

NAVAL POSTGRADUATE SCHOOL
Monterey, California



DTIC QUALITY INSPECTED 2

THESIS

**REENGINEERING THE UNITED STATES MARINE
CORPS' RECRUIT DISTRIBUTION MODEL (RDM)**

by

Kevin J. Snoap

September 1998

Thesis Advisor:
Associate Advisor:

Hemant K. Bhargava
Suresh Sridhar

19981009 117

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 1998	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE : REENGINEERING THE UNITED STATES MARINE CORPS' RECRUIT DISTRIBUTION MODEL (RDM)			5. FUNDING NUMBERS	
6. AUTHOR(S) Kevin J. Snoop				
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)			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the authors 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 United States Marine Corps accomplishes its mission "to put the right Marine in the right place at the right time with the right skills and quality of life" in a variety of ways. One of the information systems assisting the Marine Enlisted Assignments branch is the Recruit Distribution Model (RDM). This thesis proposes changes to the RDM user interface, data management, assignment model, and analysis capability. With the use of business process reengineering, process modeling, mathematical modeling, and database design a fully functional prototype has been developed to address each identified change proposal. This reengineered system includes numerous innovations such as an intuitive navigational scheme using switchboards, and the elimination of manual data entry for data already available in the system. It also provides a number of significant contributions beneficial to the USMC. For instance, the reengineered system allows the user to objectively analyze different results by comparing four different objective measures, and its mathematical model uses commercial-off-the-shelf products eliminating a proprietary solver. All these changes will empower managers to effectively and efficiently manage the assignment of recruits in order to meet the challenges of the 21st century.				
14. SUBJECT TERMS USMC, Databases, Manpower Assignment, Models, Decision Support Systems, Graphical User Interface			15. NUMBER OF PAGES 100	
			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. Z39-18

Approved for public release; distribution is unlimited.

**REENGINEERING THE UNITED STATES MARINE
CORPS' RECRUIT DISTRIBUTION MODEL (RDM)**

Kevin J. Snoap
Lieutenant, United States Navy
B.S., University of Texas at Austin, 1991

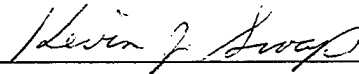
Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL
September 1998**

Author:

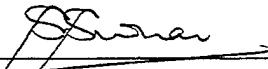


Kevin J. Snoap

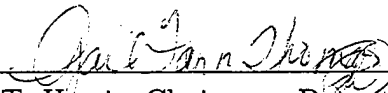
Approved by:



Hemant K. Bhargava, Advisor



Suresh Sridhar, Associate Advisor



Rueben T. Harris, Chairman, Department of Systems
Management

ABSTRACT

The United States Marine Corps accomplishes its mission “to put the right Marine in the right place at the right time with the right skills and quality of life” in a variety of ways. One of the information systems assisting the Marine Enlisted Assignments branch is the Recruit Distribution Model (RDM). This thesis proposes changes to the RDM user interface, data management, assignment model, and analysis capability. With the use of business process reengineering, process modeling, mathematical modeling, and database design a fully functional prototype has been developed to address each identified change proposal. This reengineered system includes numerous innovations such as an intuitive navigational scheme using switchboards, and the elimination of manual data entry for data already available in the system. It also provides a number of significant contributions beneficial to the USMC. For instance, the reengineered system allows the user to objectively analyze different results by comparing four different objective measures, and its mathematical model uses commercial-off-the-shelf products eliminating a proprietary solver. All these changes will empower managers to effectively and efficiently manage the assignment of recruits in order to meet the challenges of the 21st century.

