
UIC = 98765 Ten Digit Code = (1505-0002-00)
AC: 757-462-7881 DSN: 253-7881 FAX: 757-462-4122

B Desig Rank	BSC	Billet Title	FAQD	SAQD	E P-Code	Begin	End	Rank	Name	Desig	Rpt Date	P-Code	PRD	Year Group	Ult EDA Acty	Comments
5100 ENS	20020	SAFETY ENG		ACC				LT LASTNAME, FIRSTNAME	5100	0208	1000B	0405	99		

Friday, September 28, 2007 — *Page 24 of 216*

Figure 20. Sample Staffing Plan

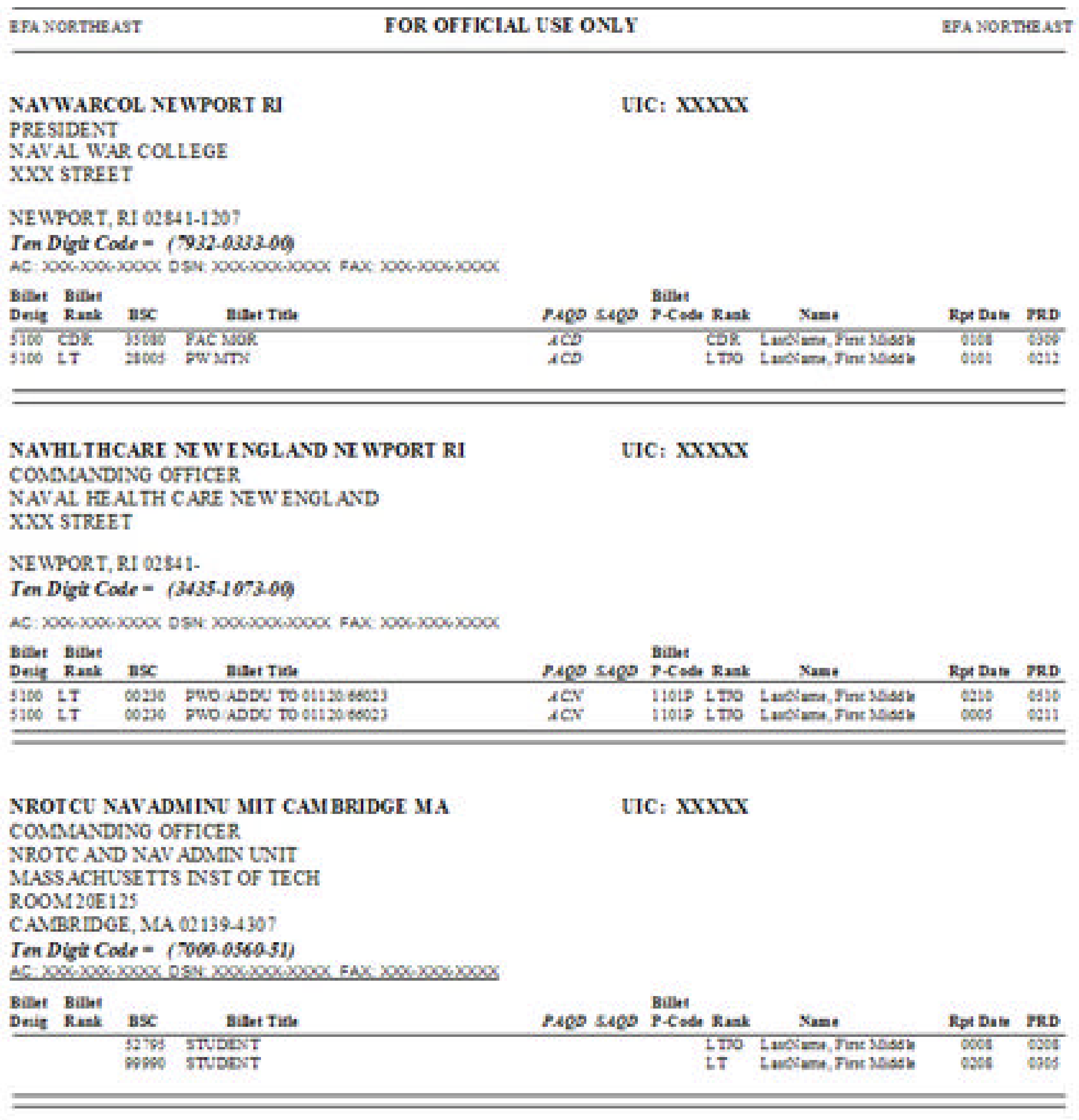
The P-1 has three components—a personnel listing, a detailed command listing with personnel, and a simple command listing. These are displayed in Figures 21-22 below. Again, actual data has been masked to protect the privacy and security of the members.



FOR OFFICIAL USE ONLY

Name	Rank	Year Group	Activity Name	Professional Qualification	P-Code	Desig
LastName First Middle	CDR	84	NMCB FOUR	*APC RA SCW*	3110P	5100
LastName First Middle	LCDR	88	USJFCOM JFTCS NORFOLK VA	*APC PE SCW*	1101P	5100
			Y			
LastName First Middle	LT	94	NMCB FOUR	*RA SCW*	1101P	5100
LastName First Middle	LT		COM THREE ONE NCR PORT	*PE SCW*		5100
LastName First Middle	LTJG	00	FISC PEARL HARBOR HI	*EIT*		5105
LastName First Middle	LCDR	83	PWC NORFOLK VA	*PE*	1101P	5100
LastName First Middle	LT	93	STU PG U OF MARYLAND	*EIT SCW*	1101T	5100
LastName First Middle	ENS	01	NMCB SEVENTY-FOUR			5105
			Z			
LastName First Middle	LTJG	99	NAVHOSP JACKSONVILLE FL	*EIT*		5105
LastName First Middle	LT	96	EFA NE CONTOFC EAST PA	*EIT SCW*		5105
LastName First Middle	LCDR	87	EFA NE CONTOFC NEWPORT RI	*APC PE*	1101P	5100
LastName First Middle	LT	95	NMCB SEVENTY-FOUR	*PE*	1101P	5100
LastName First Middle	LT	98	NAVENVIRHLTHCEN	*EIT*		5105
LastName First Middle	LTJG	00	OICC FAR EAST YOKOSUKA	*EIT*		5105
LastName First Middle	CDR	86	EFA NORTHEAST LESTER PA	*APC PE SCW KL0*	1103P	5100
LastName First Middle	LT	96	NPS STUDENTS MONTEREY CA	*EIT*	3110T	5100
LastName First Middle	LCDR	89	NAVSUPPACT NEW ORLEANS LA	*APC PE SCW KL0*	1103P	5100
LastName First Middle	CAPT	80	NAVAL HOME GULFPORT MS	*APC RA SCW*	1101P	5100
LastName First Middle	ENS	00	PWC PENSACOLA FL			5105
LastName First Middle	CAPT	75	SEPARATING OFFICERS	*APC PE SCW*	1101P	5100
LastName First Middle	LCDR	89	NAVSCOLCECOFF PORT	*APC PE SCW*	1101P	5100
LastName First Middle	LCDR	87	COMNAVFACENGCOMHQ WASH	*APC PE SCW*	1101P	5100

Figure 21. P-1 Personnel Listing



EFA NORTHEAST | **FOR OFFICIAL USE ONLY** | EFA NORTHEAST

NAVWARCOL NEWPORT RI **UIC: XXXXX**
PRESIDENT
NAVAL WAR COLLEGE
XXX STREET

NEWPORT, RI 02841-1207
Ten Digit Code = (7932-0333-00)
AC: XXX-XXX-XXXX DSN: XXX-XXX-XXXX FAX: XXX-XXX-XXXX

Billet Desig	Billet Rank	BSC	Billet Title	PAQD	SAQD	Billet P-Code	Rank	Name	Rpt Date	PRD
5100	CDR	35080	FAC MGR	ACD			CDR	LastName, First Middle	0108	0309
5100	LT	28005	PW MTN	ACD			LTJG	LastName, First Middle	0101	0212

NAVHLTHCARE NEW ENGLAND NEWPORT RI **UIC: XXXXX**
COMMANDING OFFICER
NAVAL HEALTH CARE NEW ENGLAND
XXX STREET

NEWPORT, RI 02841-
Ten Digit Code = (3435-1073-00)

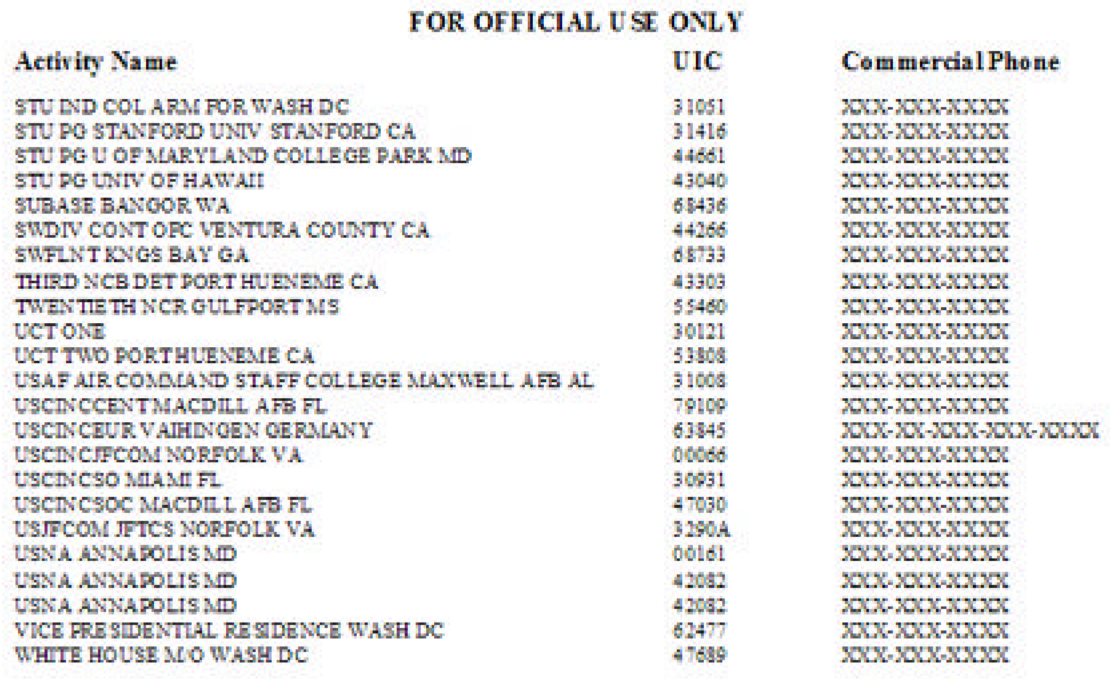
AC: XXX-XXX-XXXX DSN: XXX-XXX-XXXX FAX: XXX-XXX-XXXX

Billet Desig	Billet Rank	BSC	Billet Title	PAQD	SAQD	Billet P-Code	Rank	Name	Rpt Date	PRD
5100	LT	00230	PWO/ADDU TO 01120/66023	ACY		1101P	LTJG	LastName, First Middle	0210	0510
5100	LT	00230	PWO/ADDU TO 01120/66023	ACY		1101P	LTJG	LastName, First Middle	0005	0211

NROTCU NAVADMINU MIT CAMBRIDGE MA **UIC: XXXXX**
COMMANDING OFFICER
NROTC AND NAV ADMIN UNIT
MASSACHUSETTS INST OF TECH
ROOM 20E125
CAMBRIDGE, MA 02139-4307
Ten Digit Code = (7000-0560-51)
AC: XXX-XXX-XXXX DSN: XXX-XXX-XXXX FAX: XXX-XXX-XXXX

Billet Desig	Billet Rank	BSC	Billet Title	PAQD	SAQD	Billet P-Code	Rank	Name	Rpt Date	PRD
		52795	STUDENT				LTJG	LastName, First Middle	0008	0208
		99990	STUDENT				LT	LastName, First Middle	0208	0305

Figure 22. P-1 Billets Sample

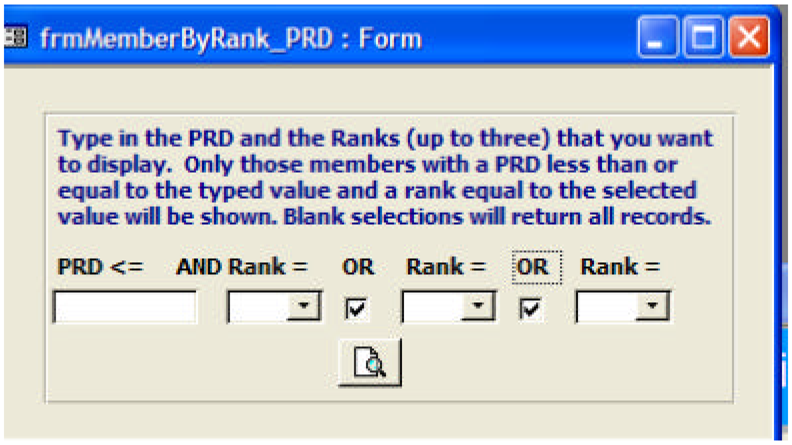


FOR OFFICIAL USE ONLY

Activity Name	UIC	Commercial Phone
STU IND COL ARM FOR WASH DC	31051	XXX-XXX-XXXX
STU PG STANFORD UNIV STANFORD CA	31416	XXX-XXX-XXXX
STU PG U OF MARYLAND COLLEGE PARK MD	44661	XXX-XXX-XXXX
STU PG UNIV OF HAWAII	43040	XXX-XXX-XXXX
SUBASE BANGOR WA	68436	XXX-XXX-XXXX
SWDIV CONT OFC VENTURA COUNTY CA	44266	XXX-XXX-XXXX
SWFLNT KNGS BAY GA	68733	XXX-XXX-XXXX
THIRD NCB DET PORT HUENEME CA	43303	XXX-XXX-XXXX
TWENTIETH NCR GULFPORT MS	55460	XXX-XXX-XXXX
UCT ONE	30121	XXX-XXX-XXXX
UCT TWO PORT HUENEME CA	53808	XXX-XXX-XXXX
USAF AIR COMMAND STAFF COLLEGE MAXWELL AFB AL	31008	XXX-XXX-XXXX
USCINCCENT MACDILL AFB FL	79109	XXX-XXX-XXXX
USCINCEUR VAIHINGEN GERMANY	63845	XXX-XX-XXX-XXX-XXXX
USCINCJFCOM NORFOLK VA	00066	XXX-XXX-XXXX
USCINCSO MIAMI FL	30931	XXX-XXX-XXXX
USCINCSOC MACDILL AFB FL	47030	XXX-XXX-XXXX
USJFCOM JFTCS NORFOLK VA	3290A	XXX-XXX-XXXX
USNA ANNAPOLIS MD	00161	XXX-XXX-XXXX
USNA ANNAPOLIS MD	42082	XXX-XXX-XXXX
USNA ANNAPOLIS MD	42082	XXX-XXX-XXXX
VICE PRESIDENTIAL RESIDENCE WASH DC	62477	XXX-XXX-XXXX
WHITE HOUSE M/O WASH DC	47689	XXX-XXX-XXXX

Figure 23. P-1 Command Listing Sample

An additional report that was requested and created for the users is a report showing members with a particular rank and PRD. The users are presented with a form in which they will fill out their desired PRD and what ranks they would like to see in the report. The form allows for a maximum of three ranks to filter for because a three rank range is typically all one particular detailer will care to examine. The form and report is shown below in Figures 24 and 25.

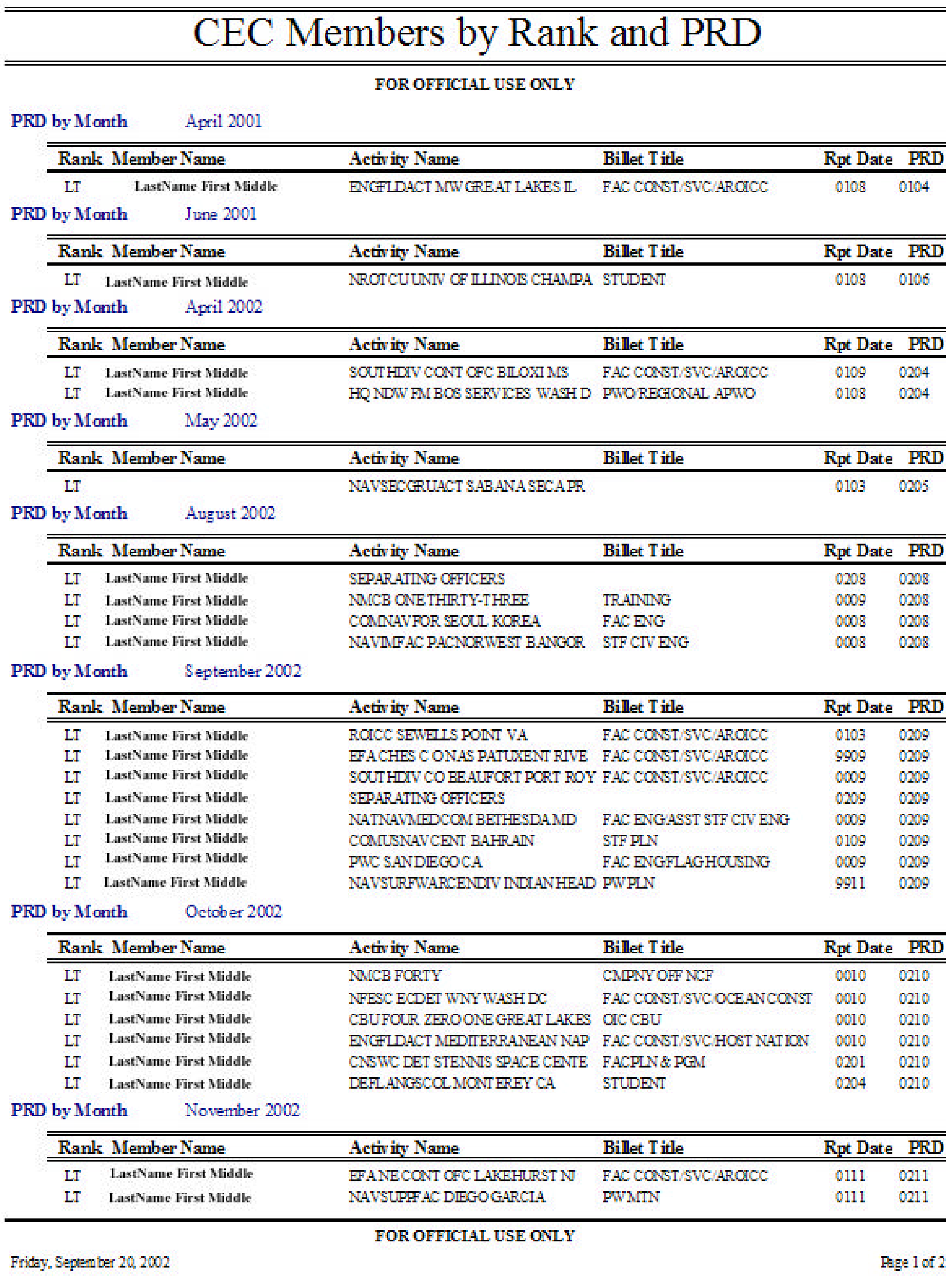


frmMemberByRank_PRD : Form

Type in the PRD and the Ranks (up to three) that you want to display. Only those members with a PRD less than or equal to the typed value and a rank equal to the selected value will be shown. Blank selections will return all records.

PRD <=	AND Rank =	OR	Rank =	OR	Rank =
		✓		✓	

Figure 24. Filter Form for Member By Rank and PRD Report



CEC Members by Rank and PRD

FOR OFFICIAL USE ONLY

PRD by Month April 2001

Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT	LastName First Middle	ENGFLDACT MW GREAT LAKES IL	FAC CONST/SVC/AROICC	0108	0104

PRD by Month June 2001

Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT	LastName First Middle	NROTCU UNIV OF ILLINOIS CHAMPA	STUDENT	0108	0106

PRD by Month April 2002

Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT	LastName First Middle	SOUTHDIV CONT OFC BILOXI MS	FAC CONST/SVC/AROICC	0109	0204
LT	LastName First Middle	HQ NDW FM BOS SERVICES WASH D	PWO REGIONAL APWO	0108	0204

PRD by Month May 2002

Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT		NAVSECGRUACT SABANA SECA PR		0103	0205

PRD by Month August 2002

Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT	LastName First Middle	SEPARATING OFFICERS		0208	0208
LT	LastName First Middle	NMCB ONE THIRTY-THREE	TRAINING	0009	0208
LT	LastName First Middle	COMNAVFOR SEOUL KOREA	FAC ENG	0008	0208
LT	LastName First Middle	NAVIMFAC PACNORWEST BANGOR	STF CIV ENG	0008	0208

PRD by Month September 2002

Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT	LastName First Middle	ROICC SEWELLS POINT VA	FAC CONST/SVC/AROICC	0103	0209
LT	LastName First Middle	EFA CHES C O NAS PATUXENT RIVE	FAC CONST/SVC/AROICC	9909	0209
LT	LastName First Middle	SOUTHDIV CO BEAUFORT PORT ROY	FAC CONST/SVC/AROICC	0009	0209
LT	LastName First Middle	SEPARATING OFFICERS		0209	0209
LT	LastName First Middle	NATNAVMEDCOM BETHESDA MD	FAC ENG/ASST STF CIV ENG	0009	0209
LT	LastName First Middle	COMUSNAVCENT BAHRAIN	STF PLN	0109	0209
LT	LastName First Middle	PWC SAN DIEGO CA	FAC ENG/FLAG HOUSING	0009	0209
LT	LastName First Middle	NAVSURFWARCENDIV INDIANHEAD	PW PLN	9911	0209

PRD by Month October 2002

Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT	LastName First Middle	NMCB FORTY	CMPNY OFF NCF	0010	0210
LT	LastName First Middle	NFESC ECDET WNY WASH DC	FAC CONST/SVC/OCEAN CONST	0010	0210
LT	LastName First Middle	CBU FOUR ZERO ONE GREAT LAKES	OIC CBU	0010	0210
LT	LastName First Middle	ENGFLDACT MEDITERRANEAN NAP	FAC CONST/SVC/HOST NATION	0010	0210
LT	LastName First Middle	CNSWC DET STENNIS SPACE CENTE	FAC PLN & PGM	0201	0210
LT	LastName First Middle	DEFLANGSCOL MONTEREY CA	STUDENT	0204	0210

PRD by Month November 2002

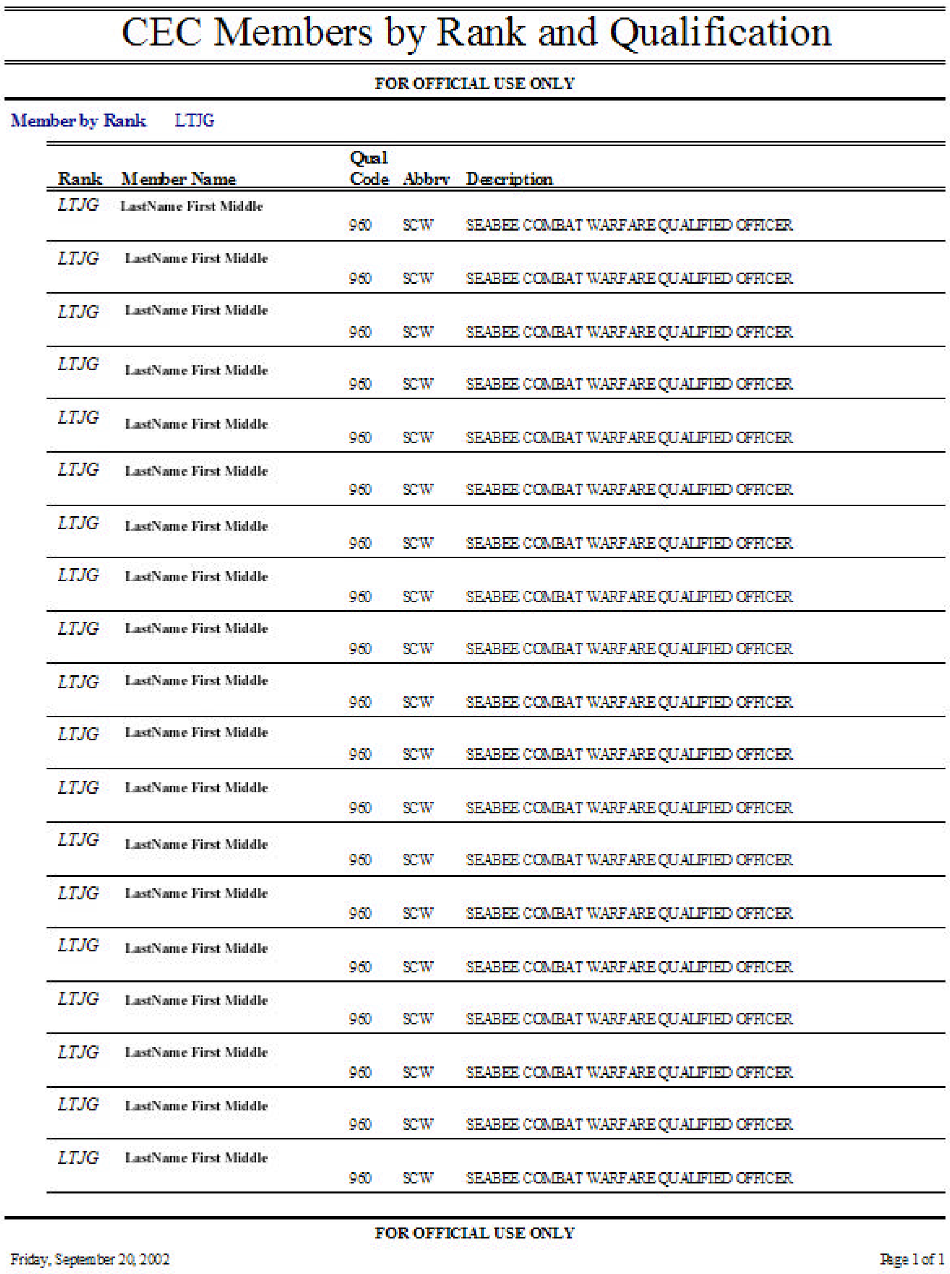
Rank	Member Name	Activity Name	Billet Title	Rpt Date	PRD
LT	LastName First Middle	EFA NE CONT OFC LAKEHURST NJ	FAC CONST/SVC/AROICC	0111	0211
LT	LastName First Middle	NAVSUPPFAC DIEGO GARCIA	PW MTN	0111	0211

FOR OFFICIAL USE ONLY

Friday, September 20, 2002 Page 1 of 2

Figure 25. CEC Members with Rank = LT and PRD <= 12/31/2002

Another useful report is one showing CEC Members who have a particular qualification and rank. The users can select up to three ranks and one qualification in a form similar that in Figure 24.



CEC Members by Rank and Qualification

FOR OFFICIAL USE ONLY

Member by Rank LTJG

Rank	Member Name	Qual Code	Abbrv	Description
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER
LTJG	LastName First Middle	960	SCW	SEABEE COMBAT WARFARE QUALIFIED OFFICER

FOR OFFICIAL USE ONLY

Friday, September 20, 2002 Page 1 of 1

Figure 26. Members by Rank and Qualification

D. PERSONALIZATION & CUSTOMIZATION

The database application has one primary user (CEC Community Management Analyst) and four secondary users (CEC detailers). The application interface shown in the section above is primarily designed for the Community Management Analyst to help him prepare the Staffing Plan and the P-1. Other reports and queries are done in anticipation of the CEC detailers' needs, but further study and analysis is required to tailor the application for their specific needs.

E. TRAINING PLAN

Converting to a new system requires that the end-users be trained and provided with documentation that guides them through using the new system. There are many philosophies on how training should be conducted. Some prefer one-on-one training and others prefer group training. Advocates of group training say that it is a better use of the trainer's time and the students will actually learn more from each other than from the instructors. Group training can take advantage of the ripple effect of education. The first group of trainees can then train additional groups (train the trainer). Proponents for one-on-one training say that nothing beats the personalized experience of having the instructor sit right at the students' desk and help them through the program as they try to execute their job with the new system. The work environment is familiar for the user and situations encountered are real instead of theoretical.

Based on my personal experiences with implementing new software programs, I believe a combination of these two techniques is the best solution for the users. The training will be conducted in three phases—user orientation, group training, and individual training.

1. User orientation

In this phase, the users will be presented with the user manual from the previous section in this chapter. The users will be encouraged to read the manual thoroughly and then be given a copy of the application for their free use. They will be encouraged to open up the application and play around with it just to familiarize themselves with the interface and its features. Some users will quickly learn the application and be able to produce the information they need right away. Others will require more instruction to understand the application completely. Since they will be using a local copy of the application, they will not be concerned with making any data mistakes. This “first look” will prepare the users for the group training session in the next phase and should generate questions about its use.

2. Group Training

In this phase, the users will be gathered for a formal introduction to the application. The application will be demonstrated to them and they will follow along on their own machines during the process. Questions generated from the first phase as well as the group training will be addressed. The intent of this session is to provide an overview of the application and everything that it can do for the users. Detailed questions that require more time to answer will be deferred to the individual sessions in the third phase.

3. Individual Training

The final phase of training will be conducted at each individual user’s desktop. After having sufficient time to work with the program individually, the user will have generated several questions on features that are not working in the expected manner, desired features, and basic operations. The intent of the individual training is to address any questions or issues the user may have and to focus more in depth on that particular user’s job and how the application can support the user. The trainer will sit with the users as they perform some of their tasks with the application and answer any questions as well provide advice on the use of the application.

The goal of this training plan is to make the end-users as comfortable with the application as possible. The more comfortable the users are with the system, the more likely it is that they will use it and demand similar functionality in the production systems.

F. PROTOTYPE

Traditionally, physical design has been a paper-and-pencil process. The designers would draw pictures that depicted the layout or structure of outputs, inputs, and the flow of the application. This approach is prone to errors and omissions, which can cause problems in the development of the application. Modern approaches are turning to prototyping. A prototype is a first full-scale and usually functional form of an application. The prototyping approach is an iterative process involving a close working relationship between the designer and users. (Whitten 1998, pg. 315) Prototyping has several advantages and disadvantages which were discussed in Chapter II.

The initial prototypes for this project focused mainly on the data import from the delimited text files. Working closely with the Community Management Analyst, the data was mapped into the new schema and tested to ensure that the automated cleanup routines were working properly and all data was brought in correctly. This process required several iterations before the import portion was finalized. Without the close communication with the Community Management Analyst, the import data functions would have missed the mark considerably. The corporate knowledge held by PERS 4413S was vital to the success of the data import and cleanup routines.

After the data was correctly imported, the focus turned to managing the data within the database and pulling the information out in a logical manner. Since most of the data management was straightforward, the prototype iteration for this portion went quickly. After a few adjustments from the initial design, the requirements for the data management interface were met.

The final portion of the prototype is the generated reports, specifically the Staffing Plan and P-1. Because of the partially normalized data model, some portions of these reports could not be made exactly as their predecessors. All the information is on the reports but in a slightly different format. The reports, as shown in the GUI section of this chapter, are in their final state based on feedback from the users. Whether the CEC detail shop will use these reports is still being decided. The format of the Staffing Plan has been approved, but the format of the P-1 is still being considered. Since this publication goes to every CEC officer, there are larger considerations on how the report is displayed. All other reports have been reviewed and modified according to the expressed desires of the detailers.

G. BETA TESTING OF THE APPLICATION

The beta test of a system, sometimes referred to as validation testing, runs the system in a live environment using real data. This validation tests for a number of items. (Whitten 1998, pg. 568)

- Systems performance – Does the application respond in a timely manner to processing requests?
- Systems performance at peak workload conditions – How does the system respond when all users are accessing the application at the same time?
- Human engineering test – Is the application as easy to learn and use as expected? Are there enhancements that could be made?
- Methods and procedures test – During the test, do the methods and procedures in the application support the users' requirements?
- Backup and recovery testing – Are procedures in place for backup and restoration of the system in case of a catastrophic data loss? What is the required response time and actual response time during a simulated disaster?

Due to scheduling conflicts, the beta test of this system has not occurred as of the time of this writing. Although not desirable and barring any major problems, the beta test and implementation may be conducted in one trip for several reasons including time constraints and funding. To minimize the risks of conducting a beta test and implementation at the same time, thorough alpha testing of the application has been completed with all discovered problems fixed. Alpha testing, sometimes referred to as verification testing, runs the system in a simulated environment using simulated data. Since the application is relatively small, the beta test will occur on the first day and the results will be evaluated for the criteria outlined above. If there are no major problems, the implementation and training will occur starting on the next day. If problems do arise which require more time to address, the implementation will be pushed back to allow time to fix any problem areas of the application.

H. EVALUATION OF ENTIRE SYSTEM

The final step prior to implementation of the system is to ensure that all the pieces necessary for the entire system are installed and working properly. For the implementation of a Microsoft Access application, some of these steps are not as crucial as they would be if we were dealing with a large enterprise system.

1. Installation of DBMS

Access comes with its own DBMS integrated into the application itself, so there is no requirement to install the DBMS separately. The data and procedures will be placed on the centralized file server that is shared only by the CEC detail shop. Each user will need an ODBC (Open Database Connectivity) client installed on their computer that links to the central database.

2. Create the Database

Since Access integrates the DBMS with the database, there is no need to create the database separately. Other products such as Oracle or Microsoft SQL Server would require a separate installation of the DBMS followed by the creation of the database, which would be run from the DBMS.

3. Load and Convert the Data

Once the database and clients are installed, the end-user must run the procedures outlined in Chapter V, Section C. When the end-user has been satisfied that all the data has been loaded and converted, the application is ready to be tested and tuned. In larger scale systems, the database programmer or administrator would typically handle the loading and converting of the data. For this application, the data import has been partially automated for the users to perform themselves.

4. Test and Fine-tune the Database

With the data loaded and ready for use, the users and database developers must now put the application through its paces to find any areas where there are performance or data problems. These problems will be evaluated and potential solutions will be proposed for implementation. Once solutions have been agreed upon, the database will be modified to accommodate the changes and then tested again to ensure the changes achieve the desired results.

5. Evaluate the Database with the Application

In larger enterprise systems, the database and application are typically separate components of the entire system. The interaction between these two components is critical to the success of the project. If the database response time is too slow or the application is poorly configured, the users will become frustrated with the program and may abandon it for their previous system. When using an Access solution, some of the same issues can also arise on a smaller scale. The GUI front-end can have the same problems interacting with the data as larger systems even though the application and database are more tightly integrated. A full system test of all procedures and methods helps to bring out any of these potential performance and data issues. A system test was conducted thoroughly during the alpha testing phase with no major problems found. Some minor problems in the user interface were found and repaired during this phase. Another system test will be conducted during the beta testing phase.

After all these steps have been completed to the satisfaction of both the end-users as well as the system designers, the application is ready to be implemented in a live data environment. There are several different strategies to consider when implementing a new system that replaces an older one.

I. GO–LIVE IMPLEMENTATION

The go-live implementation of any new system can be a nervous time for all parties involved. Several strategies are commonly used to make the transition as smooth as possible for the end-users as well as the support staff that will be maintaining the application. These strategies include, but are not limited to, the abrupt cutover and parallel conversion.

1. Abrupt Cutover

Sometimes referred to as the cold-turkey method, this strategy picks a certain date (usually a date that coincides with a significant milestone for the organization such as the end of a fiscal year) to shut off the old system and implement the new system on the same day. Users will no longer have the option to use the old system and all inputs will be done through the new system. There is inherent risk in this method in that the new system could have some latent flaw that does not become apparent until the system has been in use for a business period. With no backup system, the organization must work to figure out the solution for the flaw. On the other hand, the transition costs are minimal, as both systems do not have to be supported at the same time.

2. Parallel Conversion

Under this approach, both the old and the new systems are operated for a specified time. This is done to ensure that any major problems that are discovered are solved prior to dropping the old system. The final cutover may be abrupt or gradual, as portions of the new system are deemed adequate. This strategy minimizes the risk of the new system causing irreparable damages to the business; however, it also means the cost of the transition will be significant because both systems must be maintained. System performance could also be an issue if both systems are competing for the same computer resources.

3. Transition selection

For this project, I recommend the abrupt cutover for several reasons. This system is not being used for official production data but rather being used as an information delivery tool, which pulls the data from existing production systems. The risk of damage due to a flaw in the application is minimal as the data can be imported again at any time using either system. One inherent risk in using the parallel transition that I have experienced in the past is how the end-users react to initial problems with the new system. Many users who get quickly frustrated with a new system because it is not like their old comfortable system will quickly stop inputting information in the new system and go back to their comfort zone in the old system. If the old system is not available to go back to, the users are forced to learn how to use the new program effectively.

In the end, since this program is not a mandatory production system that must be used the decision to use the system will be up to each individual user. My goal for the implementation is to get the application in place running efficiently and to train the users in its proper use. Once in place, the decision on how and when to use the system is ultimately up to the users.

J. IMPLEMENTATION SUMMARY

In order to support a multi-user environment, the application was split into two parts. The first part, which contains the data, queries, and functions, will reside on a shared drive in the central file server accessible to the CEC detail shop. The other portion contains the GUI, which will reside on each client's computer. In this manner, all users can share the same data while running the application from their desktop, minimizing the load on the network by performing most functions on the server as opposed to the client.

Considerable effort was placed in the creation of an easy to use graphical user interface for the users. Each form is navigated to from a central switchboard that steps the user through the process of data import and cleansing as well as data maintenance. The forms are designed for ease of use by providing drop down tables wherever possible and a logical display of the information.

Extensive testing has been completed at the development site. Additionally, the key user has tested the application at the local site, providing feedback on functionality, accuracy, and errors. All known errors discovered in the testing phases have been corrected, and the application is ready for a full beta test followed by a complete implementation at the client's location. User manuals and training will be provided during the implementation phase of the application. The extent to which the application will be used and how it will be implemented is largely up to the users. The main factor in this decision is how easy it is to import and cleanse the data. An easier process will result in more use than a more time consuming one.

VI. DATA IMPORT, CLEANSING & SYSTEMS MANAGEMENT

A. DATA IMPORT AND CLEANSING

The amount of effort to clean up this data is tremendous and cannot be easily solved. I spent weeks trying to take the legacy data, import it into Access, and clean it up with some automated routines. My efforts were slightly useful at best, as it sped up the process of data cleansing, but the main problem of validation is still there. The rules that govern what changes need to be made to the data are very dynamic and change on a regular basis. Rules that are good today could be outdated and useless two months from now. The result of this thesis is a semi-automated data import and cleansing mechanism. A process flow chart for the data cleansing is shown below in Figure 27. Once the data from the two files has been imported and cleaned up as much as possible, the next step is to attempt to relate them to each other. The Billets and Bodies files were imported and cleaned up into the temporary tables, tmpBILLET and tmpMEMBER, respectively.

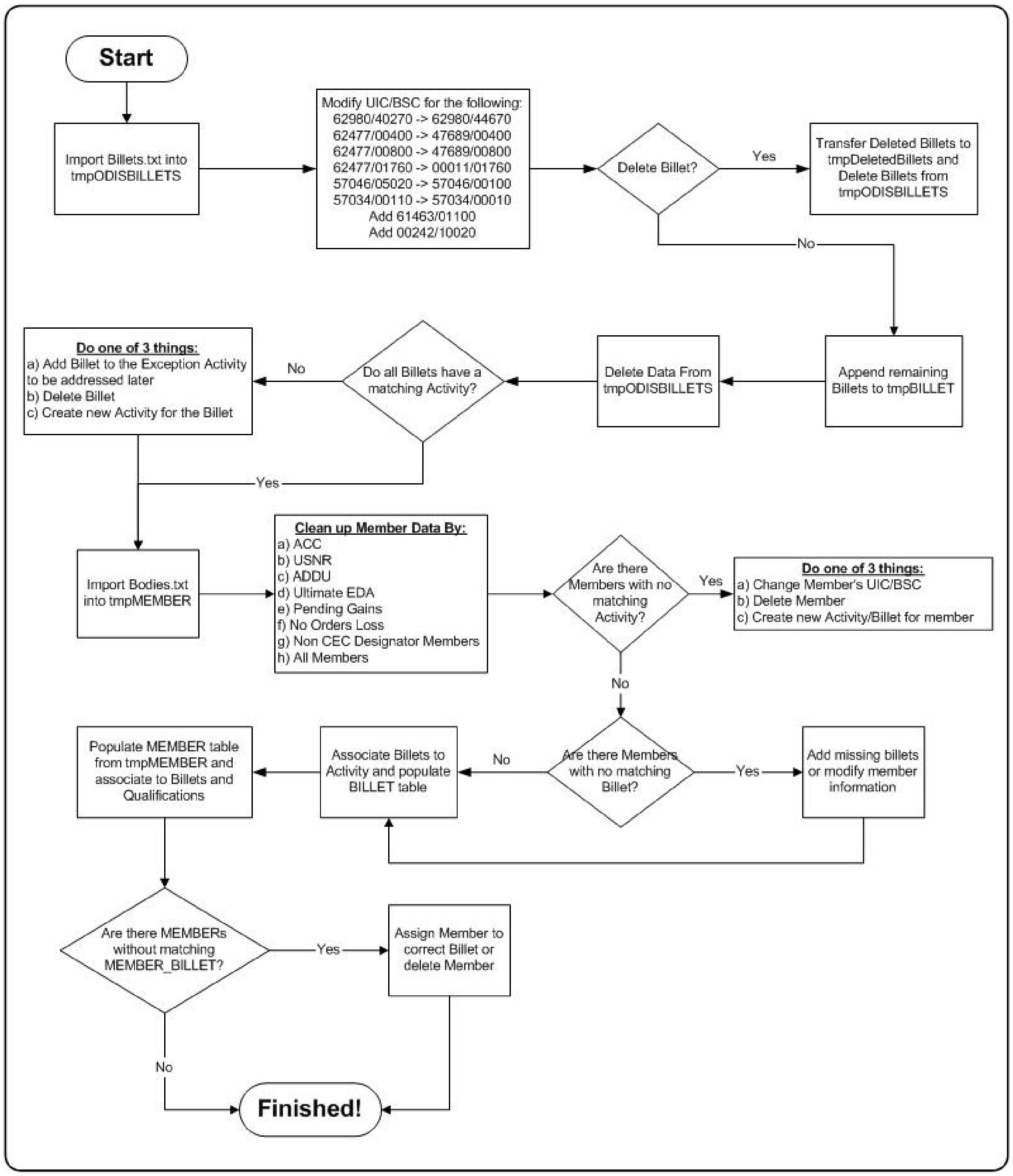

Start
Import Billets.txt into tmpODISBILLETS
Modify UIC/BSC for the following:
62980/40270 -> 62980/44670
62477/00400 -> 47689/00400
62477/00800 -> 47689/00800
62477/01760 -> 00011/01760
57046/05020 -> 57046/00100
57034/00110 -> 57034/00010
Add 61463/01100
Add 00242/10020
Delete Billet?
Yes
Transfer Deleted Billets to tmpDeletedBillets and Delete Billets from tmpODISBILLETS
No
Append remaining Billets to tmpBILLET
Delete Data From tmpODISBILLETS
Do all Billets have a matching Activity?
No
Do one of 3 things:
a) Add Billet to the Exception Activity to be addressed later
b) Delete Billet
c) Create new Activity for the Billet
Yes
Import Bodies.txt into tmpMEMBER
Clean up Member Data By:
a) ACC
b) USNR
c) ADDU
d) Ultimate EDA
e) Pending Gains
f) No Orders Loss
g) Non CEC Designator Members
h) All Members
Are there Members with no matching Activity?
Yes
Do one of 3 things:
a) Change Member's UIC/BSC
b) Delete Member
c) Create new Activity/Billet for member
No
Are there Members with no matching Billet?
Yes
Add missing billets or modify member information
No
Associate Billets to Activity and populate BILLET table
Populate MEMBER table from tmpMEMBER and associate to Billets and Qualifications
Are there MEMBERs without matching MEMBER_BILLET?
Yes
Assign Member to correct Billet or delete Member
No
Finished!

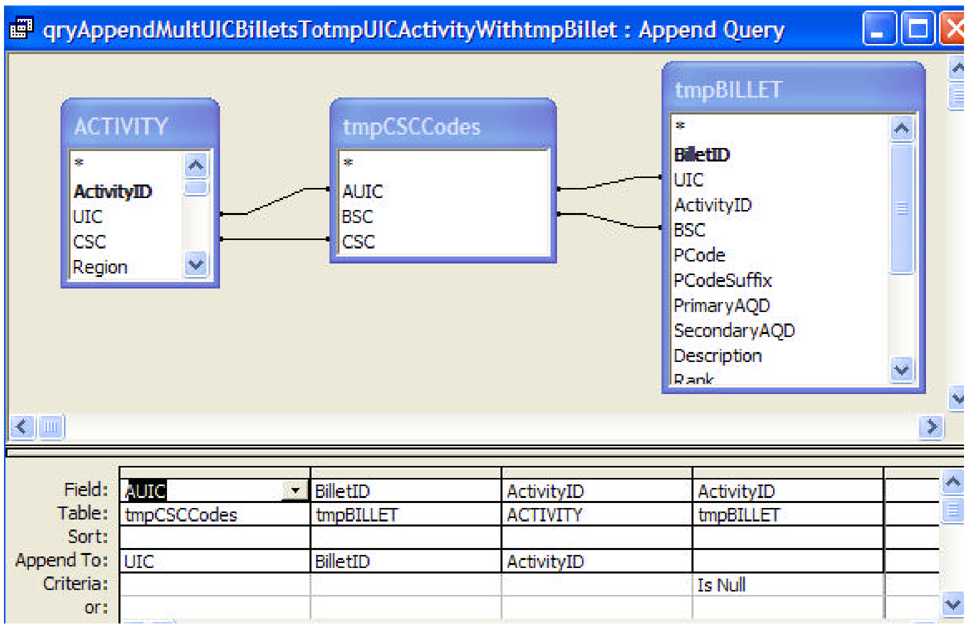
Figure 27. Data Import And Clean-Up Process

1. Relating Billets to Activities

At issue is the definition of an activity. The CEC Detail shop defines an activity differently than NAVPERSCOM. Naval Personnel Command identifies an activity by its Unit Identification Code (UIC). The CEC Detail shop further breaks down these activities into sub-activities. That means for the CEC, several activities can have the same UIC. The legacy system data contains information for the UIC and BSC of both billets and bodies. In this manner, one can figure out which personnel are attached to what activity. The problem arises when one has to figure out which personnel are attached to which sub-activity. There is no information in the legacy systems that define these sub-activities. In the existing P-1 database, this problem is overcome by an already established table of activities with an assigned Command Sequence Code (CSC). Each CSC is associated with the UICs and BSCs from the legacy systems. The combination of CSC, UIC, and BSC make a unique activity. In order to maintain the same list of activities, I had to duplicate the UIC, BSC, CSC combination from the older system in the new system. I imported this table into a temporary table called tmpCSCCodes. This table contained all the known combinations of UIC, BSC, and CSC. The ACTIVITY table was created from the previous P-1 database system and contained the relevant CSC Codes. In order to relate BILLETS to ACTIVITY, I split the Activities into two parts, Single UIC and Multiple UIC Activities. Single UIC Activities are those that have only one UIC and Multiple UIC Activities are those that have several sub-activities under one UIC. The SQL structure for these two queries is as follows:

- *Single UIC:* `SELECT ACTIVITY.UIC FROM ACTIVITY GROUP BY ACTIVITY.UIC HAVING Count(ACTIVITY.UIC)=1`
- *Multiple UIC:* `SELECT ACTIVITY.UIC FROM ACTIVITY GROUP BY ACTIVITY.UIC HAVING Count(ACTIVITY.UIC)>1`

The Single UIC Activities were easily matched up to their respective billets by using the UIC to relate the imported billets to the ACTIVITY table. These billets were appended to another temporary table called tmpUICActivityWithtmpBillet. This table contains the UIC, BilletID, and ActivityID for each billet. To match up the Multiple UIC Billets to their respective Activities, I had to use the tmpCSCCodes table to bridge the gap between the imported billets and their related activities. This was accomplished through another append query shown below in Figure 28.