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	THESIS PURPOSE	1
B.	GENERAL PROBLEM DESCRIPTION	1
C.	SIGNIFICANT CONTRIBUTIONS	5
II.	REENGINEERING MOTIVATIONS AND PROCESS VIEW	7
A.	LIMITATIONS OF OLD SYSTEM	7
1.	Navigation	7
2.	Data Management	8
3.	Assignment Procedure	8
4.	Transaction Processing Approach	9
B.	A PROCESS VIEW	9
1.	Recruit Distribution Model Operating Environment	10
2.	New System Level 0 Diagram	11
III.	MODELING THE RECRUIT DISTRIBUTION PROBLEM .	15
A.	OPTIMIZATION OBJECTIVES: FIT AND FILL	15
B.	MEASURING FITNESS	16
C.	MEASURING FILL	18
D.	ASSIGNMENT MODEL	19
1.	Assumptions	19
2.	Notation	20
3.	Objective	20
4.	Constraints	20
IV.	A DECISION SUPPORT SYSTEM FOR RECRUIT DISTRI- BUTION	23
A.	ARCHITECTURE AND IMPLEMENTATION	23
1.	Architecture	23

2.	Implementation	26
B.	USING THE RECRUIT DISTRIBUTION DECISION SUPPORT SYSTEM	28
1.	Setting up a Run	28
2.	Model Execution	31
3.	Customizing a Run and Results	33
C.	INNOVATIONS IN USER INTERACTION	34
D.	INNOVATIONS IN ANALYZING MODEL RESULTS	35
E.	OBJECTIVE COMPARISON OF OLD AND NEW SYSTEM SOLUTIONS	39
F.	SUMMARY	41
V.	CONCLUSION	43
A.	SIGNIFICANT CONTRIBUTIONS	43
B.	LESSONS LEARNED	43
C.	PROTOTYPE IMPROVEMENTS	44
1.	Speed Improvements	44
2.	Multiple Solution Storage	45
3.	Administration Switchboard	46
	APPENDIX A. ACRONYMS	47
	APPENDIX B. AS-IS RD BUSINESS PROCESS (IDEF0) MODEL	49
	APPENDIX C. TO-BE RD BUSINESS PROCESS (IDEF0) MODEL	61
	APPENDIX D. RDDSS VBA CODE	67
	LIST OF REFERENCES	87
	INITIAL DISTRIBUTION LIST	89

LIST OF FIGURES

1.	Assigning Marines to Schools	2
2.	Assigning Marines to School Classes	4
3.	Current Recruit Distribution Operating Environment	10
4.	RDdss Process, Level 0 Diagram	12
5.	Exponential Function for Calculating $Score_{level(p)}$	17
6.	Two Candidate Penalty Functions	19
7.	RDdss Architecture	23
8.	RDdss Main Switchboard	24
9.	RDdss Relational Schema	25
10.	RDdss Demonstration Solver	26
11.	RDdss Decomposition Diagram	27
12.	Actual RDdss Build Sequence	28
13.	Setting up a Model Run	29
14.	Model Execution	32
15.	Changing a Graduation Date	35
16.	Two alternative results: Fit	37
17.	Two alternative results: Fill	37

I. INTRODUCTION

A. THESIS PURPOSE

The main purpose of this thesis was to reengineer a United States Marine Corps' Manpower Assignment model concerned with the distribution of recruits to schools. This model is called the Recruit Distribution Model (RDM) [Ref. 1]. Throughout this thesis, the RDM is addressed as either RDM or the old system.

Additionally, an important purpose of this thesis was to build a functional prototype of the reengineered RDM. The new system is called the Recruit Distribution Decision Support System (RDdss). It demonstrates the functionality of the reengineered RDM. Throughout this thesis, the RDdss is addressed as either RDdss or the new system.

The majority of this thesis is devoted to a discussion of the RDdss. As necessary, the RDM is discussed. The following section is a general discussion designed to set the stage for understanding the rest of the thesis. It provides a problem description of recruit distribution in the United States Marine Corps (USMC). The last section of this chapter discusses the significant contributions of this thesis.