Field:	AUIC	BilletID	ActivityID	ActivityID
Table:	tmpCSCCodes	tmpBILLET	ACTIVITY	tmpBILLET
Sort:				
Append To:	UIC	BilletID	ActivityID	
Criteria:				Is Null
or:				

```
INSERT INTO tmpUICActivityWithtmpBillet ( UIC, BilletID,
ActivityID )SELECT tmpCSCCodes.AUIC, tmpBILLET.BilletID,
ACTIVITY.ActivityID FROM ACTIVITY INNER JOIN (tmpBILLET
INNER JOIN tmpCSCCodes ON (tmpBILLET.BSC =
tmpCSCCodes.BSC) AND (tmpBILLET.UIC = tmpCSCCodes.AUIC))
ON (ACTIVITY.CSC = tmpCSCCodes.CSC) AND (ACTIVITY.UIC =
tmpCSCCodes.AUIC) WHERE tmpBILLET.ActivityID Is Null
```

Figure 28.   Associating Multiple UIC Billets To Activities

2. Relating Bodies to Billets

The issues relating personnel to their respective billets are similar to the previous problem of relating billets to activities. The information received from the ODIS text files on personnel contains a UIC and BSC for each member. Again, because PERS 4413 does business differently, the UIC is not a unique identifier for an activity. There are eleven occurrences of multiple activities for a given UIC. The process is relatively the same regardless of whether the member is in an activity that has a unique UIC or one that shares the UIC with other activities. In each case, a member may have both a current billet and an ultimate billet listed. The current billet is generally where the member is stationed at the time of the data pull, and the ultimate billet information exists if the member has orders for their next duty station. This combination of information breaks down into the following four cases:

- Single UIC activity with current billet information
- Single UIC activity with ultimate billet information
- Multiple UIC activity with current billet information
- Multiple UIC activity with ultimate billet information

With both the MEMBER and BILLET table populated, the task here is to populate the intersection table MEMBER_BILLET with the correct information. A graphical representation of the query for single UIC activities (current billet) is shown below in Figure 29. Upon examination or the SQL statement (Figure 30) for this update, one will notice that a function had to be called to convert the date information from tmpBILLET to an acceptable date/time format for Microsoft Access. Since the date information in the original data was in a yyyymmdd format, Access could not properly interpret the string as a date, especially because many times the data was in a yyyymm format without the day information. In order to handle this, I created a custom function called ConvertDate using Visual Basic. This function takes the string in its current format and returns a valid date/time formatted string. The code for this function and all other VBA code is located in Appendix H.

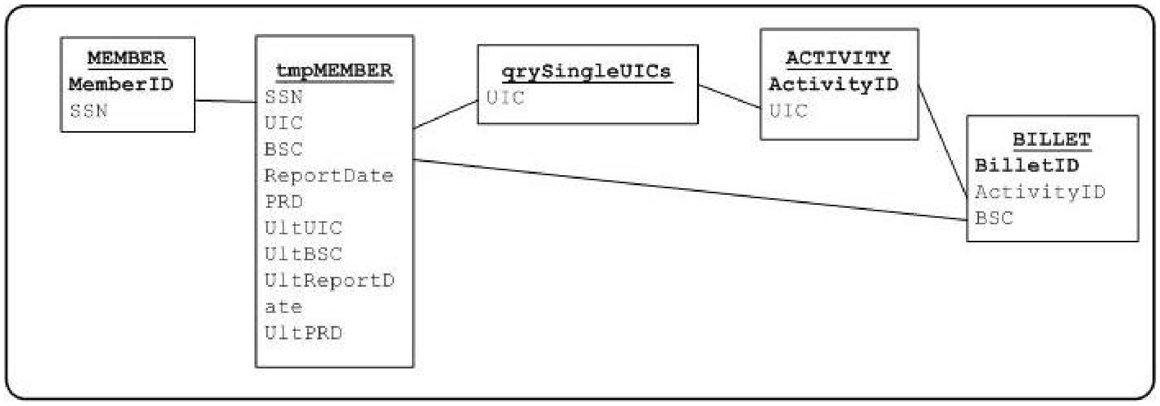


Figure 29. Associate Members to Billets

```
INSERT INTO MEMBER_BILLET ( MemberID, BilletID,
ReportDate, PRD, Type )
SELECT MEMBER.MemberID, BILLET.BilletID,
ConvertDate([tmpMEMBER]![ReportDate]) AS ReportDate,
IIf(ConvertDate([tmpMEMBER]![PRD])="",DateAdd("yyyy",2,Co
nvertDate([tmpMEMBER]![ReportDate])),ConvertDate([tmpMEMB
ER]![PRD])) AS PRD, "Current" AS Type
FROM (ACTIVITY INNER JOIN (MEMBER INNER JOIN
(qrySingleUICs INNER JOIN tmpMEMBER ON qrySingleUICs.UIC
= tmpMEMBER.UIC) ON MEMBER.SSN = tmpMEMBER.SSN) ON
ACTIVITY.UIC = qrySingleUICs.UIC) INNER JOIN BILLET ON
(ACTIVITY.ActivityID = BILLET.ActivityID) AND
(tmpMEMBER.BSC = BILLET.BSC);
```

Figure 30. Associate Members To Billets SQL Statement

The process for the other three cases is essentially the same with slight variances on the joins in the query and which fields are selected to be appended to the MEMBER_BILLET table. The ultimate billet queries are joined on the UltUIC and UltBSC fields and the multiple UIC queries replace qrySingleUICs with qryMultipleUICs.

B. MAINTENANCE PLAN

No matter how well designed, built, and tested a system may be, errors or bugs will inevitably occur. Bugs can be caused by a variety of reasons including, but not limited to, miscommunication of requirements, design flaws, unanticipated situations that were not tested, and unanticipated misuse of the programs. The challenge for every new system that has been implemented is having a plan for how to handle any problems that may come up.

1. Database

Microsoft Access has a built-in tool for database maintenance that compacts and repairs the database upon command. When users delete data or objects in an Access database, the file can become fragmented and use disk space inefficiently. Compacting optimizes the performance of Access databases by making a copy of the file and rearranging how the file is stored on the disk. The database can also be set to compact every time the application is closed.

In most cases, Microsoft Access can detect whether an Access file is damaged when it is opened and gives the option to repair it at that time. In some situations, Access may not detect that a file is damaged. It is a good idea to compact and repair the database on a regular basis. Access can repair a corrupted Access database table.

The users will be instructed in the proper procedure for compacting and repairing the database. As mentioned in Chapter II, the IT support staff does not have any plans to support this database application in any manner. Responsibility for the maintenance and support of the database will fall within the Community Management Analyst's purview. He will be responsible for ensuring that the data is well maintained and if any technical problems arise, he will be the one to try to solve the problem. If the problem is beyond his capacity, he also has the ability to contact me for support.

2. Application

The repair feature mentioned above can also help with minor application problems by fixing a problem with corruption in a form, report, or module. The Access repair function can also fix missing information that Access needs to open a particular form, report, or module. The database and application should be backed up on a weekly basis in case a crash occurs and causes irreparable damage to the application. For this particular application, the backup is a simple procedure of copying the two files (database file on the server and the GUI on the client) to a CD that is stored in a separate location from the server. The overall maintenance of Microsoft Access is the responsibility of the IT staff. They should ensure that the users have the latest software on their machines with all relevant patches installed.

The main objectives of system maintenance are two-fold. The first is to make changes to existing programs to correct errors that were made during design and implementation. Enhancements and new requirements are thus excluded from this activity. Future enhancements are addressed in Chapter VII. Secondly, system maintenance should preserve the aspects of the program that were already correct by attempting to avoid the cases where a "fix" to one area causes undesirable effects in another area of the system. Keeping these objectives in mind, minor modifications will be corrected in the event errors or omissions are discovered. Major enhancements as a result of new requirements or customer desires will be left for others to accomplish in a follow-on project if desired.

C. SCALABILITY

1. Database

If the need to expand to more users grows, the database is scalable to grow with it. As more and more users connect to the database and try to use it in its current state, the application's performance could degrade. Applications that do not scale well usually end up causing two problems. The first problem is that the developer of the application will have to spend more time trying to get the most out of the database in order to increase performance. The second problem is that the database system is now viewed as something less than perfect, and the users lose respect for the developer as well as the application and may give up on the system in its entirety. A scalable solution is capable of performing adequately even when a larger than expected group of people begin to use it. The key to building a scalable application is to minimize the amount of information being passed across the network and preventing record contentions when multiple people try to use the same information.

The size of the CEC detail shop is small enough that one person (Community Management Analyst) can be the administrator of the database with the other detailers only using a local copy of his database. There are no current plans to make this database into a production system or grow its user base beyond its current amount. Should the desire exist to expand the database to include other surrounding areas, then a few key changes would be considered:

a. Split the Database

One of the options already discussed in Chapter V is to split the database into two separate databases are linked via an ODBC connection. All the forms are placed in a front-end database while the data tables, queries, reports, modules, and macros are placed in a back-end database. Each client machine then uses its own personal copy of the front-end database to interact with a single copy of the back-end database that is located on a central server, thus reducing network traffic for the use of the application.

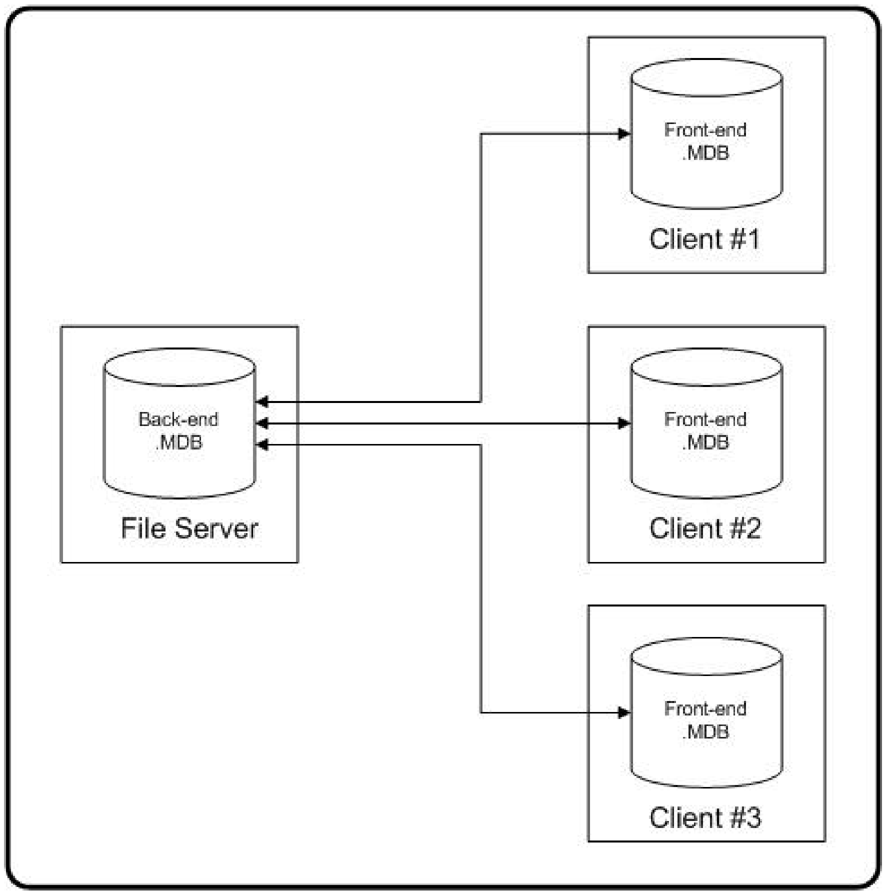


Figure 31. Scalability – Split Access Database

b. Migrate to the Web

Another way to improve the scalability of the application is to use an Access database and a web server. In this manner, anyone can communicate to the central database engine through Active Server Pages or any other type of web page capable of interfacing with a database. This puts the burden of running queries and responding to requests on the web server and will not tie up the network as much as multiple people accessing the database application over the network.

c. Migrate to a More Robust DBMS

If neither of the above is sufficient, the database can be migrated to a more robust Database Management Tool to handle a larger number of simultaneous users, such as SQL Server or Oracle. The task of migrating an Access database to a larger DBMS is not that difficult and could be done with this particular database relatively simply.

2. Application

If the need arises, the application itself could be written in Visual Basic instead of using Microsoft Access. The application could then interface with the Access data tables or another DBMS to provide functionality to the users. A Visual Basic program that interfaces with a SQL server would most likely give the user the best performance for the application. The downside to this solution is that it requires much more maintenance and oversight to keep the program running.

D. HUMAN RESOURCES

When implementing a new system, the question of who will maintain it is a very important one to address. If there is not anyone who is qualified and has the time to keep the system running and handle any problems with the system, then the system may be doomed to failure. An analysis of the current organization’s staff and their capabilities can go a long way in determining what type of support the system will have with the current staff. Some organizations may decide to hire a person in direct support of the new application depending on their needs. In the case of the CEC detail shop, there are not any additional personnel requirements to support the new system. The Community Management Analyst will be the system administrator and handle any problems with the database. The IT staff will support the detail shop by providing a network to work on and the latest Microsoft Office software.

E. TRAINING

Training is not a one-time event in the implementation of a new system. The CEC detail shop has a revolving door for its military staff. Detailers will typically spend one to two years working in their position and then move on to something else. Their replacements will come into the office with no prior knowledge of the database system. These replacements will need training on how to use the system. Each new user will be given a copy of the user’s manual to help them learn the application. As discussed in the training plan in Chapter V, the end-users will train new personnel on the systems on an as-needed basis. This training will coincide with the turnover of duties from one officer to the other. The Community Management Analyst is a civilian employee who provides continuity to the office and will be the primary source of information on the system for new users.

The detailing management system was designed and built with objective that it would be easy to use and require minimal maintenance for the end-users. With a little care, the end-users will be able to use this system effectively and efficiently to help them manage the personnel for which they are responsible.

F. DATA IMPORT AND MAINTENANCE SUMMARY

A database is only as good as the data that is in it. A common expression in database circles is “garbage in equals garbage out.” If the data that is imported is not reliable, then the entire application is useless. Users will not waste their time with a system that does not provide them with accurate and timely information. The data import routines developed for this application help the users clean the data to make it as accurate as possible. Inconsistency in data formats are also addressed in these routines.

Ensuring that the data is accurate and up to date is just one part of maintaining a database application. The physical data itself must also be maintained. Microsoft Access provides several simple methods to maintain the data tables as well as repair any corrupted tables. Keeping the Microsoft Access application up to date with the latest recommended patches will also help maintain the stability of the application.

To ensure that the application will last for a reasonable length of time, it is important to design and build scalability into the system. This application is designed with future growth and enhancements in mind. The system can be migrated to a larger DBMS or to the web with relative ease, minimizing the impact on the users.

As CEC personnel rotate in and out of the detail shop, the need for training will continue. Easy to understand user manuals will assist in this area as well as training from current personnel during their turnover of duties. The Community Management Analyst, a civilian employee, provides continuity in the office and will become the resident expert in the application.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. LESSONS LEARNED, RECOMMENDATIONS, AND CONCLUSION

A. LESSONS LEARNED

In the development of any system, there are always problems that arise and mistakes made. A few specific problem areas have been identified to avoid in future developments of similar projects. The purpose of this section is not to identify every problem area and mistake made during the development but to point out key changes in the process that would have streamlined the development.

1. Ask the Right Questions

The most important step in any systems development is the requirements analysis. Key information about the design of the system will be obtained through various interviews with the organization. If the analyst does not ask the proper questions from the appropriate personnel, crucial information may be left undiscovered until late in the design.

As an example, the discovery of the existence of the Navy Personnel Database (NPDB) was not made until one month prior to the completion of this thesis. Linking the new system to a relational database instead of downloading and importing separate text files would have made the new system more efficient and easier to use. Attempts to gain approval for read-only access to the NPDB were unsuccessful. Had this system been discovered earlier, the approval process may have been accomplished and the new system could have been designed to interact with the NPDB. Because the requirements analysis trip was conducted on short notice, there was not adequate time to prepare questions and arrange for interviews. Thorough preparation and arrangements may have resulted in the discovery of the NPDB.

2. Avoid Duplication of Existing Systems

One of the pitfalls to avoid when designing a new system for an organization is to duplicate the old system in design or functionality. It is very easy to look at the old system and try to recreate it in the new system without investigating the reason the old system was designed in that manner. One area that may be susceptible to duplication is report generation and formatting.

As an example, the original P-1 report contained an index showing all the CEC personnel with information on their rank, year group, P-Code, and qualifications earned. Since this report was based of a de-normalized table, it was easy to display each qualification a member had obtained. In the new system, qualifications are related to members in a many-to-many relationship. This normalized structure makes it more challenging to display each member's earned qualifications on the same row as the member. After considerable effort, this challenge was overcome using a cross-tab query and a concatenation statement to merge all the qualifications for each member into one field for display purposes. Perhaps the better way to address these types of issues is to return to the customer and discuss the importance of their report format. If all the information is there, but displayed in a slightly different manner, then maybe the report is sufficient for the customer's needs.

3. Familiarize Yourself with A CASE Tool Early In the Process

When working on any project, whether it be building a piece of furniture or developing a software application, it is crucial to have the right tools for the job. Even with the right tools, the project can be delayed or flawed if the builder does not know how to use them properly. While Visible Analyst provided assistance in the development of the database model, the process was not an easy one. There was a steep learning curve in using the tool proficiently.

For example, initial attempts to generate the database schema in Access from Visible Analyst resulted in an error message that prevented the generation from occurring. Several days were spent trying to resolve the problem and it was finally resolved by shipping the Visible Analyst project and the Access database to the technical support personnel for Visible Analyst. The fix was a simple case of incorrectly entering data into one field on the schema generation options.

Another case involved the creation of referential integrity constraints in the Visible Analyst project that would transfer to the Access database when exported. According the user manual and help documents in Visible Analyst, this particular edition of the program was capable of creating referential integrity constraints and exporting them to Access. However, the check box to create these constraints was grayed out and not selectable. Without this feature working, each time the database schema was exported to Access, I had to manually add the referential integrity constraints in Access for each relationship. Visible Analyst technical support supplied a solution by providing instructions on how to modify one of the program files in Visible Analyst to allow referential integrity to work properly.

These problems caused some delay in the development of the prototype and could have been avoided had I been more proficient in my understanding of the tool and all its features. Deciding on a tool to use early on in the project would have helped streamline the development phases later on in the project.

4. Create a Detailed Project Plan Based on Methodology

Development projects of this nature can often lead the developer in a different direction than what was envisioned. A detailed project plan based on the methodology chosen can help to keep the developer focused on the task at hand. A plan that identifies each phase of the project with every task required to accomplish the phase included with due dates reduces the risk of errors and omissions in the project.

For this project, a general project plan was created but it did not include the level of detail for each phase nor did it include a timeline for completion of individual tasks. Had this been done, the prototype application would have been completed earlier allowing for more reviews and iterations, resulting in a better product.

A detailed project plan would have also generated a checklist of items to ensure that all the requirements had been met. This problem arose when discussing the prototype application with the Community Management Analyst. He pointed out that the Staffing Plan also has a personnel listing at the front of the plan. This item had been missed initially, requiring additional work to add the personnel listing to the Staffing Plan. Had a checklist of required items been created, the personnel listing would have been included in the first iteration of the prototype.

B. RECCOMENDATIONS

1. Support A Follow-On Project That Will Include The Following:

a. Integration with the NPDB

Any future work on this system should focus on integrating the application with the NPDB. The application should not be allowed to modify anything in the NPDB without proper approval, but a read-only view of the data would provide a more efficient method of providing the information required by the CEC detailers. It is likely that using a view from the NPDB would negate the need for much of the data cleansing currently required. The data cleansing portion of work is the most time consuming and frustrating for the users. Taking this step out would improve office efficiency and productivity.

b. Provide Web Access to the System

The P-1 document is read by every CEC officer in every part of the world. The best tool to deliver this document worldwide is the Internet. A follow on thesis project could provide a web management tool for the detailers as well as an Internet version of the P-1. The detailers maintain a heavy travel schedule. A web management tool for the detailers would give them access to the information they may need while traveling to different regions. A web version of the P-1 could have many enhanced search capabilities so that a CEC officer could search for a particular type of billet where the incumbent member is due to rotate in a given period. A prototype for this type of interface was recently completed in another thesis project. Combining that prototype with this application would be an interesting and worthwhile project that would provide great benefits to the CEC community.

c. Enhance Functionality

Currently, every time the users want to create the P-1 or Staffing Plan, they must import a fresh set of data to the database and clean it up all over again. An idea for increased functionality could be to import the fresh data and compare it to the existing data. Only the conflicting data would be shown to discard, add, or change. This may streamline the data import process if done on a regular basis. Other functionality enhancements requested by the users could be accomplished as well.

2. Reexamine Current Business Processes and Rules

One of the issues encountered during the development of this system was how the CEC defines activities differently than the Navy. In the existing ODIS system, activities are defined by their UIC, whereas the CEC further breaks down those activities into several more. This situation made it extremely difficult to relate billets and personnel to activities. Perhaps the NPDB contains a unique identifier that further breaks down the activities to the level of detail that the CEC wants to see, but I was not able to get a look at the data or the structure of the NPDB. Standardization across the Navy for what defines an activity should be addressed for future enterprise systems.

Another issue is how a billet is defined. Currently a Public Works Officer billet at one activity is different from a Public Works Officer billet at another activity. In an ideal model, there would be one Public Works Officer billet that could reside in many activities. In other words, the relationship between activity and billet would be many-to-many instead of one-to-many. Perhaps the billets are currently organized due to financial reasons. Each billet is associated with certain cost information. This information could easily be input into the intersection table, making for a much cleaner database schema.

C. CONCLUSIONS

The original concept of this thesis was to create a much larger system that the CEC detailers would use on a daily basis and have a web interface for the CEC community to access. The web interface would pull its information directly from the database residing at BUPERS. The application would be used to manage all data related to CEC officers and support the production of the P-1 and Staffing Plan. The scope of this project was reduced considerably after the initial discovery trip. The web site could not be hosted at BUPERS, but would have to be hosted at the Naval Facilities Engineering Command Information Technology Center (NITC) in Port Hueneme, CA. The detailers did not want any new system to manage their information. They only wanted an easier method to produce the Staffing Plan and P-1. The scope was further reduced to its current state when the other CEC officer received orders to leave earlier than expected.

Working under the constraints of importing the data from the delimited text files resulted in a system that is not as flexible as desired. Any change to the delimited files will result in a requirement to modify the data import routines. A view of the existing NPDB would have provided a more adaptable system than the current one.

Although the system is not perfect, it does meet the goals set forth in the beginning of this thesis. The system successfully integrates itself with the legacy systems via data import routines. The method of importing and cleansing the data is now easier to than before. Once the data is in the system, simple maintenance with user-friendly forms can be done allowing for the rapid generation of the Staffing Plan and P-1. In its current state, the system meets the requirements of the end-users and serves its purpose until a permanent solution can be developed.

APPENDIX A: DEFINITION OF TERMS

Macro	A saved sequence of commands or keyboard strokes that can be stored and then recalled with a single command or keyboard stroke.
Primary Key	A unique field in a table which identifies the row.
Relationships	Defines how entities interact with each other
Queries	A question required to be expressed in a specific manner to locate, update, add, or delete records in a database table
Legacy system	System that have been inherited from languages, platforms, and techniques earlier than current technology
Entities	Some unit of data that can be classified and have stated relationships to other entities
Scalability	The ability of a computer application or product (hardware or software) to continue to function well as it (or its context) is changed in size or volume in order to meet a user need
Stovepipe	A system that is in place to serve only the primary user and does not interact or interface with other systems in use
End-user	Personnel for whom a hardware or software product is designed
Data integrity	Refers to the validity/reliability of data

APPENDIX B: INITIAL DISCOVERY TRIP REPORT

Trip Report

Initial Visit to Navy Personnel Command
18 – 20 June 2001

Presented By:

P1 Online
Development Team

P1Team@nps.navy.mil

Brian Weinstein LT, CEC, USN
&
Chris Rader LTJG, CEC, USNR

3 July 2001

Executive Summary

The ultimate goal of our project is to make the P1 readily available online to the entire CEC community. With this goal in mind and a thesis to complete, we considered the technological and design possibilities and came to the conclusion that a Community Management System would be the logical companion to the P1. This combination is also perfect in size and scope for our thesis work. This initial trip to Naval Personnel Command (NAVPERSCOM), funded by Naval Postgraduate School, was necessary to begin work on a coherent P1Online and a Community Management System. This report is intended to serve as a research findings document provided to PERS 4413 as a result of the trip.

The initial trip to Navy Personnel Command had several purposes and was critical for establishing a direction for the next phase of the project, the Design and Development Phase. Our initial vision was to provide an integrated Community Management System that would tie directly into the existing NAVPERSCOM system including a web-based capability that would be available to all members of the CEC community worldwide. The integration of the system with the existing NAVPERSCOM system would have allowed for near real-time data access. Unfortunately, technical as well as policy related factors make the integration of a new system with the NAVPERSCOM system not feasible within reasonable time and cost constraints.

The primary goals of the trip were to learn the fundamentals of the detailing process, to evaluate NAVPERSCOM systems (hardware and software), and to refine the scope of the project. A basic understanding of the detailing process was obtained through question and answer sessions with each of the PERS 4413 detailers. Each of the detailers was questioned on how they accomplish their tasks and what information systems they use to facilitate their processes. Several thorough discussions were held with NAVPERSCOM staff members in coordination with participant observation of existing hardware and software systems currently in use. Both hardware and software systems at NAVPERSCOM are very outdated as noted in Paragraph 1.b, Page 3. The existing systems do not readily lend themselves to integration with a new system and PERS 4413 does not wish to create a new system that could be incompatible with existing or planned software. With this in mind, a revised, limited scope of work has been developed to communicate the specific deliverables that will be provided to PERS 4413. Specifically, the deliverables include the P1Online, an improved Access database to facilitate management of the P1 data, and an advanced reporting capability available to the members of the CEC detail shop. On our own, we will also continue on with the entire original scope of the Community Management System for the purpose of our thesis using the basic knowledge obtained in the initial trip. We plan to demonstrate this capability to the PERS 4413 staff in order to allow an opportunity for the staff to request additional functionality in the deliverable system and to obtain feedback relative to our thesis.

The trip to Millington was highly informative and beneficial. We discovered basic information regarding the detailing process and how information systems currently facilitate the process. Our goal is to complete the P1Online and the improved Access database with additional reporting capabilities for the Civil Engineer Corps and its detailers, as well as, develop a full blown conceptual system for the completion of our thesis for Naval Postgraduate School.

Introduction

Our initial trip to Navy Personnel Command (NAVPERSCOM) Code 4413 was challenging, but successful. Our goal was to develop a basic understanding pertaining to Navy Personnel Command operations and information systems. Specifically, we focused our efforts on the following areas:

1. Achieve a basic understanding of the detailing process.
2. Understand P1/Staffing Plan generation
3. Understand the purpose of the Staffing Plan
4. Evaluation of current NAVPERSCOM systems and databases in use
5. Investigate current architecture/infrastructure of IT systems at NAVPERSCOM
6. Discuss fiscal issues

We were successful in accomplishing our goals for the initial trip. We were able to obtain a substantial amount of information from the NAVPERSCOM staff. The following section describes the trip's results.

Discoveries

1. Existing Systems

    a. Standard Navy Personnel Management Systems
    Based on the initial visit, our understanding of the current system architecture is displayed in Figure 1.

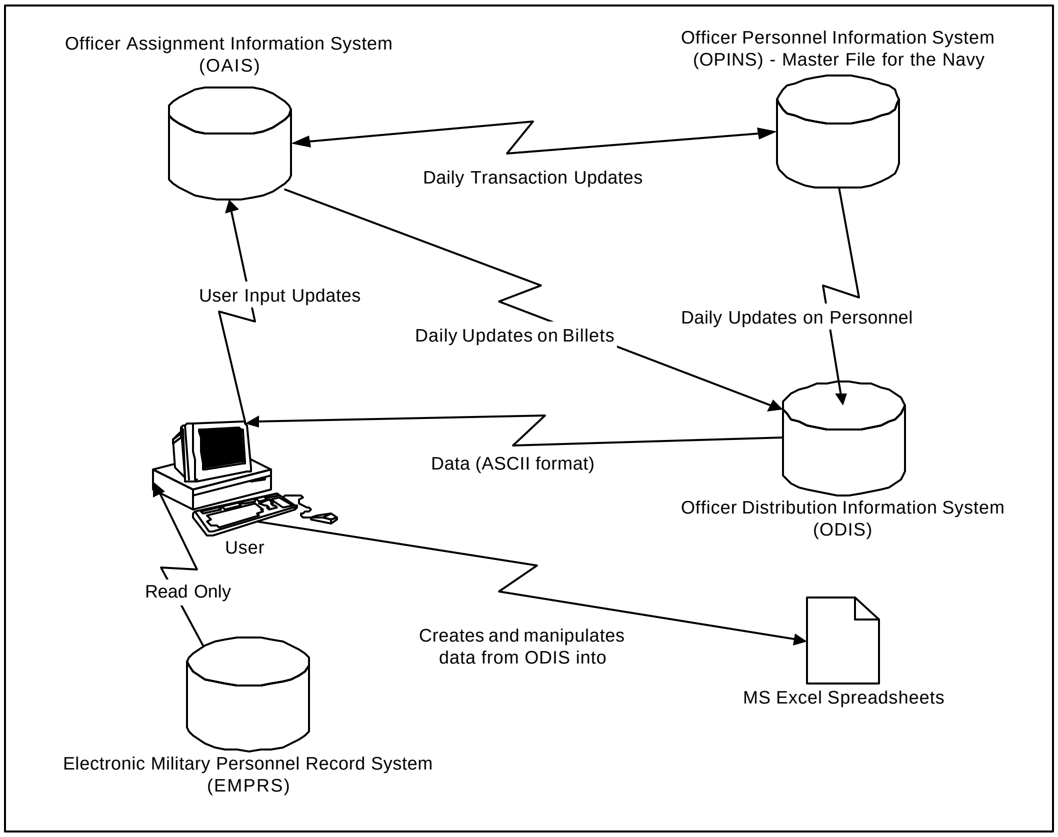

Officer Assignment Information System (OAIS)
Officer Personnel Information System (OPINS) - Master File for the Navy
Daily Transaction Updates
User Input Updates
Daily Updates on Billets
Daily Updates on Personnel
Data (ASCII format)
Officer Distribution Information System (ODIS)
User
Read Only
Creates and manipulates data from ODIS into
MS Excel Spreadsheets
Electronic Military Personnel Record System (EMPRS)

Figure 1: Current System Architecture

The master file for the Navy (OPINS) is maintained on a large mainframe located in Cleveland, OH. The locally maintained mainframe (OAIS) handles the day-to-day operations for personnel management. The personnel in the detail shop make any changes with regards to personnel or billets in this system. OAIS and OPINS perform nightly transaction updates to synchronize the information. When a staff member needs to extract information to create reports or spreadsheets to perform their duties, this data is taken from the Officer Distribution Information System (ODIS) in ASCII format. The ASCII data is downloaded in a delimited format and imported into Microsoft Excel for manipulation and reporting. The Electronic Military Personnel Record System (EMPRS) is a separate application that the detailers use to view their constituents' digital records during the detailing process. The four systems represented here are all stovepipe mainframe applications that do not relate to each other except in a nightly transaction-oriented process for synchronization of data.

b. PERS 4413 Systems

Evaluation of NAVPERSCOM systems, databases, and spreadsheets currently in use was one of the objectives of the visit. The current platforms used are the following:

Hardware: Pentium desktop systems that are approximately 4 years old. The average processor speed is around 166 MHz, with each containing 64 MB of RAM.

Software: All of the personnel are running on Microsoft Windows 95 Operating System with Microsoft Office 97 for their Office Application Suite.

Windows 95 is still being used because the EMPRS system was coded specifically for Windows 95 and the contractor went out of business and took the source code with them, therefore no changes have been made. There are current efforts underway at NAVPERSCOM to rewrite the EMPRS system so it can be used with the latest operating systems. The Navy Marine Corps Intranet (NMCI) is another reason for a delay in the upgrade of systems and software at NAVPERSCOM. The command intends to upgrade the systems during the rollout of NMCI, which has been delayed until next year with no specific timeframe for execution.

2. NAVPERSCOM Support

NAVPERSCOM is currently undertaking many IT related projects including, but not limited to, NMCI, legacy systems upgrades, and an upgrade to the EMPRS system. There is a local development project underway (SEERS) which is supposed to combine all the stovepipe systems into one relational model for use by NAVPERSCOM. The systems described above are all planned to be included in this local project along with other systems not mentioned. There is currently no timeframe for the execution of this project and it is understood to be just in the conceptual phase, which leads to the estimation that the successful completion of this project is at least 5 years away. There is also a project underway at NAVPERSCOM to try and make these management systems entirely web based. DOD is also examining the possibility of developing the Defense Integrated Human Resource System (DIMRS), which is intended to be a personnel management application for the entire military. It is not known whether these two projects will be integrated, superceded, or separated. Given the number and magnitude of the projects that NAVPERSCOM is undertaking, they expressed great concern in allowing another system to be created to support PERS 4413 outside the realm of their oversight. This issue alone led us to modify our original scope as will be discussed later in this document.

3. Data Discrepancies and Integration

Inherent with static display applications come certain data discrepancies. These are created by the lag time between updates of each stovepipe system. A transaction oriented update system can also lend itself to the possibility of errors and omissions. An update may be sent to one system with no error checking to see if the other system received all the information correctly. There may also be old information in the system that has not been removed or updated. Given these possibilities, there must be a data validation method. This validation, or sanity check, must be performed each time data is extracted from ODIS and imported into MS Excel. This type of validation requires a considerable amount of corporate knowledge only known by the staff. In terms of preparation of the P1 and the Staffing Plan, PERS 4413s (Dennis Potter) spends approximately 8 hours every two to three months performing the data manipulation and validation to produce each instance of the P1 and Staffing Plan.

After discovering what current systems were being used and how they related, we determined that actual real-time integration of the existing systems with the P1 database was not possible. This is due to the architecture of the systems and also the policies and projects underway by NAVPERSCOM. The current systems are mainframe applications that maintain separate flat files with the data. This type of architecture does not lend itself to an integrated relational model, which is generally the most efficient for data handling.

4. Work Processes

During the trip, time was spent with each member of the PERS 4413 team investigating how they do business and what tools they use to accomplish their day-to-day tasks. We discovered that while the methodologies that each detailer uses to perform their tasks are generally different; the data that is required to support them is quite similar. The main differences in how they use the data is in the sorting, display, and a few different field selections. Each detailer uses a spreadsheet that originates from an ODIS data pull. All of the spreadsheets are very similar, but have minor differences in implementation. Each detailer also uses Microsoft Outlook for a journal of detailing activity. This journal capability provides a link between e-mails and notes made by the detailer. Each detailer keeps a handwritten phone log, varying from a record of date, time, and conversation to just a message of who called with a note to call them back. All the detailers use e-mail and the telephone extensively to communicate with their constituents.

5. Staffing Plan & P1 Generation

As mentioned earlier, the process by which the Staffing Plan and P1 are generated is lengthy and arduous. It generally takes a minimum of one full day of Mr. Potter's time every time the Staffing Plan or the P1 is generated. During the trip, we witnessed the generation of the Staffing Plan. The process starts with two separate ASCII formatted data pulls from ODIS. One pull is data on bodies (personnel) and the other is data on billets. Once these ASCII files are pulled, they are imported into MS Excel and then manually manipulated for data validation and cleanup. This manual manipulation and validation takes approximately four hours for completion. It appears that much of this manipulation and validation can be automated, but there is still some that must be checked manually. For example, the data pull brings in many reserve CEC officers who are not included in the Staffing Plan or P1. Currently, the only way to know whether or not to include them is by corporate knowledge of the CEC community. Once all the manipulation is completed in MS

Excel, the data is then imported into an existing MS Access Database where internal queries and macros in the database attach the billets to bodies and then perform some further validation and manipulation. The manual manipulation in MS Access also takes approximately four hours. After this process was completed, the Staffing Plan was finished. The existing MS Access Database is essentially a collection of spreadsheets that do not have primary keys or relations to each other. Therefore, the Access database is not a relational database rather a reporting tool. The existing queries appear to create the relationships as needed. There is duplication of data in these tables that can be solved by creating a relational model, which will make the process of generating the Staffing Plan and P1 simpler and much more efficient. Once the P1 is generated in the Access database, it is shipped via email to Mr. John Wang at NITC in Port Hueneme, CA. Mr. Wang then takes the database and places it on a server for the NAVFAC intranet web pages to access. This process is done once or twice a year. Theoretically, if the P1 database is kept current through a more efficient means, a web site hosted at NAVPERSCOM that pointed to the updated database would be more current then the one which is shipped to NITC once or twice a year. Because the P1 is a community management tool and not an officially supported document and given the current projects underway at Navy Personnel Command, they did not express any interest or desire to host the P1 website. For this reason, it was determined to continue to work with NTIC for the online publication of the P1.

Revised Scope

After an initial visit to Navy Personnel Command Code 4413 and at the request of PERS 4413, it has been determined that the deliverable scope of work will be reduced. The detail shop did not request the original scope of work, which included the Community Management System, nor do they plan on implementing our complete product. For the purposes of fulfilling the thesis requirements and our continuing education at the Naval Postgraduate School, we will continue to develop the Community Management System based on the knowledge we have gained thus far at no additional cost. If, during the course of development, PERS 4413 requests additional functionality that is included in our thesis work it will be possible to incorporate those changes into the specific deliverables. The following sections describe how the project will be broken up into deliverables and thesis work.

1. Deliverables

The NAVPERSCOM deliverable will have its own scope of work that will be a small subset of the entire scope of work. It will include an Access database built to incorporate all the information utilized in the P1 and the staffing plan. This deliverable's criteria are that it must allow for fast transactions and be user friendly and easily maintained. It will be designed in a relational manner, and it will provide fast search, sort, and reporting capabilities to the PERS 4413 staff. Mr. Potter, as part of his day-to-day business, will maintain this database. We will provide him with a users guide. The database will be very user friendly and will require minimal work to validate against the ODIS data pulls with which it must be coordinated. The goal of the project is to allow for this task to be performed on a daily basis. When this goal is met, it will greatly facilitate the database's reporting capability, and it will also provide the backbone for a very current P1Online database. The deliverable scope will also include the P1Online personnel directory. This P1Online will be similar in purpose to the current directory that is available on the NAVFAC intranet. The new website will have expanded search capabilities to include easy to use methods of searching for members of the CEC, billets, activities, and combinations of the above. The new site will be designed in keeping with the style of the current P1 and will utilize Active Server Page technology to access the P1 database. The website will also provide, to the extent the community desires and regulations allow, an integration of personal contact information. The website will be secured using modern web-security technologies in order to make sure that the information is available only to authorized users of the system. The Civil Engineer Corps is pushing towards more web based applications and the P1Online is just another piece of this picture.

2. Thesis Work

The entire scope of work for this project is larger than described above. For the purpose of thesis research, an ent ire Community Management System will be developed. This system will combine a desktop system with an online capability. The desktop system will provide detailers an integrated system for managing billets, bodies, and assignments. It will provide a detailing journal, detailing preference viewing, contact information, assignment history and summary, and other features as developed. This system will be integrated with an expanded version of the online personnel directory that will include contact information updates, detailing preference submissions, as well as a search capability for people, jobs, and activities real time from the active Community Management System database. The entire system will be built with the intention of being a stand-alone system capable of supporting a detailing office for a community similar in size to the Civil Engineer Corps. The system will not be designed with the intention of implementation because some of the capability is already provided by the OAIS system used by Navy Personnel Command and any newly created systems will not be supported. We would like the opportunity to demonstrate the capability of the system to the PERS 4413 staff in order to obtain feedback relative to the completion of our thesis. This will also give the detailers an opportunity to evaluate the capabilities of the Community Management System and consider incorporation of some of its advanced features into the above NAVPERSCOM deliverables.

Revised Cost Proposal

As with any project, cost is a factor. This project's primary expense includes reference materials, limited software licenses, and travel cost to NAVPERSCOM and NITC. The initial cost proposal submitted included costs for additional hardware and software purchases necessary for the implementation of the originally envisioned system hosted at NAVPERSCOM. The revised cost proposal submitted following the trip removed the additional hardware and software costs but also included additional costs for travel to NITC to discuss and implement the web solution there. Based on discoveries from the trip and further discussions with PERS 4413 since returning, the scope has been revised as previously mentioned. With this revised scope comes an additional revision in the cost proposal. The revised proposal is provided below.

Phase I – Feasibility Study & Business Model Analysis	**FY**	**Estimate**	**Notes**
1. Initial scope of work			
2. Fact finding trip to NAVPERSCOM		Funded by NPS	
a. Business model analysis			
b. Hardware evaluation			
c. Software evaluation			
d. Network evaluation			
4. Fact finding trip to NITC Port Hueneme (John Wang)	01	$500	POV travel down to Port Hueneme (1 night stay)
5. Refine scope of work			
6. Cost Benefit Analysis			
7. Identification of data migration issues			
8. Data Modeling			
Phase I Subtotal		$500	

Phase II – Design & Development	**FY**	**Estimate**	**Notes**
1. Resolution of data migration issues			
2. Microsoft Access database development	01	$205	Reference Books
a. GUI interface conceptual design			
b. GUI interface development			
c. Prototyping & Testing			
3. Web interface development	01	$310	Reference Books
a. Web site conceptual diagram			
b. Website development			
c. Prototyping & Testing			
4. Report development	01	$430	Crystal Reports 8.5
a. Report layout drafting			
b. Report development			

c. Prototyping & testing			
5. Complete data migration			
6. Software purchase (required)	01	$320	Office 2000 or XP for Dennis Potter
Phase II Subtotal		$1,265	

Phase III – Beta Testing	**FY**	**Estimate**	**Notes**
1. Beta testing trip to NAVPERSCOM	02	$2,000	
2. Beta revision discussions			
3. Beta revisions per previous discussions			
Phase III Subtotal		$2,000	

Phase IV – Implementation & Training	**FY**	**Estimate**	**Notes**
1. Help file development	02	$300	Software and Books
2. Final application & web interface changes			
3. Final data migration and validation			
4. Implementation trip to NITC, Port Hueneme	02	$1,500	
5. Implementation & training trip to NAVPERSCOM	02	$2,500	
Phase IV Subtotal		$4,300	

Cost Summary	**Estimate**	**Notes**
Fiscal Year 2001 Total	$1,765	
Fiscal Year 2002 Total	$6,300	
Grand Total	**$8,065**	

Please note that additional trip funding may be necessary if additional functionality is requested during the Beta Testing Phase of the project.

It is also recommended that the systems (hardware and software) in PERS 4413 be upgraded at the earliest opportunity. At a minimum, Dennis Potter will require an upgrade to Microsoft Office 2000 Professional or XP Professional. This cost is included in the above proposal.

Closing Remarks

We would like to take this opportunity to thank the staff of PERS 4413 for their support during our initial visit to NAVPERSCOM. We recognize that the trip came about with very short notice, and we appreciate the detailer's willingness to be flexible in allowing us time to meet with them. We look forward to providing PERS 4413 and the Civil Engineer Corps a quality product. Your continued support is critical to the success of this goal. Thanks again, and please feel free to contact us with any comments or questions. The team can be reached at P1Team@nps.navy.mil.