B. GENERAL PROBLEM DESCRIPTION

Recruit distribution in the USMC is the process that assigns recruits to an entry level school (ELS) leading to a military occupational specialty (See Figure 1). These assignments are made about 48 times a year, during the last week of Marine Corps Recruit Depot (MCRD) training. In this ending period of the MCRD, the recruits are facing the "Crucible," which is the final wicket a recruit endures before officially becoming a Marine. Consequently, the use of the titles recruit and Marine are used interchangeably in this paper.

For at least two reasons, this assignment process is a critical manpower function. First, a Marine's military occupational specialty (MOS) ultimately determines

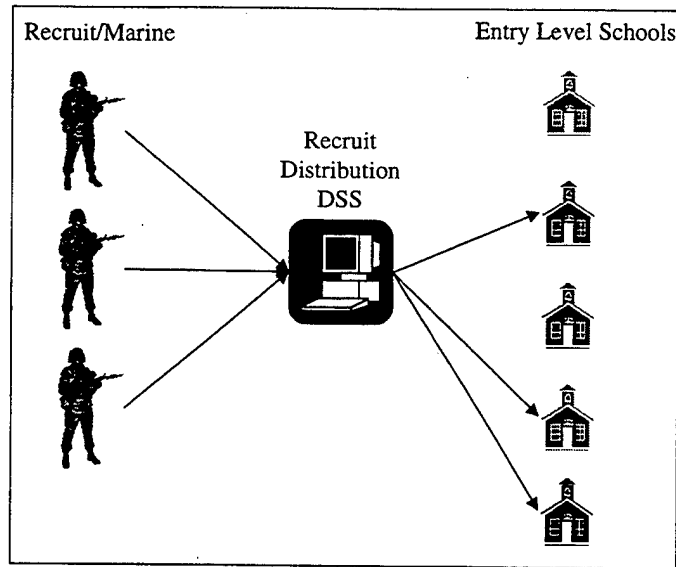


Figure 1. Assigning Marines to Schools

the member's career. Therefore, it is in the best interest of both the Marine and the USMC that a school assignment matching the Marine's desire is made. Second, the success obtained by a service member during his or her time in the USMC is partially based upon successful completion of their ELS, where a pattern of success is established. Therefore, a school assignment maximizing the chances of the Marine completing their training is important. So, fulfilling the Marine's desire and matching him or her to an MOS (i.e. ELS) that fits their personal characteristics is critical to the overall health of the USMC, making the assignment process a critical manpower function [Ref. 2].

Fulfilling the desire of the Marine is accomplished through the use of a contract guarantee. This is called a program enlisted for (PEF), and is specified during the recruiting process. For instance, a PEF = 19 is the "Tank and Assault Amphibian Option." There are currently two ELS's associated with this, "M1A1 Tank Crewman" and "Assault Amphibian Crewman." So, a Marine who chose the PEF = 19 is a possible candidate for these two schools and no others.

Once the Marine's school options are known, a Marine-to-school fit is deter-

mined for each of these schools (See Chapter III, Section B for details). This fitness determination is partly made by looking at each ELS's minimum eligibility requirements, which are called mandatory properties. This term "property" is used as defined by Webster's Ninth New Collegiate Dictionary: "a quality or trait belonging and especially peculiar to an individual or thing." Examples are Age > 18 , Clerical ≥ 80 , and Electrical ≥ 95 . The meaning of the first example is obvious. The other two are based on test scores from the Armed Services Vocational Aptitude Battery (ASVAB) test.

In addition to the mandatory properties, most schools also specify desirable properties. A desired property is the same as a mandatory property, except they are not prerequisites for attending the school. For example, the Traffic Management Coordination school desires Marines with a Clerical score of at least 100. So, a desired property of Clerical ≥ 100 is specified for this ELS.

By using the information obtained from the PEF, mandatory and desired properties, a fitness matrix is generated. This shows the fit of every Marine to every school he or she is eligible for. Since there are about 100 schools and on average 700 Marines considered for every run, this matrix has the potential of 70,000 matches.