APPENDIX C: SPREADSHEET TOOLS USED BY DETAILERS

SPREADSHEETS ARE REPRESENTATIVE OF FORMAT, BUT CONTAIN NO VALID DATA

P4413A DETAILING MANAGEMENT SPREADSHEET

BSC	COMMAND	TITLE	BILLET RANK	I-RANK	PROM	INCUMBENT	YG	RPTD	PRD	ULT BILLET	PRIMARY	IDEA	IDEA	COMMENTS
05784	ACB 2	Cmpny Off	LCDR	LCDR	z1	Name	90	9909	0109	PG Schl	Name 1	Name 2	Name 3	

P4413E DETAILING MANAGEMENT SPREADSHEET

RANK	NAME	YG	PROM	PREC #	DOR	DUTY STATION	RPTD	PRD	APPROVED	PREF 1	PREF 2	PREF 3	PREF 4	COMMENTS
LT	NAME	1990	S0	9843098	19940201	NSCHCECOFF PHUNE	20000201	200202						

P4413C DETAILING MANAGEMENT SPREADSHEET

B-RANK	B-DESIG	ACTVY	B-TITLE	YG	RANK	PROM	NAME	PRD	ULT ACTVY	BACK-FILL	AUIC	BSC	TYPE	LOCN
LTJG	5100	NMCB FIVE	CMPY OFF	2002	ENS		NAME	200202	DLI	NAME	93439	82538		

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D: SAMPLE ASCII DELIMITED FILES

DATA PRESENTED BELOW ARE REPRESENTATIVE OF FORMAT, BUT CONTAIN NO VALID DATA

```
4530907500!00021!40140!LCDR!AIDE/N4A1 AIDE/ADMIN ASSISTANCE!ACD!!!!OPNAV!5100!000000!999999!11!02110100!ARLING

4530907500!00021!41415!LCDR!LOGISTICS/NN22D CARGO OFFLOAD SYS!ACD!!!!OPNAV!5100!000000!999999!11!02110100!ARLING
```

SAMPLE FROM THE BILLETS FILE

```
12345!98765!LT!WEINSTEIN BRIAN LOUIS!19940!5100!!20001226!*!LNFEC NORVA!00000000!!!!!!55114!05290!MCB 5!5100!FAC ENG/ASST
CONTINGENCY ENGR!LT!200207!5100!LT!CMPNY OFF NCF!200407!!!!!!02511760!!Y!C!M!123456789!100!!!!!
```

SAMPLE FROM THE BODIES FILE

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E: SAMPLE SPREADSHEETS OF IMPORTED DATA

DATA PRESENTED BELOW ARE REPRESENTATIVE OF FORMAT, BUT CONTAIN NO VALID DATA

IMPORTED BILLET INFORMATION

AUIC	BSC	BRANK	BTITLE	PAQD	SAQD	BSSC	ADDU	SNAME	BDESIG	BEGIN	END	MMC	COMMENTS	GEOLOC
11111	00101	LCDR	AIDE	ACD				OPNAV	5100	000000	999999	11	02110100	ARLING
11111	00110	LCDR	LOGISTICS	ACD				OPNAV	5100	000000	999999	11	02110100	ARLING

IMPORTED BODIES INFORMATION

AUIC	BSC	RANK	NAME	YRGRP	DESIG	SSC	RECDDT	PRD	ACTNAME	PRECNBR
12345	99990	LT	WEINSTEIN BRIAN LOUIS	19940	5100	6201T	20000813	200209	S PG MONTEREY	082354658

CONTINUATION OF MEMBER'S DATA

AQD1	AQD2	AQD3	AQD4	AQD5	ULTAUIC	ULTBSC	ULTSNAME	BDESIG	BTITLE	BRANK
950	AC1				*					

CONTINUATION OF MEMBER'S DATA

ULTEDA	ULTBDESIG	ULTBRANK	ULTBTITLE	ULTPRD	AQD6	AQD7	AQD8	AQD9	AQD10	GEOLOC
										02062250

CONTINUATION OF MEMBER'S DATA

PROMSTAT	ETHNIC	RACE	SEX	SSN
	X	C	M	123456789

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F: ORIGINAL P-1 PROGRAM DATA MODIFICATION RULES

RULES FOR MODIFYING BILLET INFORMATION FROM THE IMPORTED DATA

P1 BILLETS

Make majority of changes in EXCEL
- Add Headers (Titles)
- sort on PAQD (Realign/Fix top records)
- sort on ADDU
- delete "N" except (entire record)

UIC	BSC	
00011	47100	(OPNAV N46)
	47240	
57034	00110	(22 NCR)
57046	05020	(30 NCR)
62980	44580	(DETAILERS)
62980	44580	
62980	40245	
62980	40270	

- sort on AUIC
- delete the following records

N1 Students	(UIC 31013)	
TPPN		(UIC 33119)
	PEP Billets	(not in use)
	N4 Students	(UIC 44952)
	N7 Students	

<u>Delete PEP billets not is use</u>

41591/20445 PEP-USAF NELLIS
42048/34100 PEP-USAF RAMSTEIN
49429/00110 PEP-USA PANAMA
42051/36100 PEP-USAF OFFUTT
42049/36100 PEP-USAF TYNDALL
57034/00560 (ARMY CAPT) 2NCB
57046/05920 (ARMY CAPT) 3NCB

<u>Delete USNR billets</u>

46650/06061 HQ NAVFAC
00025/42495 HQ NAVFAC
57034/00135 ADMIN/NAVRES 2NCB
57092/00560 NAVCOASTWARGRP1
62638/00560 NAVCOASTWARGRP2
63139/13010 NAVRES S CLARA
32791/42550 OPNAV

<u>Delete CEC ADDU billets</u>

44221/05000 EFD SOUTH (NPC)
44221/05100 EFD SOUTH (NPC)
44221/05200 EFD SOUTH (NPC)
62477/01450 EFA CHES (OPNAV)
62477/01400 EFA CHES (OPNAV)
62477/05210 EFA CHES (BUPERS)

<u>Delete NCR billets</u>

55614/00110 CAPT 22NCR
55611/00100 CAPT 31NCR

<u>Rename/Change</u>

62980/40270 to 62980/44670
62477/00800 to 47689/00800 (WHMO)

<u>P1People</u>

Make majority of changes in EXCEL
- Add Headers (Titles)
- sort on BRANK (Realign/Fix top records)
- sort on ULTTITLE (Realign/Fix top records)
- sort on Rank/Desig
  - delete applicable USNRs (O4 and above)
- delete applicable other desigs
- replace RDMU/RDML with RADM
- cut and paste AQD 6-10
- sort on AQD6
  - ensure following AQDs are in the first 5 AQD fields:

950	EIT
951	PE
952	RA
960	SCW
APM	Aquisition Professional
JS7	JPME Phase I
JS8	JPME Phase II

(above step for AQD7 thru AQD10)
- delete AQD6 through AQD10

<u>Replace and update ADDU relationship</u>

44221/05005 with 62980/44585
44221/05100 with 62980/44670 (BSC 40270)
44221/05200 with 62980/44580
62477/01450 with 00011/47100 (CDR)
62477/01400 with 00011/47240 (LCDR)
62477/00800 with 47689/00800 (WHMO)
62477/05210 with 62980/44570 (BUPERS)
55614/00110 with 57034/00110 (2NCB)
55611/00100 with 57046/00100 (3NCB)

- sort on ULTEDA
  - Check each record prior to current month
- correct any NO ORDERS files
- add STUDENT to all "S" BTITLES
- correct any OVERALLOWANCES
- correct any PNDING NAVY GAINS
- delete any Non-Accession personnel
  - forward an email to Carlito Paguio to have them deleted from OAIS/OPINS

APPENDIX G: DATABASE SCHEMA

Table: ACTIVITY

Properties

DateCreated:	9/1/2002 7:58:13 PM	DefaultView:	Datasheet
Filter:	((ACTIVITY.UIC="45967"))	GUID:	{guid {B36FF00D-85D5-4EF4-85E9-0018889A76B6}}
LastUpdated:	9/20/2002 2:01:08 AM	NameMap:	Long binary data
OrderBy:	ACTIVITY.UIC	OrderByOn:	True
Orientation:	Left-to-Right	RecordCount:	496
Updatable:	True		

Columns

Name		Type	Size
ActivityID		Long Integer	4
AllowZeroLength:	False		
Attributes:	Fixed Size, Auto-Increment		
CollatingOrder:	General		
ColumnHidden:	False		
ColumnOrder:	Default		
ColumnWidth:	Default		
DataUpdatable:	False		
Description:	Primary Key		
GUID:	{guid {2C3D24C3-2EB8-424B-BABB-A0F51DD6AAC6}}		
OrdinalPosition:	0		
Required:	True		
SourceField:	ActivityID		
SourceTable:	ACTIVITY		
UIC		Text	10
AllowZeroLength:	True		
Attributes:	Variable Length		
CollatingOrder:	General		
ColumnHidden:	False		
ColumnOrder:	Default		
ColumnWidth:	Default		
DataUpdatable:	False		
Description:	Unit Identification Code		
DisplayControl:	Text Box		
GUID:	{guid {5FBEC277-C902-4BD4-B17F-AB26D1E6FDD0}}		
IMEMode:	0		
IMESentenceMode:	3		
OrdinalPosition:	1		
Required:	False		
SourceField:	UIC		
SourceTable:	ACTIVITY		
UnicodeCompression:	False		
CSC		Text	10
AllowZeroLength:	True		
Attributes:	Variable Length		
CollatingOrder:	General		
ColumnHidden:	False		
ColumnOrder:	Default		
ColumnWidth:	Default		
DataUpdatable:	False		
Description:	Command Sequence Code		
DisplayControl:	Text Box		

```
          GUID:                 {guid {1F88D4D2-A2EA-4A90-BA37-4B71B8EC8BB6}}
          IMEMode:              0
          IMESentenceMode:      3
          OrdinalPosition:      2
          Required:             False
          SourceField:          CSC
          SourceTable:          ACTIVITY
          UnicodeCompression:   False

Region                                         Text                        50
          AllowZeroLength:      True
          Attributes:           Variable Length
          CollatingOrder:       General
          ColumnHidden:         False
          ColumnOrder:          Default
          ColumnWidth:          Default
          DataUpdatable:        False
          Description:          Geographical Region
          DisplayControl:       Text Box
          GUID:                 {guid {766E2509-5167-4545-9D02-FE66CCE44251}}
          IMEMode:              0
          IMESentenceMode:      3
          OrdinalPosition:      3
          Required:             False
          SourceField:          Region
          SourceTable:          ACTIVITY
          UnicodeCompression:   False

Name                                           Text                        75
          AllowZeroLength:      True
          Attributes:           Variable Length
          CollatingOrder:       General
          ColumnHidden:         False
          ColumnOrder:          Default
          ColumnWidth:          5730
          DataUpdatable:        False
          Description:          Activity Name
          DisplayControl:       Text Box
          GUID:                 {guid {EA84F12B-C80A-4886-842F-CC8B3BA685C8}}
          IMEMode:              0
          IMESentenceMode:      3
          OrdinalPosition:      4
          Required:             False
          SourceField:          Name
          SourceTable:          ACTIVITY
          UnicodeCompression:   False

DisplayOrder                                   Long Integer                 4
          AllowZeroLength:      False
          Attributes:           Fixed Size
          CollatingOrder:       General
          ColumnHidden:         False
          ColumnOrder:          Default
          ColumnWidth:          Default
          DataUpdatable:        False
          DecimalPlaces:        Auto
          Description:          Activity Sorting Order
          DisplayControl:       Text Box
          GUID:                 {guid {D98D96FD-2D16-4555-AB4D-86083D7A2C5C}}
          OrdinalPosition:      5
          Required:             False
          SourceField:          DisplayOrder
          SourceTable:          ACTIVITY

TenDigitCode                                   Text                        10
          AllowZeroLength:      True
```

```
            Attributes:              Variable Length
            CollatingOrder:          General
            ColumnHidden:            False
            ColumnOrder:             Default
            ColumnWidth:             Default
            DataUpdatable:           False
            Description:             Ten Digit Code for Activity
            DisplayControl:          Text Box
            GUID:                    {guid {EEAD35D3-2458-453A-A759-9D8B94D8D875}}
            IMEMode:                 0
            IMESentenceMode:         3
            OrdinalPosition:         6
            Required:                False
            SourceField:             TenDigitCode
            SourceTable:             ACTIVITY
            UnicodeCompression:      False

Category                                         Text                          100
            AllowZeroLength:         True
            Attributes:              Variable Length
            CollatingOrder:          General
            ColumnHidden:            False
            ColumnOrder:             Default
            ColumnWidth:             Default
            DataUpdatable:           False
            Description:             Type of Activity
            DisplayControl:          Text Box
            GUID:                    {guid {5CF638C9-1319-47A3-A8B1-2A01CC51CABD}}
            IMEMode:                 0
            IMESentenceMode:         3
            OrdinalPosition:         7
            Required:                False
            SourceField:             Category
            SourceTable:             ACTIVITY
            UnicodeCompression:      False

Claimancy                                        Text                            2
            AllowZeroLength:         True
            Attributes:              Variable Length
            CollatingOrder:          General
            ColumnHidden:            False
            ColumnOrder:             Default
            ColumnWidth:             Default
            DataUpdatable:           False
            Description:             Major Claimant Code
            DisplayControl:          Text Box
            GUID:                    {guid {BC5F5B83-1AE5-4E6E-9A4D-64BD3E20624F}}
            IMEMode:                 0
            IMESentenceMode:         3
            OrdinalPosition:         8
            Required:                False
            SourceField:             Claimancy
            SourceTable:             ACTIVITY
            UnicodeCompression:      False

Street1                                          Text                           75
            AllowZeroLength:         True
            Attributes:              Variable Length
            CollatingOrder:          General
            ColumnHidden:            False
            ColumnOrder:             Default
            ColumnWidth:             Default
            DataUpdatable:           False
            Description:             Street Address 1
            DisplayControl:          Text Box
            GUID:                    {guid {C936471B-0A94-4AB4-A218-DC9B48CFBCED}}
```

```
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            9
          Required:                   False
          SourceField:                Street1
          SourceTable:                ACTIVITY
          UnicodeCompression:         False

Street2                                                Text                                75
          AllowZeroLength:            True
          Attributes:                 Variable Length
          CollatingOrder:             General
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          DataUpdatable:              False
          Description:                Street Address 2
          DisplayControl:             Text Box
          GUID:                       {guid {353E4507-82A5-4B6C-8C79-DA9FEB4466D9}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            10
          Required:                   False
          SourceField:                Street2
          SourceTable:                ACTIVITY
          UnicodeCompression:         False

Street3                                                Text                                75
          AllowZeroLength:            True
          Attributes:                 Variable Length
          CollatingOrder:             General
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          DataUpdatable:              False
          Description:                Street Address 3
          DisplayControl:             Text Box
          GUID:                       {guid {54C01F8C-F07C-48F3-9109-4568AA9F2BA1}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            11
          Required:                   False
          SourceField:                Street3
          SourceTable:                ACTIVITY
          UnicodeCompression:         False

Street4                                                Text                                75
          AllowZeroLength:            True
          Attributes:                 Variable Length
          CollatingOrder:             General
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          DataUpdatable:              False
          Description:                Street Address 4
          DisplayControl:             Text Box
          GUID:                       {guid {387E34D5-50FD-4CEE-B7B5-ADBF8E46D291}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            12
          Required:                   False
          SourceField:                Street4
          SourceTable:                ACTIVITY
          UnicodeCompression:         False

City                                                   Text                                25
```

```
          AllowZeroLength:        True
          Attributes:             Variable Length
          CollatingOrder:         General
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          DataUpdatable:          False
          Description:            City
          DisplayControl:         Text Box
          GUID:                   {guid {9BD13FC3-831E-4F40-94FC-3E44BA2DCCF8}}
          IMEMode:                0
          IMESentenceMode:        3
          OrdinalPosition:        13
          Required:               False
          SourceField:            City
          SourceTable:            ACTIVITY
          UnicodeCompression:     False

State                                        Text                              2
          AllowZeroLength:        True
          Attributes:             Variable Length
          CollatingOrder:         General
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          DataUpdatable:          False
          Description:            State
          DisplayControl:         Text Box
          GUID:                   {guid {3C555BBC-9F57-45DE-B9D9-5D87A8FC3C57}}
          IMEMode:                0
          IMESentenceMode:        3
          OrdinalPosition:        14
          Required:               False
          SourceField:            State
          SourceTable:            ACTIVITY
          UnicodeCompression:     False

Zip                                          Text                              5
          AllowZeroLength:        True
          Attributes:             Variable Length
          CollatingOrder:         General
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          DataUpdatable:          False
          Description:            Zip
          DisplayControl:         Text Box
          GUID:                   {guid {8DEEDBA4-37E4-4D6B-A25B-6693F268FAF2}}
          IMEMode:                0
          IMESentenceMode:        3
          OrdinalPosition:        15
          Required:               False
          SourceField:            Zip
          SourceTable:            ACTIVITY
          UnicodeCompression:     False

ZipPlusFour                                  Text                              4
          AllowZeroLength:        True
          Attributes:             Variable Length
          CollatingOrder:         General
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          DataUpdatable:          False
          Description:            Zip + 4
          DisplayControl:         Text Box
          GUID:                       {guid {A6861A71-DF8B-487C-BA07-223907E40C99}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            16
          Required:                   False
          SourceField:                ZipPlusFour
          SourceTable:                ACTIVITY
          UnicodeCompression:         False

PhoneCommercial                                           Text                          40
          AllowZeroLength:            True
          Attributes:                 Variable Length
          CollatingOrder:             General
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          DataUpdatable:              False
          Description:                Commercial Phone Number
          DisplayControl:             Text Box
          GUID:                       {guid {40524B4C-93F5-4034-B578-B20B5D2DFD99}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            17
          Required:                   False
          SourceField:                PhoneCommercial
          SourceTable:                ACTIVITY
          UnicodeCompression:         False



PhoneDSN                                                  Text                          40
          AllowZeroLength:            True
          Attributes:                 Variable Length
          CollatingOrder:             General
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          DataUpdatable:              False
          Description:                DSN Phone Number
          DisplayControl:             Text Box
          GUID:                       {guid {140BCCEF-661A-4A4B-A37F-6F177B700E32}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            18
          Required:                   False
          SourceField:                PhoneDSN
          SourceTable:                ACTIVITY
          UnicodeCompression:         False

PhoneFax                                                  Text                          40
          AllowZeroLength:            True
          Attributes:                 Variable Length
          CollatingOrder:             General
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          DataUpdatable:              False
          Description:                Fax Number
          DisplayControl:             Text Box
          GUID:                       {guid {9219A197-7812-4B10-8BEE-0747F0CDD348}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            19
          Required:                   False
          SourceField:                PhoneFax
          SourceTable:                ACTIVITY
```

UnicodeCompression:	False

Relationships

Table: ACTIVITY

ACTIVITYBILLET

ACTIVITY			BILLET
ActivityID	1	∞	ActivityID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table Indexes

Name		Number of Fields
PKC_ACTIVITY0000		1
	Clustered:	False
	DistinctCount:	496
	Foreign:	False
	IgnoreNulls:	False
	Name:	PKC_ACTIVITY0000
	Primary:	True
	Required:	True
	Unique:	True
	Fields:	
	ActivityID	Ascending
UIC		1
	Clustered:	False
	DistinctCount:	470
	Foreign:	False
	IgnoreNulls:	False
	Name:	UIC
	Primary:	False
	Required:	False
	Unique:	False
	Fields:	
	UIC	Ascending

Table: BILLET

Properties

DateCreated:	9/1/2002 7:58:13 PM	DefaultView:	Datasheet
GUID:	{guid {2818B7D7-8000-4DFD-BA10-4D00B45F2FB3}}	LastUpdated:	9/20/2002 1:58:41 AM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	1318
Updatable:	True		

Columns

Name		Type	Size
BilletID		Long Integer	4
	AllowZeroLength:	False	
	Attributes:	Fixed Size, Auto-Increment	
	CollatingOrder:	General	
	ColumnHidden:	False	

```
            ColumnOrder:          Default
            ColumnWidth:          Default
            DataUpdatable:        False
            Description:          Primary Key
            GUID:                 {guid {6A751EE3-9760-4E88-9D62-9700BDB52654}}
            OrdinalPosition:      0
            Required:             True
            SourceField:          BilletID
            SourceTable:          BILLET

ActivityID                                   Long Integer                    4
            AllowZeroLength:      False
            Attributes:           Fixed Size
            CollatingOrder:       General
            ColumnHidden:         False
            ColumnOrder:          Default
            ColumnWidth:          Default
            DataUpdatable:        False
            DecimalPlaces:        Auto
            Description:          Foreign Key to ACTIVITY table
            DisplayControl:       Text Box
            GUID:                 {guid {085CF1D2-EAF2-4CD7-82DC-6FEB53FAAEA7}}
            OrdinalPosition:      1
            Required:             True
            SourceField:          ActivityID
            SourceTable:          BILLET

BSC                                          Text                           10
            AllowZeroLength:      True
            Attributes:           Variable Length
            CollatingOrder:       General
            ColumnHidden:         False
            ColumnOrder:          Default
            ColumnWidth:          Default
            DataUpdatable:        False
            Description:          Billet Sequence Code
            DisplayControl:       Text Box
```

Table: BILLET

GUID:	{guid {13D2A66E-C2FA-4E38-A984-9315D7BCBCE4}}
IMEMode:	0
IMESentenceMode:	3
OrdinalPosition:	2
Required:	False
SourceField:	BSC
SourceTable:	BILLET
UnicodeCompression:	False

PCode	Text	7

AllowZeroLength:	True
Attributes:	Variable Length
BoundColumn:	1
CollatingOrder:	General
ColumnCount:	2
ColumnHeads:	False
ColumnHidden:	False
ColumnOrder:	Default
ColumnWidth:	Default
ColumnWidths:	576;5760
DataUpdatable:	False
Description:	Billet subspecialty code
DisplayControl:	Combo Box
GUID:	{guid {09D93CFF-E02A-4F6B-8584-ABDD1CFD27FC}}
IMEMode:	0
IMESentenceMode:	3
LimitToList:	True
ListRows:	8
ListWidth:	6336twip
OrdinalPosition:	3
Required:	False
RowSource:	SELECT lookupPCODE.PCode, lookupPCODE.Description FROM lookupPCODE ORDER BY lookupPCODE.Importance;
RowSourceType:	Table/Query
SourceField:	PCode
SourceTable:	BILLET
UnicodeCompression:	False