However, the matrix size is actually bigger. This is because each of the schools is broken down by classes. Some ELS's have a class starting each week, others every month, and others every quarter. Over a given year, these classes total about 1,800. Following the practice of the USMC, only the classes over the next 3-4 months are considered during the assignment process (see Figure 2). Therefore, the fitness matrix is increased to a potential size of 350,000.

The fact that the school classes start at different times throughout the year and that the assignment process is only conducted 48 times a year (normally this occurs on Friday) causes the "3-month look ahead" in the model and has implications important to the USMC. The concern is school seats which never get filled. Each seat is prepaid, guaranteeing its availability to the USMC. This means every vacant

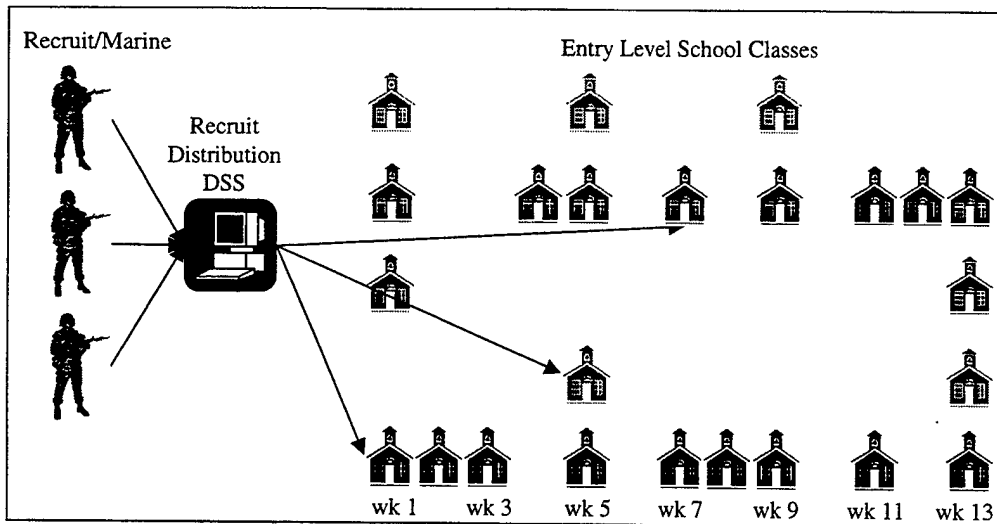


Figure 2. Assigning Marines to School Classes

spot is potentially a lost resource to the USMC. Therefore, in addition to the Marine-to-school fitness concern, there is also the concern of filling all available school seats before their report date is passed.

Another concern is the problem of unassigned Marines. There are numerous reasons why they may not get assigned. Maybe the only school class the Marine was eligible for is already full. Or, possibly he or she is not qualified for any of the school's promised by their PEF. A third possibility is errors in the data provided to the assignment process. Regardless of the cause, it requires identification and corrective action.

Finally, a discussion of the internal and external stakeholders associated with the USMC recruit distribution is given. Internally, there are the USMC and the recruits. Since their concerns were discussed above, they are not given any further consideration. Externally, there is the contractor Decision Support Associates, Inc. (DSAI) who has maintained and upgraded the RDM for over 30 years. They also maintain about eight other major systems for the USMC. For their services, the USMC pays a significant amount of money each year. Additionally, DSAI has proprietary software in some of these systems, which has locked-in the USMC to this

company. Besides the contractor, there is one last significant stakeholder, the American taxpayer. In a time of increasing fiscal constraints and shrinking budgets, it is imperative that wise decisions are made in regards to assigning Marines to ELS's. The American taxpayers expect nothing less.

C. SIGNIFICANT CONTRIBUTIONS

Four significant contributions beneficial to the USMC have been made in this thesis. A detailed discussion of these is given throughout the next three chapters. Here, we list and summarize the contributions.

- Analysis and articulation of the recruit distribution process - by interviewing USMC and DSAI personnel, reviewing available documentation, and operating the RDM we were able to articulate an understanding of the recruit distribution process using IDEF process modeling. Additional data modeling was articulated in a third normal form relational schema [Ref. 3].
- Development of a mathematical model - by analyzing the assignment process, criteria, and constraints, USMC policies and objectives, a mathematical model for the new system was developed.
- Fully functional prototype - an intuitive and easy to understand new system that seamlessly interfaces with the old system was built. It provides an objective means of comparing assignment results of both systems, and was developed using commercial-off-the-shelf (COTS) software applications.
- Elimination of the proprietary solver and associated contractor lock-in - this was accomplished by replacing the proprietary solver with two COTS applications.

II. REENGINEERING MOTIVATIONS AND PROCESS VIEW

The RDM has a number of limitations which motivated the USMC to reengineer it. We start this chapter by listing and describing each identified limitation. Then, to develop a better understanding of recruit distribution in the USMC, two different process views are examined. The first is concerned with the RDM operating environment, and the second is concerned with the first level of the RDdss IDEF0 model.

A. LIMITATIONS OF OLD SYSTEM

Throughout this reengineering effort, a number of limitations of the old system have become apparent. Many of these limits were recognized earlier, and were part of the USMC's motivation to reengineer the RDM. The following is a list of the identified limitations of the old system.

- Navigation
- Data management
- Assignment procedure
- Transaction processing approach

A description of each limitation is now given.

1. Navigation

Navigating through the RDM is not intuitive. It is neither obvious where one should start or where one should go. Navigation is accomplished by initially selecting an option from one of the main display's drop down menus. This normally results in a window appearing on the computer's desktop. Then, by pointing and clicking on the displayed window's buttons, further navigation is accomplished. After working with the RDM for a number of hours, we were able to navigate through the application

to find and display specific windows. However, after a week or two of not using the system, we had difficulty finding our way around again. In the RDdss, we have created an intuitive navigational scheme using switchboards.

2. Data Management

Data management in the RDM is poor, leading to the introduction of numerous errors. The biggest problem in this regard deals with data entry. The RDM violates the basic rule of never requiring the user to enter data already in the system [Ref. 4]. For example, creating a new school in the RDM requires the user to type data into seven different data fields. Only the course identification number has been automated by a drop down list. In the RDdss, when creating a new school we have reduced the manual data entry to one field.

3. Assignment Procedure

There are at least five limitations associated with the old system's assignment procedure. It

- is encoded into proprietary software,
- examines schools sequentially, rather than globally,
- makes assignments based on school priority rather than weights,
- attempts to maximize fill rather than fit-and-fill, and
- is relatively inflexible.

The solver performing these assignments was designed back in the 1950's, where the major concern was speed and using the minimal amount of memory. It is written in Fortran, and is proprietary code owned by DSAI. It does not search repetitively for an optimal solution by trying to maximize or minimize an objective function. Instead, it maximizes the fill of prioritized school seats. In the RDdss, we use a well know algorithm called CPLEX [Ref. 5]. After conducting around 600 iterations of comparing 344 Marines to 576 school classes, it produces an *optimal* solution. Its

objective function was written to make the assignment procedure flexible. It allows the RDdss manager to "game" the system by making fit-and-fill trade-offs, until a "good" solution is found.

4. Transaction Processing Approach

Finally, the old system follows a transaction processing approach, vice a decision support approach [Ref. 6]. This is partially due to the inflexibility of the solver. However, another contributor is the extreme difficulty in comparing one run to another. Other than providing a means for manually computing the numerical difference in Marines assigned, the RDM provides little insight or information views for comparing runs. This is because there is no way to objectively compare one run to another. As it is now, if the RDM manager launches the assignment model and everybody is assigned, the result becomes the approved assignments. In the RDdss, we have created an entire process devoted to providing insightful analysis of a given run (See Chapter IV, Section D for details). Its purpose is to support the making of wise assignment decisions.