PCodeSuffix	Text	1

AllowZeroLength:	True
Attributes:	Variable Length
BoundColumn:	1
CollatingOrder:	General
ColumnCount:	2
ColumnHeads:	False
ColumnHidden:	False
ColumnOrder:	Default
ColumnWidth:	Default
ColumnWidths:	288;3888
DataUpdatable:	False
Description:	Billet subspecialty code suffix
DisplayControl:	Combo Box
IMEMode:	0
IMESentenceMode:	3
LimitToList:	True
ListRows:	9
ListWidth:	4176twip
OrdinalPosition:	4
Required:	False
RowSource:	SELECT lookupPCodeSuffix.PCodeSuffix, lookupPCodeSuffix.Description FROM lookupPCodeSuffix ORDER BY lookupPCodeSuffix.SortOrder;
RowSourceType:	Table/Query

```
          SourceField:            PCodeSuffix
          SourceTable:            BILLET
          UnicodeCompression:     False


PrimaryAQD                                        Text                      10
          AllowZeroLength:        True
          Attributes:             Variable Length
          BoundColumn:            1
          CollatingOrder:         General
          ColumnCount:            2
          ColumnHeads:            False
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          ColumnWidths:           432;4320
          DataUpdatable:          False
          Description:            Primary billet qualification requirement
          DisplayControl:         Combo Box
          GUID:                   {guid {EF6124D5-8875-43BA-90E6-FE95A4A362AE}}
          IMEMode:                0
          IMESentenceMode:        3
          LimitToList:            True
          ListRows:               8
          ListWidth:              4752twip
          OrdinalPosition:        5
          Required:               False
          RowSource:              SELECT QUALIFICATION.QualificationCode,
                                  QUALIFICATION.Description, QUALIFICATION.Importance
                                  FROM QUALIFICATION ORDER BY
          RowSourceType:          Table/Query
          SourceField:            PrimaryAQD
          SourceTable:            BILLET
          UnicodeCompression:     False


SecondaryAQD                                      Text                      10
          AllowZeroLength:        True
          Attributes:             Variable Length
          BoundColumn:            1
          CollatingOrder:         General
          ColumnCount:            2
          ColumnHeads:            False
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          ColumnWidths:           432;4320
          DataUpdatable:          False
          Description:            Secondary billet qualification requirement
          DisplayControl:         Combo Box
          GUID:                   {guid {4269C6BE-2C0E-42E8-9F25-023DEA3FA3BE}}
          IMEMode:                0
          IMESentenceMode:        3
          LimitToList:            True
          ListRows:               8
          ListWidth:              4752twip
          OrdinalPosition:        6
          Required:               False
          RowSource:              SELECT QUALIFICATION.QualificationCode,
                                  QUALIFICATION.Description, QUALIFICATION.Importance
                                  FROM QUALIFICATION ORDER BY
          RowSourceType:          Table/Query
          SourceField:            SecondaryAQD
          SourceTable:            BILLET
          UnicodeCompression:          False


Description                                        Text                     250
          AllowZeroLength:             True
          Attributes:                  Variable Length
          CollatingOrder:              General
          ColumnHidden:                False
          ColumnOrder:                 Default
          ColumnWidth:                 2520
          DataUpdatable:               False
          Description:                 Billet Title
          DisplayControl:              Text Box
          GUID:                        {guid {8B8FF8D4-A480-4A61-9BDB-09991CB9F6A4}}
          IMEMode:                     0
          IMESentenceMode:             3
          OrdinalPosition:             7
          Required:                    False
          SourceField:                 Description
          SourceTable:                 BILLET
          UnicodeCompression:          False

Rank                                               Text                       7
          AllowZeroLength:             True
          Attributes:                  Variable Length
          BoundColumn:                 1
          CollatingOrder:              General
          ColumnCount:                 1
          ColumnHeads:                 False
          ColumnHidden:                False
          ColumnOrder:                 Default
          ColumnWidth:                 Default
          ColumnWidths:                720
          DataUpdatable:               False
          Description:                 Billet Rank
          DisplayControl:              Combo Box
          GUID:                        {guid {9E338F7F-8D9C-4DF7-AC05-BB464CF66D69}}
          IMEMode:                     0
          IMESentenceMode:             3
          LimitToList:                 True
          ListRows:                    8
          ListWidth:                   720twip
          OrdinalPosition:             8
          Required:                    False
          RowSource:                   SELECT lookupRANK.Rank, lookupRANK.Importance FROM
                                       lookupRANK ORDER BY lookupRANK.Importance;
          RowSourceType:               Table/Query
          SourceField:                 Rank
          SourceTable:                 BILLET
          UnicodeCompression:          False


Designator                                         Text                       7
          AllowZeroLength:             True
          Attributes:                  Variable Length
          BoundColumn:                 1
          CollatingOrder:              General
          ColumnCount:                 2
          ColumnHeads:                 False
          ColumnHidden:                False
          ColumnOrder:                 Default
          ColumnWidth:                 Default
          ColumnWidths:                720;5760
          DataUpdatable:               False
          Description:                 Designator required by billet
```

DisplayControl:	Combo Box
GUID:	{guid {01C7A8AB-BF09-4E0F-8558-DAD0CBA967B4}}
IMEMode:	0
IMESentenceMode:	3
LimitToList:	True
ListRows:	8
ListWidth:	6480twip
OrdinalPosition:	9
Required:	False
RowSource:	SELECT lookupDESIGNATOR.Designator, lookupDESIGNATOR.Description, lookupDESIGNATOR.Importance FROM lookupDESIGNATOR
RowSourceType:	Table/Query
SourceField:	Designator
SourceTable:	BILLET
UnicodeCompression:	False

StartDate	Text	10
AllowZeroLength:	True	
Attributes:	Variable Length	
CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	Default	
DataUpdatable:	False	
Description:	Billet start date	
DisplayControl:	Text Box	
IMEMode:	0	
IMESentenceMode:	3	
OrdinalPosition:	10	
Required:	False	
SourceField:	StartDate	
SourceTable:	BILLET	
UnicodeCompression:	False	

EndDate	Text	10
AllowZeroLength:	True	
Attributes:	Variable Length	
CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	Default	
DataUpdatable:	False	
Description:	Billet end date	
DisplayControl:	Text Box	
IMEMode:	0	
IMESentenceMode:	3	
OrdinalPosition:	11	
Required:	False	
SourceField:	EndDate	
SourceTable:	BILLET	
UnicodeCompression:	False	

<u>Relationships</u>

ACTIVITYBILLET

ACTIVITY		**BILLET**
ActivityID	1 ∞	ActivityID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

BILLETMEMBER_BILLET

BILLET		MEMBER_BILLET
BilletID	1 ∞	BilletID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table Indexes

Name	Number of Fields
ACTIVITYBILLET	1
Clustered:	False
DistinctCount:	457
Foreign:	True
IgnoreNulls:	False
Name:	ACTIVITYBILLET
Primary:	False
Required:	False
Unique:	False

Table: BILLET

Fields:	
ActivityID	Ascending
ActivityID	1
Clustered:	False
DistinctCount:	457
Foreign:	False
IgnoreNulls:	False
Name:	ActivityID
Primary:	False
Required:	False
Unique:	False
Fields:	
ActivityID	Ascending
PKC_BILLET0002	1
Clustered:	False
DistinctCount:	1318
Foreign:	False
IgnoreNulls:	False
Name:	PKC_BILLET0002
Primary:	True
Required:	True
Unique:	True
Fields:	
BilletID	Ascending

Table: MEMBER

Properties

DateCreated:	9/1/2002 7:58:13 PM	DefaultView:	Datasheet
GUID:	{guid {B6600329-BDBE-4117-944E-BBE73A5FC563}}	LastUpdated:	9/20/2002 1:56:52 AM
NameMap:	Long binary data	OrderByOn:	True
Orientation:	Left-to-Right	RecordCount:	1347
Updatable:	True		

Columns

Name	Type	Size
MemberID	Long Integer	4

AllowZeroLength:	False
Attributes:	Fixed Size, Auto-Increment
CollatingOrder:	General
ColumnHidden:	False
ColumnOrder:	Default
ColumnWidth:	Default
DataUpdatable:	False
Description:	Primary Key
GUID:	{guid {24083BE9-EFF3-4690-9589-5A60B376AF01}}
OrdinalPosition:	0
Required:	True
SourceField:	MemberID
SourceTable:	MEMBER

Name	Type	Size
Designator	Text	7

AllowZeroLength:	True
Attributes:	Variable Length
BoundColumn:	1
CollatingOrder:	General
ColumnCount:	2
ColumnHeads:	False
ColumnHidden:	False
ColumnOrder:	Default
ColumnWidth:	Default
ColumnWidths:	576;5760
DataUpdatable:	False
Description:	Officer's Designator
DisplayControl:	Combo Box
GUID:	{guid {22C55386-052A-4F01-9E7E-875DF0EEE6DC}}
IMEMode:	0
IMESentenceMode:	3
LimitToList:	True
ListRows:	8
ListWidth:	6336twip
OrdinalPosition:	1
Required:	False
RowSource:	SELECT lookupDESIGNATOR.Designator, lookupDESIGNATOR.Description, lookupDESIGNATOR.Importance FROM lookupDESIGNATOR
RowSourceType:	Table/Query
SourceField:	Designator
SourceTable:	MEMBER
UnicodeCompression:	False

Name	Type	Size
Rank	Text	7

AllowZeroLength:	True
Attributes:	Variable Length
BoundColumn:	1
CollatingOrder:	General
ColumnCount:	1
ColumnHeads:	False
ColumnHidden:	False
ColumnOrder:	Default
ColumnWidth:	Default
ColumnWidths:	720
DataUpdatable:	False
Description:	Rank
DisplayControl:	Combo Box
GUID:	{guid {88BFAC4D-CEE1-44D8-A875-EF6FD85BC05B}}
IMEMode:	0
IMESentenceMode:	3
LimitToList:	True
ListRows:	16

	ListWidth:	720twip	
	OrdinalPosition:	2	
	Required:	False	
	RowSource:	SELECT lookupRANK.Rank, lookupRANK.Importance FROM lookupRANK ORDER BY lookupRANK.Importance;	
	RowSourceType:	Table/Query	
	SourceField:	Rank	
	SourceTable:	MEMBER	
	UnicodeCompression:	False	
LastName		Text	25
	AllowZeroLength:	True	
	Attributes:	Variable Length	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	Description:	Last Name	
	DisplayControl:	Text Box	
	GUID:	{guid {E9CF0FF9-55EA-444C-98C1-9650DF13B56F}}	
	IMEMode:	0	
	IMESentenceMode:	3	
	OrdinalPosition:	3	
	Required:	True	
	SourceField:	LastName	
	SourceTable:	MEMBER	
	UnicodeCompression:	False	
FirstName		Text	25
	AllowZeroLength:	True	
	Attributes:	Variable Length	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	Description:	First Name	
	DisplayControl:	Text Box	
	GUID:	{guid {6AAC0D95-1011-4C3B-BF04-F75A46F412BC}}	
	IMEMode:	0	
	IMESentenceMode:	3	
	OrdinalPosition:	4	
	Required:	True	
	SourceField:	FirstName	
	SourceTable:	MEMBER	
	UnicodeCompression:	False	
MiddleNameSuffix		Text	50
	AllowZeroLength:	True	
	Attributes:	Variable Length	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	Description:	Middle Name with Suffix (jr, sr, etc)	
	DisplayControl:	Text Box	
	GUID:	{guid {565A4755-CF61-4AD8-A0D6-2F997ECCB858}}	
	IMEMode:	0	
	IMESentenceMode:	3	
	OrdinalPosition:	5	
	Required:	False	
	SourceField:	MiddleNameSuffix	
	SourceTable:	MEMBER	

```
          UnicodeCompression:        False

NickName                                             Text                           25
          AllowZeroLength:           True
          Attributes:                Variable Length
          CollatingOrder:            General
          ColumnHidden:              False
          ColumnOrder:               Default
          ColumnWidth:               Default
          DataUpdatable:             False
          Description:               Nickname if applicable
          DisplayControl:            Text Box
          GUID:                      {guid {7C40F997-0872-459A-9775-98ADE600EF95}}
          IMEMode:                   0
          IMESentenceMode:           3
          OrdinalPosition:           6
          Required:                  False
          SourceField:               NickName
          SourceTable:               MEMBER
          UnicodeCompression:        False

SSN                                                  Text                            9
          AllowZeroLength:           True
          Attributes:                Variable Length
          CollatingOrder:            General
          ColumnHidden:              False
          ColumnOrder:               Default
          ColumnWidth:               Default
          DataUpdatable:             False
          Description:               Social Security Number
          DisplayControl:            Text Box
          GUID:                      {guid {31DC660E-3EAD-4658-844D-2EE1E8915EEE}}
          IMEMode:                   0
          IMESentenceMode:           3
          OrdinalPosition:           8
          Required:                  True
          SourceField:               SSN
          SourceTable:               MEMBER
          UnicodeCompression:        False

Ethnic                                               Text                            5
          AllowZeroLength:           True
          Attributes:                Variable Length
          CollatingOrder:            General
          ColumnHidden:              False
          ColumnOrder:               Default
          ColumnWidth:               Default
          DataUpdatable:             False
          Description:               Ethnic Group
          DisplayControl:            Text Box
          GUID:                      {guid {0F5D4751-6994-475E-8E78-3DEFE2B0C636}}
          IMEMode:                   0
          IMESentenceMode:           3
          OrdinalPosition:           9
          Required:                  False
          SourceField:               Ethnic
          SourceTable:               MEMBER
          UnicodeCompression:        False

Race                                                 Text                            1
          AllowZeroLength:           True
          Attributes:                Variable Length
          CollatingOrder:            General
          ColumnHidden:              False
          ColumnOrder:               Default
          ColumnWidth:               Default
          DataUpdatable:          False
          Description:            Race
          DisplayControl:         Text Box
          GUID:                   {guid {D27D250B-395B-416D-B451-D888D555D470}}
          IMEMode:                0
          IMESentenceMode:        3
          OrdinalPosition:        10
          Required:               False
          SourceField:            Race
          SourceTable:            MEMBER
          UnicodeCompression:     False

Sex                                                Text                      1
          AllowZeroLength:        True
          Attributes:             Variable Length
          BoundColumn:            1
          CollatingOrder:         General
          ColumnCount:            1
          ColumnHeads:            False
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          DataUpdatable:          False
          Description:            Sex
          DisplayControl:         Combo Box
          GUID:                   {guid {B6BA6750-005D-4C24-812A-03B87E408A15}}
          IMEMode:                0
          IMESentenceMode:        3
          LimitToList:            False
          ListRows:               8
          ListWidth:              0twip
          OrdinalPosition:        11
          Required:               False
          RowSource:              M;F
          RowSourceType:          Value List
          SourceField:            Sex
          SourceTable:            MEMBER
          UnicodeCompression:     False

YearGroup                                          Text                      4
          AllowZeroLength:        True
          Attributes:             Variable Length
          CollatingOrder:         General
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          DataUpdatable:          False
          Description:            Year Group
          DisplayControl:         Text Box
          GUID:                   {guid {BB3E459D-42A0-4D9C-BE18-458054F63C87}}
          IMEMode:                0
          IMESentenceMode:        3
          OrdinalPosition:        12
          Required:               False
          SourceField:            YearGroup
          SourceTable:            MEMBER
          UnicodeCompression:     False

PrecedenceNumber                                   Text                     20
          AllowZeroLength:        True
          Attributes:             Variable Length
          CollatingOrder:         General
          ColumnHidden:           False
          ColumnOrder:            Default
          ColumnWidth:            Default
          DataUpdatable:          False
          Description:                Member's precedence number
          DisplayControl:             Text Box
          GUID:                       {guid {310B719C-71CD-44C6-A45C-8EBE355E12DF}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            13
          Required:                   False
          SourceField:                PrecedenceNumber
          SourceTable:                MEMBER
          UnicodeCompression:         False

PromotionStatus                                           Text                              6
          AllowZeroLength:            True
          Attributes:                 Variable Length
          CollatingOrder:             General
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          DataUpdatable:              False
          Description:                Member's promotion status
          DisplayControl:             Text Box
          GUID:                       {guid {190977A9-2E98-475C-9BFB-C76130E67647}}
          IMEMode:                    0
          IMESentenceMode:            3
          OrdinalPosition:            14
          Required:                   False
          SourceField:                PromotionStatus
          SourceTable:                MEMBER
          UnicodeCompression:         False

PCode                                                     Text                              7
          AllowZeroLength:            True
          Attributes:                 Variable Length
          BoundColumn:                1
          CollatingOrder:             General
          ColumnCount:                2
          ColumnHeads:                False
          ColumnHidden:               False
          ColumnOrder:                Default
          ColumnWidth:                Default
          ColumnWidths:               720;5760
          DataUpdatable:              False
          Description:                Member's subspecialty code
          DisplayControl:             Combo Box
          GUID:                       {guid {0B2C7A1E-AB10-44E8-8520-FC78DC82030C}}
          IMEMode:                    0
          IMESentenceMode:            3
          LimitToList:                True
          ListRows:                   8
          ListWidth:                  6480twip
          OrdinalPosition:            15
          Required:                   False
          RowSource:                  SELECT lookupPCODE.PCode, lookupPCODE.Description,
                                      lookupPCODE.Importance FROM lookupPCODE ORDER BY
                                      lookupPCODE.Importance;
          RowSourceType:              Table/Query
          SourceField:                PCode
          SourceTable:                MEMBER
          UnicodeCompression:         False


PCodeSuffix                                               Text                              2
          AllowZeroLength:            True
          Attributes:                 Variable Length
          BoundColumn:                1
```

```
            CollatingOrder:             General
            ColumnCount:                2
            ColumnHeads:                False
            ColumnHidden:               False
            ColumnOrder:                Default
            ColumnWidth:                Default
            ColumnWidths:               288;3888
            DataUpdatable:              False
            Description:                Subspecialty code suffix (P, T, etc)
            DisplayControl:             Combo Box
            GUID:                       {guid {D5B2C84A-C651-42F8-88D1-A5A4A14695FF}}
            IMEMode:                    0
            IMESentenceMode:            3
            LimitToList:                True
            ListRows:                   9
            ListWidth:                  4176twip
            OrdinalPosition:            16
            Required:                   False
            RowSource:                  SELECT lookupPCodeSuffix.PCodeSuffix,
                                        lookupPCodeSuffix.Description FROM lookupPCodeSuffix ORDER
                                        BY lookupPCodeSuffix.SortOrder;
            RowSourceType:              Table/Query
            SourceField:                PCodeSuffix
            SourceTable:                MEMBER
            UnicodeCompression:         False


HomeAddress1                                          Text                          75
            AllowZeroLength:            True
            Attributes:                 Variable Length
            CollatingOrder:             General
            ColumnHidden:               False
            ColumnOrder:                Default
            ColumnWidth:                Default
            DataUpdatable:              False
            Description:                Street Address Line 1
            DisplayControl:             Text Box
            GUID:                       {guid {BD4B2A18-1651-4833-9752-65DF5673B1C8}}
            IMEMode:                    0
            IMESentenceMode:            3
            OrdinalPosition:            17
            Required:                   False
            SourceField:                HomeAddress1
            SourceTable:                MEMBER
            UnicodeCompression:         False

HomeAddress2                                          Text                          75
            AllowZeroLength:            True
            Attributes:                 Variable Length
            CollatingOrder:             General
            ColumnHidden:               False
            ColumnOrder:                Default
            ColumnWidth:                Default
            DataUpdatable:              False
            Description:                Street Address Line 2
            DisplayControl:             Text Box
            GUID:                       {guid {E140C7F1-D007-470D-9F63-52174658353C}}
            IMEMode:                    0
            IMESentenceMode:            3
            OrdinalPosition:            18
            Required:                   False
            SourceField:                HomeAddress2
            SourceTable:                MEMBER
            UnicodeCompression:         False
```

```
HomeCity                                              Text                           25
          AllowZeroLength:             True
          Attributes:                  Variable Length
          CollatingOrder:              General
          ColumnHidden:                False
          ColumnOrder:                 Default
          ColumnWidth:                 Default
          DataUpdatable:               False
          Description:                 City
          DisplayControl:              Text Box
          GUID:                        {guid {915C4633-4234-4175-8020-3008F05426B3}}
          IMEMode:                     0
          IMESentenceMode:             3
          OrdinalPosition:             19
          Required:                    False
          SourceField:                 HomeCity
          SourceTable:                 MEMBER
          UnicodeCompression:          False

HomeState                                             Text                            2
          AllowZeroLength:             True
          Attributes:                  Variable Length
          CollatingOrder:              General
          ColumnHidden:                False
          ColumnOrder:                 Default
          ColumnWidth:                 Default
          DataUpdatable:               False
          Description:                 State
          DisplayControl:              Text Box
          GUID:                        {guid {8039BC9D-772D-418D-8CC2-BB2E02F0B663}}
          IMEMode:                     0
          IMESentenceMode:             3
          OrdinalPosition:             20
          Required:                    False
          SourceField:                 HomeState
          SourceTable:                 MEMBER
          UnicodeCompression:          False

HomeZip                                               Text                            5
          AllowZeroLength:             True
          Attributes:                  Variable Length
          CollatingOrder:              General
          ColumnHidden:                False
          ColumnOrder:                 Default
          ColumnWidth:                 Default
          DataUpdatable:               False
          Description:                 Zip
          DisplayControl:              Text Box
          GUID:                        {guid {42BFB645-3A02-456A-A93B-74FE7C244D39}}
          IMEMode:                     0
```

```
            IMESentenceMode:           3
            OrdinalPosition:           21
            Required:                  False
            SourceField:               HomeZip
            SourceTable:               MEMBER
            UnicodeCompression:        False

HomeZipPlusFour                                      Text                        4
            AllowZeroLength:           True
            Attributes:                Variable Length
            CollatingOrder:            General
            ColumnHidden:              False
            ColumnOrder:               Default
            ColumnWidth:               Default
            DataUpdatable:             False
            Description:               Zip Plus 4
            DisplayControl:            Text Box
            GUID:                      {guid {DBAACA0C-B9A5-448C-8214-917C5B00B607}}
            IMEMode:                   0
            IMESentenceMode:           3
            OrdinalPosition:           22
            Required:                  False
            SourceField:               HomeZipPlusFour
            SourceTable:               MEMBER
            UnicodeCompression:        False

PhoneHome                                            Text                       20
            AllowZeroLength:           True
            Attributes:                Variable Length
            CollatingOrder:            General
            ColumnHidden:              False
            ColumnOrder:               Default
            ColumnWidth:               Default
            DataUpdatable:             False
            Description:               Home phone
            DisplayControl:            Text Box
            GUID:                      {guid {98BF08ED-E6C5-4E58-A5FB-AB25B19D9BAF}}
            IMEMode:                   0
            IMESentenceMode:           3
            OrdinalPosition:           23
            Required:                  False
            SourceField:               PhoneHome
            SourceTable:               MEMBER
            UnicodeCompression:        False

PhoneCell                                            Text                       20
            AllowZeroLength:           True
            Attributes:                Variable Length
            CollatingOrder:            General
            ColumnHidden:              False
            ColumnOrder:               Default
            ColumnWidth:               Default
            DataUpdatable:             False
            Description:               Cell phone
            DisplayControl:            Text Box
            GUID:                      {guid {89A989A1-45D9-47AD-AFC2-9B39B4C1CD25}}
            IMEMode:                   0
            IMESentenceMode:           3
            OrdinalPosition:           24
            Required:                  False
            SourceField:               PhoneCell
            SourceTable:               MEMBER
            UnicodeCompression:        False
```

```
PhoneWorkComm                                  Text                        20
          AllowZeroLength:          True
          Attributes:               Variable Length
          CollatingOrder:           General
          ColumnHidden:             False
          ColumnOrder:              Default
          ColumnWidth:              Default
          DataUpdatable:            False
          Description:              Commercial phone
          DisplayControl:           Text Box
          GUID:                     {guid {1A288609-E377-444B-8D24-12D0A0A0BE36}}
          IMEMode:                  0
          IMESentenceMode:          3
          OrdinalPosition:          25
          Required:                 False
          SourceField:              PhoneWorkComm
          SourceTable:              MEMBER
          UnicodeCompression:       False

PhoneWorkCommExt                               Text                         8
          AllowZeroLength:          True
          Attributes:               Variable Length
          CollatingOrder:           General
          ColumnHidden:             False
          ColumnOrder:              Default
          ColumnWidth:              Default
          DataUpdatable:            False
          Description:              Commercial phone extension
          DisplayControl:           Text Box
          GUID:                     {guid {BC11FE94-CA2D-48F3-A4C4-1B5408EE3312}}
          IMEMode:                  0
          IMESentenceMode:          3
          OrdinalPosition:          26
          Required:                 False
          SourceField:              PhoneWorkCommExt
          SourceTable:              MEMBER
          UnicodeCompression:       False

PhoneWorkDSN                                   Text                        10
          AllowZeroLength:          True
          Attributes:               Variable Length
          CollatingOrder:           General
          ColumnHidden:             False
          ColumnOrder:              Default
          ColumnWidth:              Default
          DataUpdatable:            False
          Description:              DSN
          DisplayControl:           Text Box
          GUID:                     {guid {286B697D-7D11-446F-9BBC-6A4F84C4BF39}}
          IMEMode:                  0
          IMESentenceMode:          3
          OrdinalPosition:          27
          Required:                 False
          SourceField:              PhoneWorkDSN
          SourceTable:              MEMBER
          UnicodeCompression:       False

PhoneWorkDSNExt                                Text                         8
          AllowZeroLength:          True
          Attributes:               Variable Length
          CollatingOrder:           General
          ColumnHidden:             False
          ColumnOrder:              Default
          ColumnWidth:              Default
          DataUpdatable:            False
          Description:              DSN extension
```

DisplayControl:	Text Box
GUID:	{guid {211993E7-8B54-40AD-AD25-2D6E6DA1228D}}
IMEMode:	0
IMESentenceMode:	3
OrdinalPosition:	28
Required:	False
SourceField:	PhoneWorkDSNExt
SourceTable:	MEMBER
UnicodeCompression:	False

Pager	Text	40
AllowZeroLength:	True	
Attributes:	Variable Length	
CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	Default	
DataUpdatable:	False	
Description:	Pager	
DisplayControl:	Text Box	
GUID:	{guid {2294BA1D-BF61-48BF-A2FA-C82D4F21B4C1}}	
IMEMode:	0	
IMESentenceMode:	3	
OrdinalPosition:	29	
Required:	False	
SourceField:	Pager	
SourceTable:	MEMBER	
UnicodeCompression:	False	

EmailWork	Text	100
AllowZeroLength:	True	
Attributes:	Variable Length	
CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	Default	
DataUpdatable:	False	
Description:	Work email address	
DisplayControl:	Text Box	
GUID:	{guid {1FAD47A5-B444-46A6-A4B5-79F3A28F5DA9}}	
IMEMode:	0	
IMESentenceMode:	3	
OrdinalPosition:	30	
Required:	False	
SourceField:	EmailWork	
SourceTable:	MEMBER	
UnicodeCompression:	False	

EmailHome	Text	100
AllowZeroLength:	True	
Attributes:	Variable Length	
CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	Default	
DataUpdatable:	False	
Description:	Home email address	
DisplayControl:	Text Box	
GUID:	{guid {68D995A1-7058-4BC2-8957-2DE7E47D1FC7}}	
IMEMode:	0	
IMESentenceMode:	3	
OrdinalPosition:	31	
Required:	False	
SourceField:	EmailHome	
SourceTable:	MEMBER	
UnicodeCompression:	False	

EmailAlternate	Text	100

AllowZeroLength:	True
Attributes:	Variable Length
CollatingOrder:	General
ColumnHidden:	False
ColumnOrder:	Default
ColumnWidth:	Default
DataUpdatable:	False
Description:	Alternate email address
DisplayControl:	Text Box
GUID:	{guid {07890FF1-BB78-4466-92EA-E9883146E416}}
IMEMode:	0
IMESentenceMode:	3
OrdinalPosition:	32
Required:	False
SourceField:	EmailAlternate
SourceTable:	MEMBER
UnicodeCompression:	False

Relationships

MEMBERMEMBER_BILLET

MEMBER			MEMBER_BILLET
MemberID	1	∞	MemberID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table: MEMBER

MEMBERMEMBER_QUALIFICATION

MEMBER			MEMBER_QUALIFICA
MemberID	1	∞	MemberID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

MEMBERPREFERENCES

MEMBER			PREFERENCES
MemberID	1	∞	MemberID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table Indexes

Name	Number of Fields
MemberIDLastNameFirstNameMiddleNa	4

Clustered:	False
DistinctCount:	1347
Foreign:	False
IgnoreNulls:	False
Name:	MemberIDLastNameFirstNameMiddleNameSuffix
Primary:	False
Required:	False

	Unique:	False
	Fields:	
	MemberID	Ascending
	LastName	Ascending
	FirstName	Ascending
	MiddleNameSuffix	Ascending
PKC_MEMBER0003		1
	Clustered:	False
	DistinctCount:	1347
	Foreign:	False
	IgnoreNulls:	False
	Name:	PKC_MEMBER0003
	Primary:	True
	Required:	True
	Unique:	True
	Fields:	
	MemberID	Ascending
Rank		1
	Clustered:	False
	DistinctCount:	12
	Foreign:	False
	IgnoreNulls:	False

Table: MEMBER

Name:	Rank
Primary:	False
Required:	False
Unique:	False
Fields:	
Rank	Ascending

Table: MEMBER_BILLET

Properties

DateCreated:	9/1/2002 7:58:13 PM	DefaultView:	Datasheet
GUID:	{guid {F807CE2E-D659-4E18-BE98-9C6A837F8F7E}}	LastUpdated:	9/20/2002 2:03:00 AM
NameMap:	Long binary data	OrderByOn:	True
Orientation:	Left-to-Right	RecordCount:	1521
Updatable:	True		

Columns

Name		Type	Size
MemberBilletID		Long Integer	4
	AllowZeroLength:	False	
	Attributes:	Fixed Size, Auto-Increment	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	Description:	Primary Key (Surrogate Key)	
	GUID:	{guid {4D055002-40C7-469B-9F91-21A902424D34}}	
	OrdinalPosition:	0	
	Required:	True	
	SourceField:	MemberBilletID	
	SourceTable:	MEMBER_BILLET	
MemberID		Long Integer	4
	AllowZeroLength:	False	
	Attributes:	Fixed Size	
	CollatingOrder:	General	

```
          ColumnHidden:            False
          ColumnOrder:             Default
          ColumnWidth:             Default
          DataUpdatable:           False
          DecimalPlaces:           Auto
          Description:             Foreign Key from MEMBER
          DisplayControl:          Text Box
          GUID:                    {guid {0BA40445-5D45-44BA-BFAD-C9E73C0A6B3B}}
          OrdinalPosition:         1
          Required:                True
          SourceField:             MemberID
          SourceTable:             MEMBER_BILLET

BilletID                                          Long Integer                    4
          AllowZeroLength:         False
          Attributes:              Fixed Size
          CollatingOrder:          General
          ColumnHidden:            False
          ColumnOrder:             Default
          ColumnWidth:             Default
          DataUpdatable:           False
          DecimalPlaces:           Auto
          Description:             Foreign Key f rom BILLET
          DisplayControl:          Text Box
          GUID:                    {guid {CC0B45A7-5D0B-47CC-847D-65BB065BC313}}
          OrdinalPosition:         2
          Required:                True
          SourceField:             BilletID
          SourceTable:             MEMBER_BILLET

ReportDate                                        Date/Time                       8
          AllowZeroLength:         False
          Attributes:              Fixed Size
          CollatingOrder:          General
          ColumnHidden:            False
          ColumnOrder:             Default
          ColumnWidth:             Default
          DataUpdatable:           False
          Description:             Date Member Reports for Duty
          GUID:                    {guid {8B67AF36-BEDF-4415-96AB-F53F80DFB673}}
          IMEMode:                 0
          IMESentenceMode:         3
          OrdinalPosition:         3
          Required:                False
          SourceField:             ReportDate
          SourceTable:             MEMBER_BILLET

PRD                                               Date/Time                       8
          AllowZeroLength:         False
          Attributes:              Fixed Size
          CollatingOrder:          General
          ColumnHidden:            False
          ColumnOrder:             Default
          ColumnWidth:             Default
          DataUpdatable:           False
          Description:             Projected Rotation Date
          GUID:                    {guid {56EF83A4-CA09-40BE-82A1-419772E9C0A6}}
          IMEMode:                 0
          IMESentenceMode:         3
          OrdinalPosition:         4
          Required:                False
          SourceField:             PRD
          SourceTable:             MEMBER_BILLET

Type                                              Text                           10
          AllowZeroLength:         True
```

Attributes:	Variable Length
BoundColumn:	1
CollatingOrder:	General
ColumnCount:	1
ColumnHeads:	False
ColumnHidden:	False
ColumnOrder:	Default
ColumnWidth:	Default
DataUpdatable:	False
Description:	Type of Billet (Current, Previous, Ultimate)
DisplayControl:	Combo Box
GUID:	{guid {C994F4B4-56AE-4762-8BE6-15E624638F85}}
IMEMode:	0
IMESentenceMode:	3

Table: MEMBER_BILLET

LimitToList:	True
ListRows:	8
ListWidth:	0twip
OrdinalPosition:	6
Required:	True
RowSource:	"Previous";"Current";"Ultimate"
RowSourceType:	Value List
SourceField:	Type
SourceTable:	MEMBER_BILLET
UnicodeCompression:	False

Relationships

BILLETMEMBER_BILLET

BILLET			**MEMBER_BILLET**
BilletID	1	∞	BilletID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

MEMBERMEMBER_BILLET

MEMBER			**MEMBER_BILLET**
MemberID	1	∞	MemberID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table Indexes

Name	Number of Fields
BILLETMEMBER_BILLET	1

Clustered:	False
DistinctCount:	1207
Foreign:	True
IgnoreNulls:	False
Name:	BILLETMEMBER_BILLET
Primary:	False
Required:	False
Unique:	False
Fields:	
BilletID	Ascending

MEMBERMEMBER_BILLET	1
Clustered:	False
DistinctCount:	1347
Foreign:	True
IgnoreNulls:	False
Name:	MEMBERMEMBER_BILLET
Primary:	False
Required:	False
Unique:	False
Fields:	
MemberID	Ascending
PKC_MEMBER_BILLET0006	1
Clustered:	False
DistinctCount:	1521
Foreign:	False
IgnoreNulls:	False
Name:	PKC_MEMBER_BILLET0006
Primary:	True
Required:	True
Unique:	True
Fields:	
MemberBilletID	Ascending

Table: MEMBER_QUALIFICATION

Properties

DateCreated:	9/1/2002 7:58:13 PM	DefaultView:	Datasheet
GUID:	{guid {8DF4A080-E7B6-4922-9323-9472C4DA2321}}	LastUpdated:	9/20/2002 2:04:50 AM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	4322
Updatable:	True		

Columns

Name		Type	Size
MemberQualificationID		Long Integer	4
AllowZeroLength:	False		
Attributes:	Fixed Size, Auto-Increment		
CollatingOrder:	General		
ColumnHidden:	False		
ColumnOrder:	Default		
ColumnWidth:	Default		
DataUpdatable:	False		
Description:	Primary Key (Surrogate Key)		
GUID:	{guid {34F75407-C81E-4756-A7D2-F43BC4E1BBD0}}		
OrdinalPosition:	0		
Required:	True		
SourceField:	MemberQualificationID		
SourceTable:	MEMBER_QUALIFICATION		
MemberID		Long Integer	4
AllowZeroLength:	False		
Attributes:	Fixed Size		
CollatingOrder:	General		
ColumnHidden:	False		
ColumnOrder:	Default		
ColumnWidth:	Default		
DataUpdatable:	False		
DecimalPlaces:	Auto		
Description:	Foreign Key from MEMBER table		
DisplayControl:	Text Box		
GUID:	{guid {AD070369-E520-4C41-BB0F-74D529E4268F}}		

	OrdinalPosition:	1	
	Required:	True	
	SourceField:	MemberID	
	SourceTable:	MEMBER_QUALIFICATION	
QualificationID		Long Integer	4
	AllowZeroLength:	False	
	Attributes:	Fixed Size	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	DecimalPlaces:	Auto	
	Description:	Foreign Key from QUALIFICATION table	

Table: MEMBER_QUALIFICATION

	DisplayControl:	Text Box	
	GUID:	{guid {FCFE6148-E987-4587-B99B-115681EE6CA2}}	
	OrdinalPosition:	2	
	Required:	True	
	SourceField:	QualificationID	
	SourceTable:	MEMBER_QUALIFICATION	
Display		Yes/No	1
	AllowZeroLength:	False	
	Attributes:	Fixed Size	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	Description:	Display this qualification in the P-1?	
	DisplayControl:	106	
	Format:	Yes/No	
	GUID:	{guid {050835CD-C374-4C05-ABBC-FCA25B3C4692}}	
	OrdinalPosition:	3	
	Required:	False	
	SourceField:	Display	
	SourceTable:	MEMBER_QUALIFICATION	
Order		Long Integer	4
	AllowZeroLength:	False	
	Attributes:	Fixed Size	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	DecimalPlaces:	Auto	
	Description:	Sorting Order	
	DisplayControl:	Text Box	
	GUID:	{guid {7D71FBB0-5C5C-463F-BC30-0EDFE79D2083}}	
	OrdinalPosition:	4	
	Required:	False	
	SourceField:	Order	
	SourceTable:	MEMBER_QUALIFICATION	

<u>Relationships</u>

MEMBERMEMBER_QUALIFICATION

MEMBER			MEMBER_QUALIFICA
MemberID	1	∞	MemberID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table: MEMBER_QUALIFICATION

QUALIFICATIONMEMBER_QUALIFICATION

QUALIFICATION			MEMBER_QUALIFICA
QualificationID	1	∞	QualificationID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table Indexes

Name	Number of Fields
MEMBERMEMBER_QUALIFICATION	1
Clustered:	False
DistinctCount:	1214
Foreign:	True
IgnoreNulls:	False
Name:	MEMBERMEMBER_QUALIFICATION
Primary:	False
Required:	False
Unique:	False
Fields:	
MemberID	Ascending
PKC_MEMBER_QUALIFICATION0009	1
Clustered:	False
DistinctCount:	4322
Foreign:	False
IgnoreNulls:	False
Name:	PKC_MEMBER_QUALIFICATION0009
Primary:	True
Required:	True
Unique:	True
Fields:	
MemberQualificationID	Ascending
QUALIFICATIONMEMBER_QUALIFIC	1
Clustered:	False
DistinctCount:	77
Foreign:	True
IgnoreNulls:	False
Name:	QUALIFICATIONMEMBER_QUALIFICATION
Primary:	False
Required:	False
Unique:	False
Fields:	
QualificationID	Ascending

Table: PREFERENCES

Properties

DateCreated:	9/1/2002 7:58:13 PM	DefaultView:	Datasheet
GUID:	{guid {BE9E111D-F0AD-4D2E-8FF5-557B39B859FF}}	LastUpdated:	9/20/2002 2:07:14 AM
NameMap:	Long binary data	OrderByOn:	False
Orientation:	Left-to-Right	RecordCount:	0
Updatable:	True		

Columns

```
Name                                                        Type                          Size
PreferenceID                                                Long Integer                     4
          AllowZeroLength:              False
          Attributes:                   Fixed Size, Auto-Increment
          CollatingOrder:               General
          ColumnHidden:                 False
          ColumnOrder:                  Default
          ColumnWidth:                  Default
          DataUpdatable:                False
          Description:                  Primary Key
          GUID:                         {guid {FFA39BA2-EF83-426F-9D53-1FAFAD53ADB2}}
          OrdinalPosition:              0
          Required:                     True
          SourceField:                  PreferenceID
          SourceTable:                  PREFERENCES

MemberID                                                    Long Integer                     4
          AllowZeroLength:              False
          Attributes:                   Fixed Size
          CollatingOrder:               General
          ColumnHidden:                 False
          ColumnOrder:                  Default
          ColumnWidth:                  Default
          DataUpdatable:                False
          DecimalPlaces:                Auto
          Description:                  Foreign Key
          DisplayControl:               Text Box
          GUID:                         {guid {ABB979C0-73F5-43C2-9CAE-C0A0C19AE623}}
          OrdinalPosition:              1
          Required:                     True
          SourceField:                  MemberID
          SourceTable:                  PREFERENCES

DateSubmitted                                               Date/Time                        8
          AllowZeroLength:              False
          Attributes:                   Fixed Size
          CollatingOrder:               General
          ColumnHidden:                 False
          ColumnOrder:                  Default
          ColumnWidth:                  Default
          DataUpdatable:                False
          Description:                  Date Preference was submitted
          GUID:                         {guid {3E62BDFA-C75F-4A65-B26B-A5F03A182390}}
          IMEMode:                      0
          IMESentenceMode:              3
          OrdinalPosition:              2
          Required:                     False
          SourceField:                  DateSubmitted
          SourceTable:                  PREFERENCES

Description                                                 Text                           250
          AllowZeroLength:              True
          Attributes:                   Variable Length
          CollatingOrder:               General
          ColumnHidden:                 False
          ColumnOrder:                  Default
          ColumnWidth:                  Default
          DataUpdatable:                False
          Description:                  Description of preference request
          DisplayControl:               Text Box
          GUID:                         {guid {089D8669-8B3E-446D-BB05-A0A14246C7F4}}
          IMEMode:                      0
          IMESentenceMode:              3
          OrdinalPosition:              3
          Required:                     False
          SourceField:                  Description
```

	SourceTable:	PREFERENCES	
	UnicodeCompression:	False	
DetailerNotes		Text	250
	AllowZeroLength:	True	
	Attributes:	Variable Length	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	Description:	Comments made by the detailer	
	DisplayControl:	Text Box	
	GUID:	{guid {DAA0F385-0BFE-416B-990A-EEADDE8AC34F}}	
	IMEMode:	0	
	IMESentenceMode:	3	
	OrdinalPosition:	4	
	Required:	False	
	SourceField:	DetailerNotes	
	SourceTable:	PREFERENCES	
	UnicodeCompression:	False	
Priority		Long Integer	4
	AllowZeroLength:	False	
	Attributes:	Fixed Size	
	CollatingOrder:	General	
	ColumnHidden:	False	
	ColumnOrder:	Default	
	ColumnWidth:	Default	
	DataUpdatable:	False	
	DecimalPlaces:	Auto	
	Description:	Member's priority assigned to the preference	
	DisplayControl:	Text Box	
	GUID:	{guid {C4C04B06-5561-4849-9E6A-0D9530EFEBA5}}	
	OrdinalPosition:	5	
	Required:	False	
	SourceField:	Priority	
	SourceTable:	PREFERENCES	

Relationships

MEMBERPREFERENCES

MEMBER			PREFERENCES
MemberID	1	∞	MemberID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table Indexes

Name		Number of Fields
MEMBERPREFERENCES		1
	Clustered:	False
	DistinctCount:	0
	Foreign:	True
	IgnoreNulls:	False
	Name:	MEMBERPREFERENCES
	Primary:	False
	Required:	False
	Unique:	False
	Fields:	

MemberID	Ascending

PKC_PREFERENCES000B	1
Clustered:	False
DistinctCount:	0
Foreign:	False
IgnoreNulls:	False
Name:	PKC_PREFERENCES000B
Primary:	True
Required:	True
Unique:	True
Fields:	
PreferenceID	Ascending

Table: QUALIFICATION

Properties

DateCreated:	9/1/2002 7:58:13 PM	DefaultView:	Datasheet
GUID:	{guid {45F85DF2-D393-45C2-8BB6-B3830530F7E4}}	LastUpdated:	9/20/2002 2:08:31 AM
NameMap:	Long binary data	OrderBy:	QUALIFICATION.Importance
OrderByOn:	True	Orientation:	Left-to-Right
RecordCount:	87	Updatable:	True

Columns

Name		Type	Size
QualificationID		Long Integer	4
AllowZeroLength:	False		
Attributes:	Fixed Size, Auto-Increment		
CollatingOrder:	General		
ColumnHidden:	False		
ColumnOrder:	Default		
ColumnWidth:	Default		
DataUpdatable:	False		
Description:	Primary Key		
GUID:	{guid {1ED5493E-74DE-4F34-86EE-371A51268388}}		
OrdinalPosition:	0		
Required:	True		
SourceField:	QualificationID		
SourceTable:	QUALIFICATION		
QualificationCode		Text	15
AllowZeroLength:	True		
Attributes:	Variable Length		
CollatingOrder:	General		
ColumnHidden:	False		
ColumnOrder:	Default		
ColumnWidth:	1695		
DataUpdatable:	False		
Description:	Code that gets imported from ODIS		
DisplayControl:	Text Box		
GUID:	{guid {867818B6-37FF-4D8A-AB68-1092E280A209}}		
IMEMode:	0		
IMESentenceMode:	3		
OrdinalPosition:	1		
Required:	False		
SourceField:	QualificationCode		
SourceTable:	QUALIFICATION		
UnicodeCompression:	False		
Abbreviation		Text	15
AllowZeroLength:	True		
Attributes:	Variable Length		

CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	Default	
DataUpdatable:	False	
Description:	Abbreviated translation for members' codes	
DisplayControl:	Text Box	
GUID:	{guid {96F197EA-2764-4621-97DC-4FDF84A30D1D}}	
IMEMode:	0	
IMESentenceMode:	3	
OrdinalPosition:	2	
Required:	False	
SourceField:	Abbreviation	
SourceTable:	QUALIFICATION	
UnicodeCompression:	False	

Description	Text	250
AllowZeroLength:	True	
Attributes:	Variable Length	
CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	5280	
DataUpdatable:	False	
Description:	Full description of qualification	
DisplayControl:	Text Box	
GUID:	{guid {418E200F-F92C-461E-828B-F3C9B95A5820}}	
IMEMode:	0	
IMESentenceMode:	3	
OrdinalPosition:	3	
Required:	False	
SourceField:	Description	
SourceTable:	QUALIFICATION	
UnicodeCompression:	False	

Importance	Long Integer	4
AllowZeroLength:	False	
Attributes:	Fixed Size	
CollatingOrder:	General	
ColumnHidden:	False	
ColumnOrder:	Default	
ColumnWidth:	Default	
DataUpdatable:	False	
DecimalPlaces:	Auto	
Description:	Sorting order	
DisplayControl:	Text Box	
GUID:	{guid {275EEAF1-F3C6-499A-A209-36281A6EC8EA}}	
OrdinalPosition:	4	
Required:	True	
SourceField:	Importance	
SourceTable:	QUALIFICATION	

Relationships

QUALIFICATIONMEMBER_QUALIFICATION

QUALIFICATION			**MEMBER_QUALIFICA**
QualificationID	1	∞	QualificationID

Attributes:	Enforced, Cascade Updates, Cascade Deletes
RelationshipType:	One-To-Many

Table Indexes

Name	Number of Fields
PKC_QUALIFICATION000C	1
Clustered:	False
DistinctCount:	87
Foreign:	False
IgnoreNulls:	False
Name:	PKC_QUALIFICATION000C
Primary:	True
Required:	True
Unique:	True
Fields:	
QualificationID	Ascending

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX H: VISUAL BASIC CODE

Form: frmACTIVITY

<u>**Code**</u>

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmACTIVITY"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cmdFirstRecord_Click()
On Error GoTo Err_cmdFirstRecord_Click


    DoCmd.GoToRecord , , acFirst

Exit_cmdFirstRecord_Click:
    Exit Sub

Err_cmdFirstRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdFirstRecord_Click

End Sub
Private Sub cmdPreviousRecord_Click()
On Error GoTo Err_cmdPreviousRecord_Click

    Dim intResponse As Integer
    DoCmd.GoToRecord , , acPrevious

Exit_cmdPreviousRecord_Click:
    Exit Sub

Err_cmdPreviousRecord_Click:

    intResponse = MsgBox("You are on the first record.", vbOKOnly)
    Resume Exit_cmdPreviousRecord_Click