B. A PROCESS VIEW

To develop a better understanding of recruit distribution in the USMC, two different process views are now examined. The first is concerned with the current operating environment of the old system. The goal is to develop a big picture view of this process. For those interested in further study of the old system, Appendix A contains the RDM business process IDEF0 model.

The second process view examined concerns the first level of the RDdss business process IDEF0 model. The goal is to develop a better understanding of the new system, without going into great detail. This lays a good foundation for the RDdss discussions in the remainder of this thesis. Appendix B contains the entire IDEF0 model of the new system.

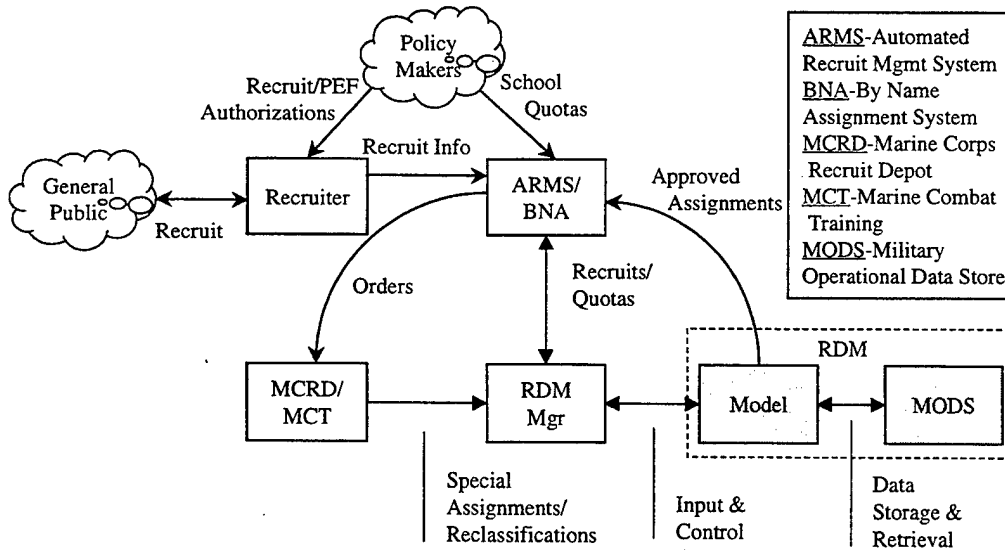


Figure 3. Current Recruit Distribution Operating Environment

1. Recruit Distribution Model Operating Environment

Figure 3 graphically depicts the operating environment of the USMC recruit distribution. At the top are the policy makers, who forecast and decide how many recruits and corresponding school seats are needed over the next few years. In addition to this, they determine how many PEF's or school guarantees to make available for a given year. Currently, about 65-70% of the recruits enter the USMC with a guarantee. The recruit and PEF authorizations for the following year are given to the USMC recruiters, and the school seats or quotas determined for the following years is input into the By Name Assignment (BNA) system.

The recruiters use the recruit and PEF authorizations in recruiting from the general public. Once a potential candidate is found they take the ASVAB, if they have not already done so. Based on the results, the recruiter can offer different guarantees. Once the candidate signs a contract with the USMC, their test scores, PEF, and other personal information such as age and height are entered into the Automated Recruit Management System (ARMS).

Both the ARMS and BNA systems utilize large main frame databases. They

serve as central repositories for maintaining data on many aspects dealing with the USMC. Keeping this information stored in one location ensures everyone is using the same data, providing consistency for all users. One user is the RDM manager, who retrieves recruit and school information from these systems for use in the RDM.

Additionally, the RDM receives data from two other sources. The MCRD instructors provide special assignment inputs. These are personnel identified as having the talents or abilities well suited for a particular school. The other data comes from the MCT. They provide the RDM manager with reclassification information. For instance, a Marine is reclassified if he or she is injured during MCT and is unable to make the start date of their assigned ELS.