End Sub
Private Sub cmdNextRecord_Click()
On Error GoTo Err_cmdNextRecord_Click

    Dim intResponse As Integer
    DoCmd.GoToRecord , , acNext

Exit_cmdNextRecord_Click:
    Exit Sub

Err_cmdNextRecord_Click:
    intResponse = MsgBox("There are no more records.", vbOKOnly)
    Resume Exit_cmdNextRecord_Click

End Sub
Private Sub cmdLastRecord_Click()
On Error GoTo Err_cmdLastRecord_Click


    DoCmd.GoToRecord , , acLast

Exit_cmdLastRecord_Click:
    Exit Sub

Err_cmdLastRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdLastRecord_Click

End Sub
Private Sub cmdFind_Click()
On Error GoTo Err_cmdFind_Click


    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70

Exit_cmdFind_Click:
    Exit Sub

Err_cmdFind_Click:
    MsgBox Err.Description
    Resume Exit_cmdFind_Click

End Sub
Private Sub cmdNewRecord_Click()
On Error GoTo Err_cmdNewRecord_Click


    DoCmd.GoToRecord , , acNewRec

Exit_cmdNewRecord_Click:
    Exit Sub

Err_cmdNewRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewRecord_Click

End Sub
Private Sub cmdDeleteRecord_Click()
On Error GoTo Err_cmdDeleteRecord_Click

    Dim intResponse As Integer

    intResponse = MsgBox("This will delete the current activity and cannot be undone." _
        & vbCr & "Are you sure you want to delete this activity?", vbYesNo _
        + vbCritical + vbDefaultButton2, "DELETE ACTIVITY")

    If intResponse = 6 Then
        DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
        DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
    End If


Exit_cmdDeleteRecord_Click:
    Exit Sub

Err_cmdDeleteRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdDeleteRecord_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click


    DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
    DoCmd.PrintOut acSelection

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub

Private Sub Form_Current()


    If Me.NewRecord Then
        Me!lblNavigate.Caption = "New Record"
    Else
        With Me.RecordsetClone
            .Bookmark = Me.Bookmark
            Me!lblNavigate.Caption = "Record " & _
                    .AbsolutePosition + 1 _
                    & " of " & .RecordCount
        End With

    End If
End Sub


```

Form: frmACTIVITYBilletsSubform

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmActivityBilletsSubform"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cmdDeleteRecord_Click()
On Error GoTo Err_cmdDeleteRecord_Click

    Dim intResponse As Integer

    intResponse = MsgBox("This will delete the current billet and cannot be undone." _
        & vbCr & "Are you sure you want to delete this billet?", vbYesNo _
        + vbCritical + vbDefaultButton2, "DELETE BILLET")

    If intResponse = 6 Then
        DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
        DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
    End If

Exit_cmdDeleteRecord_Click:
    Exit Sub

Err_cmdDeleteRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdDeleteRecord_Click

End Sub
```

Form: frmAssociateData

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmAssociateData"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub Detail_Click()
Private Sub cmdAppendMemberData_Click()
On Error GoTo Err_cmdAppendMemberData_Click

    Dim stDocName As String
    Dim intResponse As Integer

    intResponse = MsgBox("This will transfer the current member data and cannot be undone." _

        & vbCr & "Please ensure you have cleaned up the data and are ready to transfer." _
        & vbCr & "Do you want to proceed with the data transfer?", vbYesNo _
         + vbCritical + vbDefaultButton2, "TRANSFER DATA TO MEMBER TABLE")

    If intResponse = 6 Then
        stDocName = "qryAppendtmpMemberToMEMBER"
        DoCmd.OpenQuery stDocName, acNormal, acEdit
    End If

Exit_cmdAppendMemberData_Click:
    Exit Sub

Err_cmdAppendMemberData_Click:
    MsgBox Err.Description
    Resume Exit_cmdAppendMemberData_Click

End Sub
End Sub
Private Sub cmdRun_Click()
On Error GoTo Err_cmdRun_Click

    Dim stDocName As String

    stDocName = "mcrAssociateBillets"
    DoCmd.RunMacro stDocName

Exit_cmdRun_Click:
    Exit Sub

Err_cmdRun_Click:
    MsgBox Err.Description
    Resume Exit_cmdRun_Click

End Sub
```

Form: frmMEMBER

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmMEMBER"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cmdFirstRecord_Click()
On Error GoTo Err_cmdFirstRecord_Click


    DoCmd.GoToRecord , , acFirst

Exit_cmdFirstRecord_Click:
    Exit Sub

Err_cmdFirstRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdFirstRecord_Click

End Sub
Private Sub cmdPreviousRecord_Click()
On Error GoTo Err_cmdPreviousRecord_Click

    Dim intResponse As Integer
    DoCmd.GoToRecord , , acPrevious

Exit_cmdPreviousRecord_Click:
    Exit Sub

Err_cmdPreviousRecord_Click:

    intResponse = MsgBox("You are on the first record.", vbOKOnly)
    Resume Exit_cmdPreviousRecord_Click

End Sub
Private Sub cmdNextRecord_Click()
On Error GoTo Err_cmdNextRecord_Click

    Dim intResponse As Integer
    DoCmd.GoToRecord , , acNext

Exit_cmdNextRecord_Click:
    Exit Sub

Err_cmdNextRecord_Click:
    intResponse = MsgBox("There are no more records.", vbOKOnly)
    Resume Exit_cmdNextRecord_Click

End Sub
Private Sub cmdLastRecord_Click()
On Error GoTo Err_cmdLastRecord_Click


    DoCmd.GoToRecord , , acLast

Exit_cmdLastRecord_Click:
    Exit Sub

Err_cmdLastRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdLastRecord_Click

End Sub
Private Sub cmdFind_Click()
On Error GoTo Err_cmdFind_Click


    Screen.PreviousControl.SetFocus
    DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70

Exit_cmdFind_Click:
    Exit Sub

Err_cmdFind_Click:
    MsgBox Err.Description
    Resume Exit_cmdFind_Click

End Sub
Private Sub cmdNewRecord_Click()
On Error GoTo Err_cmdNewRecord_Click


    DoCmd.GoToRecord , , acNewRec

Exit_cmdNewRecord_Click:
    Exit Sub

Err_cmdNewRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewRecord_Click

End Sub
Private Sub cmdDeleteRecord_Click()
On Error GoTo Err_cmdDeleteRecord_Click

    Dim intResponse As Integer
    Dim frmMain As Form
    Set frmMain = Forms!frmMEMBER

```

```
   intResponse = MsgBox("This will delete " & frmMain.txtRank _
      & " " & frmMain.txtLastName & " from the database and cannot be undone." _
      & vbCr & "Are you sure you want to delete this record?", vbYesNo _
      + vbCritical + vbDefaultButton2, "DELETE SERVICE MEMBER")

   If intResponse = 6 Then
      DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
      DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
   End If


Exit_cmdDeleteRecord_Click:
   Exit Sub

Err_cmdDeleteRecord_Click:
   MsgBox Err.Description
   Resume Exit_cmdDeleteRecord_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click


   DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
   DoCmd.PrintOut acSelection

Exit_cmdPrint_Click:
   Exit Sub

Err_cmdPrint_Click:
   MsgBox Err.Description
   Resume Exit_cmdPrint_Click

End Sub




Private Sub Form_Current()

   Me.Refresh

   If Me.NewRecord Then
      Me!lblNavigate.Caption = "New Record"
   Else
      With Me.RecordsetClone
         .Bookmark = Me.Bookmark
         Me!lblNavigate.Caption = "Record " & _
               .AbsolutePosition + 1 _
               & " of " & .RecordCount
      End With

   End If
End Sub
```

```


Private Sub cmdAssignBillet_Click()
On Error GoTo Err_cmdAssignBillet_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmBilletAssignment"

    stLinkCriteria = "[MemberID]=" & Me![txtMemberID]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdAssignBillet_Click:
    Exit Sub

Err_cmdAssignBillet_Click:
    MsgBox Err.Description
    Resume Exit_cmdAssignBillet_Click

End Sub
Private Sub cmdMbrDetails_Click()
On Error GoTo Err_cmdMbrDetails_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmMemberBilletHistory"

    stLinkCriteria = "[SSN]=" & "'" & Me![txtSSN] & "'"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdMbrDetails_Click:
    Exit Sub

Err_cmdMbrDetails_Click:
    MsgBox Err.Description
    Resume Exit_cmdMbrDetails_Click

End Sub
```

Form: frmMemberBilletHistory

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmMemberBilletHistory"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub Form_Load()

    If fraPhone.Value = 1 Then

        txtPhoneHome.InputMask = "!(999) 000-0000;;_"
        txtPhoneCell.InputMask = "!(999) 000-0000;;_"
        txtPhoneWorkComm.InputMask = "!(999) 000-0000;;_"

    Else
        txtPhoneHome.InputMask = ""
        txtPhoneCell.InputMask = ""
        txtPhoneWorkComm.InputMask = ""

    End If

End Sub

Private Sub fraPhone_BeforeUpdate(Cancel As Integer)

    If fraPhone.Value = 1 Then

        txtPhoneHome.InputMask = "!(999) 000-0000;;_"
        txtPhoneCell.InputMask = "!(999) 000-0000;;_"
        txtPhoneWorkComm.InputMask = "!(999) 000-0000;;_"

    Else
        txtPhoneHome.InputMask = ""
        txtPhoneCell.InputMask = ""
        txtPhoneWorkComm.InputMask = ""

    End If

End Sub


Private Sub cmdNewBillet_Click()
On Error GoTo Err_cmdNewBillet_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmBILLET"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdNewBillet_Click:
    Exit Sub

Err_cmdNewBillet_Click:
    MsgBox Err.Description
    Resume Exit_cmdNewBillet_Click

End Sub
```

Form: frmMemberByRank_PRD

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmMemberByRank_PRD"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub chkOR1_Click()

    If chkOR1.Value = True Then
       cboRank2.Visible = True
        lblRank2.Visible = True
        lblOR2.Visible = True
        chkOR2.Visible = True
    Else
        cboRank2.Visible = False
        lblRank2.Visible = False
        lblOR2.Visible = False
        chkOR2.Visible = False
        chkOR2.Value = False
        cboRank2.Value = Null
        cboRank3.Visible = False
        cboRank3.Value = Null
        lblRank3.Visible = False

    End If


End Sub

Private Sub chkOR2_Click()

    If chkOR2.Value = True Then
       cboRank3.Visible = True
        lblRank3.Visible = True
    Else
        cboRank3.Visible = False
        lblRank3.Visible = False
        cboRank3.Value = Null
    End If

End Sub
Private Sub cmdOpenReport_Click()
On Error GoTo Err_cmdOpenReport_Click

    Dim stDocName As String

    stDocName = "rptMemberByRank_PRD"
    DoCmd.OpenReport stDocName, acPreview

Exit_cmdOpenReport_Click:
    Exit Sub

Err_cmdOpenReport_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenReport_Click

End Sub

```

```
Private Sub Form_Load()
   chkOR1.Value = False
   chkOR2.Value = False

End Sub
```

Form: frmMemberByRank_Qualification

<u>Code</u>

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmMemberByRank_Qualification"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub chkOR1_Click()

   If chkOR1.Value = True Then
      cboRank2.Visible = True
      lblRank2.Visible = True
      lblOR2.Visible = True
      chkOR2.Visible = True
   Else
      cboRank2.Visible = False
      lblRank2.Visible = False
      lblOR2.Visible = False
      chkOR2.Visible = False
      chkOR2.Value = False
      cboRank2.Value = Null
      cboRank3.Visible = False
      cboRank3.Value = Null
      lblRank3.Visible = False

   End If


End Sub

Private Sub chkOR2_Click()

   If chkOR2.Value = True Then
      cboRank3.Visible = True
      lblRank3.Visible = True
   Else
      cboRank3.Visible = False
      lblRank3.Visible = False
      cboRank3.Value = Null
   End If

End Sub
Private Sub cmdOpenReport_Click()
On Error GoTo Err_cmdOpenReport_Click

   Dim stDocName As String

   stDocName = "rptMemberByRank_Qualification"
   DoCmd.OpenReport stDocName, acPreview

Exit_cmdOpenReport_Click:
   Exit Sub

Err_cmdOpenReport_Click:
   MsgBox Err.Description
   Resume Exit_cmdOpenReport_Click

End Sub

Private Sub Form_Load()
   chkOR1.Value = False
   chkOR2.Value = False

End Sub
```

Form: frmMemberCleanup

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmMemberCleanup"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub cmdAddBillets_Click()
On Error GoTo Err_cmdAddBillets_Click

    Dim stDocName As String
    Dim intResponse As Integer
    Dim intResponse2 As Integer

    intResponse = MsgBox("This will add one billet for each instance of a unique UIC,BSC liste here." _
        & vbCr & "This action cannot be undone. Please ensure you have cleaned up the data and are" _
        & " ready to add these billets." & vbCr & "Do you want to proceed with the data transfer?", vbYesNo _
        + vbCritical + vbDefaultButton2, "ADD MISSING BILLETS")

    If intResponse = 6 Then
        stDocName = "qryAppendMissingBilletsFromtmpMembertotmpBILLET"
        DoCmd.OpenQuery stDocName, acNormal, acEdit
        DoCmd.Close

        intResponse2 = MsgBox("If you have finished cleaning up all the member information " _
        & "you can complete the process of importing the bodies by clicking the " _
        & "Yes button below.  If you want to review the members once more, click No.", vbYesNo _

        + vbCritical + vbDefaultButton1, "FINALIZE MEMBER INFORMATION" )

        If intResponse2 = 6 Then
            stDocName = "qryAppendtmpMemberToMEMBER"
            DoCmd.OpenQuery stDocName, acNormal, acEdit
        Else
            stDocName = "frmMemberCleanup"
            DoCmd.OpenForm stDocName, acNormal
        End If

    End If

Exit_cmdAddBillets_Click:
    Exit Sub

Err_cmdAddBillets_Click:
    MsgBox Err.Description
    Resume Exit_cmdAddBillets_Click
```

```

End Sub
```

Form: frmProjectionDate

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmProjectionDate"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database

Private Sub chkProjDate_Click()


    If chkProjDate.Value = True Then
        txtProjectionDate.Visible = False
        lblProjectionDate.Visible = False
        lblDateFormat.Visible = False
        cmdSubmitProjectionDate.Visible = False
        cmdStaffingPlan.Visible = True
        cmdP1.Visible = True
        cmdP1Billets.Visible = True
        cmdP1Commands.Visible = True

    Else
        txtProjectionDate.Visible = True
        lblProjectionDate.Visible = True
        cmdSubmitProjectionDate.Visible = True
        lblDateFormat.Visible = True
        cmdStaffingPlan.Visible = False
        cmdP1.Visible = False
        cmdP1Billets.Visible = False
        cmdP1Commands.Visible = False

    End If

End Sub

Private Sub cmdP1Billets_Click()
On Error GoTo Err_cmdP1Billets_Click

    Dim stDocName As String

    stDocName = "rptP1"
    DoCmd.OpenReport stDocName, acPreview

Exit_cmdP1Billets_Click:
    Exit Sub

Err_cmdP1Billets_Click:
```

Form: frmProjectionDate

```
    MsgBox Err.Description
    Resume Exit_cmdP1Billets_Click
End Sub

Private Sub cmdP1Commands_Click()
On Error GoTo Err_cmdP1Commands_Click

    Dim stDocName As String

    stDocName = "rptP1CommandListing"
    DoCmd.OpenReport stDocName, acPreview

Exit_cmdP1Commands_Click:
    Exit Sub

Err_cmdP1Commands_Click:
    MsgBox Err.Description
    Resume Exit_cmdP1Commands_Click
End Sub

Private Sub cmdSubmitProjectionDate_Click()
On Error GoTo Err_cmdSubmitProjectionDate_Click

    Dim stDocName As String
    Dim stDocName2 As String

    stDocName = "qryProjectionDateUltToCurrent"
    stDocName2 = "qryUpdateDuplicateCurrentToPrevious"

    DoCmd.OpenQuery stDocName, acNormal, acEdit
    DoCmd.OpenQuery stDocName2, acNormal, acEdit
    MsgBox "Records successfully updated.", vbOKOnly, "UPDATE SUCCESSFUL"
    cmdStaffingPlan.Visible = True
    cmdP1.Visible = True
    cmdP1Billets.Visible = True
    cmdP1Commands.Visible = True

Exit_cmdSubmitProjectionDate_Click:
    Exit Sub

Err_cmdSubmitProjectionDate_Click:
    MsgBox Err.Description
    Resume Exit_cmdSubmitProjectionDate_Click

End Sub
Private Sub cmdStaffingPlan_Click()
On Error GoTo Err_cmdStaffingPlan_Click

    Dim stDocName As String

    stDocName = "rptStaffingPlan"
    DoCmd.OpenReport stDocName, acPreview

Exit_cmdStaffingPlan_Click:
```

Form: frmProjectionDate

```
    Exit Sub

Err_cmdStaffingPlan_Click:
    MsgBox Err.Description
    Resume Exit_cmdStaffingPlan_Click

End Sub
Private Sub cmdP1_Click()
On Error GoTo Err_cmdP1_Click

    Dim stDocName As String

    stDocName = "rptP1CECPersonnel"
    DoCmd.OpenReport stDocName, acPreview

Exit_cmdP1_Click:
    Exit Sub

Err_cmdP1_Click:
    MsgBox Err.Description
    Resume Exit_cmdP1_Click

End Sub
```

Form: frmReviewBillets

Code

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "Form_frmReviewBillets"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub cmdRefresh_Click()
On Error GoTo Err_cmdRefresh_Click


    Me.Requery

Exit_cmdRefresh_Click:
    Exit Sub

Err_cmdRefresh_Click:
    MsgBox Err.Description
    Resume Exit_cmdRefresh_Click

End Sub
```

Module: mdlConvertDate

Code

```
Attribute VB_Name = "mdlConvertDate"
Option Compare Database

Function ConvertDate(ByVal strDate) As Variant
'Takes the string and puts it into proper date format
On Error GoTo Err_ConvertDate

    If Len(strDate) = 6 Then
        strDate = Right(strDate, 2) + "/01/" + Left(strDate, 4)
    ElseIf strDate = "" Then
        strDate = ""
    ElseIf strDate = "*" Then
        strDate = ""
    Else
        strDate = Mid(strDate, 5, 2) + "/" + Right(strDate, 2) + "/" + Left(strDate, 4)
    End If

ConvertDate = strDate

Exit_ConvertDate:
    Exit Function
Err_ConvertDate:
    MsgBox Err.Description
    Resume Exit_ConvertDate


End Function
```

Module: mdlLastDay

Code

```
Attribute VB_Name = "mdlLastDay"
Option Compare Database

Function LastDay(ByVal m) As String
'Returns the last day of the month for a given month. Leap Years are not included
Dim strDay As String

    If m = "01" Or m = "03" Or m = "05" Or m = "07" Or m = "08" Or m = "10" Or m = "12" Then
        strDay = "31"
    ElseIf m = "04" Or m = "06" Or m = "09" Or m = "11" Then
        strDay = "30"
    Else
        strDay = "28"
    End If

    LastDay = strDay

End Function
```

Module: mdlNameParsing

Code

```
Attribute VB_Name = "mdlNameParsing"
Option Compare Database


Function CountCSWords(ByVal s) As Integer
'Counts the words in a string that are separated by spaces.
Dim WC As Integer, Pos As Integer
   If VarType(s) <> 8 Or Len(s) = 0 Then
      CountCSWords = 0
      Exit Function
   End If
   WC = 1
   Pos = InStr(s, " ")
   Do While Pos > 0
      WC = WC + 1
      Pos = InStr(Pos + 1, s, " ")
   Loop
   CountCSWords = WC
End Function

Function GetCSWord(ByVal s, Indx As Integer)
'Returns the nth word in a specific field.
Dim WC As Integer, Count As Integer
Dim SPos As Integer, EPos As Integer

   WC = CountCSWords(s)
   If Indx < 1 Or Indx > WC Then
      GetCSWord = Null
      Exit Function
   End If
   Count = 1
   SPos = 1
   For Count = 2 To Indx
      SPos = InStr(SPos, s, " ") + 1
   Next Count
   EPos = InStr(SPos, s, " ") - 1
   If EPos <= 0 Then EPos = Len(s)
   GetCSWord = Trim(Mid(s, SPos, EPos - SPos + 1))
End Function
```

Module: mdlTitleCase

Code

```
Attribute VB_Name = "mdlTitleCase"
Option Compare Database

Function TCase(ByVal strWord) As String
'Returns the string sent in whatever format to Title Case

   Dim intLength As Integer
   Dim strFirst As String, strRest As String
   Dim WC As Integer, Pos As Integer
   Dim WordCount As Integer

   If VarType(strWord) <> 8 Or Len(strWord) = 0 Then
      WordCount = 0
      Exit Function
   End If
   WC = 1
   Pos = InStr(strWord, " ")
   Do While Pos > 0
      WC = WC + 1
      Pos = InStr(Pos + 1, strWord, " ")
   Loop
   WordCount = WC

   intLength = Len(strWord)
   strFirst = UCase(Left(strWord, 1))
   strRest = LCase(Right(strWord, intLength - 1))

   TCase = strFirst + strRest


End Function
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I: PROJECT CD-ROM

THE PROJECT CD-ROM CAN BE FOUND IN THE LIBRARY COPY OF THIS THESIS. NO DATA IS STORED IN THE DATABASE ON THIS COPY.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Kroenke, David M., *Database Processing-Fundamentals, Design & Implementation*, pp. 11-15, 30, 41-42, 113-126, Seventh Edition, Prentice Hall, 2000.

2. Whitten, Jeffrey, and Bentley, Lonnie. *Systems Analysis and Design Methods*, pp. 72-73, 315, 337, 407-408, 568, Fourth Edition, Irwin McGraw-Hill, 1998.

3. Osmundson, John. “Government and Other Standards,” Notes presented in Software Management Class – Naval Postgraduate School, Monterey, CA, Winter Quarter 2002.

4. Cook, Glenn. “Tangible and Intangible Factors,” Notes presented in Economic Evaluation of Information Technology Class – Naval Postgraduate School, Monterey, CA, Summer Quarter 2001.

5. Blackburn, Ian, Dewson, Robin, Hanselman, Scott, et. al., *Professional Access 2000 Programming*, pg. 23, Wrox Press Ltd., 2000.

6. Dr Xinping SHI, http://www.hkbu.edu.hk/~xpshi/ism2600/lecture8.ppt

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Commander
   Naval Facilities Engineering Command
   Chief Information Office
   Washington Navy Yard, District of Columbia

4. Commander
   Navy Personnel Command
   PERS-4413
   Millington, Tennessee

5. CAPT Michael P. Doyle, CEC, USN
   Deputy for Management
   US Naval Academy
   Annapolis, Maryland

6. Dr. Magdi N. Kamel
   Code IS
   Naval Postgraduate School
   Monterey, California

7. Professor Dale M. Courtney
   Code IS
   Naval Postgraduate School
   Monterey, California

8. D.C. Boger
   Code IS
   Naval Postgraduate School
   Monterey, California