All this recruit and school data is input into the RDM, where it is stored in the Military Operational Data Store (MODS). The model is then run. Once a satisfactory set of assignments is obtained, the RDM manager uploads the approved assignments to the ARMS and BNA systems. From this assignment information, the MCRD generates orders for the Marines graduating from the MCRD.

2. New System Level 0 Diagram

Focus is now shifted to the examination of the first process level of the RDdss. Looking any deeper will provide more detail than is necessary at this point. As mentioned earlier, the intent is to lay the foundation for the RDdss discussions in the remainder of this thesis.

Figure 4 shows the level 0, or context diagram of the RDdss. The program used to generate this illustration was BPWin, which is based on IDEF0 modeling. Consequently, all arrows entering from the left are considered "inputs," arrows entering from the top are "controls," arrows exiting to the right are "outputs," and arrows entering from the bottom are "mechanisms." A convenient acronym for this is ICOM [Ref. 7].

- I = Input: something consumed in the process
- C = Control: a constraint on the operation of the process

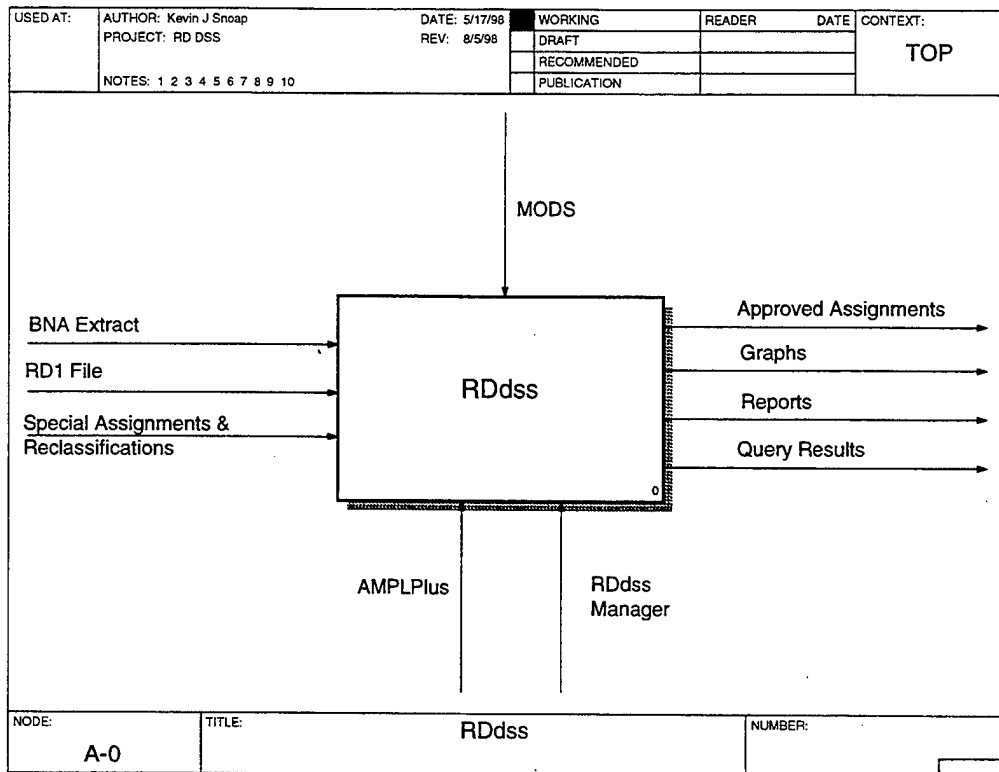


Figure 4. RDdss Process, Level 0 Diagram

- O = Output: something resulting from the process
- M = Mechanism: something used to perform the process, but is not itself consumed

There are three inputs to the RDdss, the BNA extract, the RD1 file, and the special assignments/reclassifications. The first of these contains a set of school class quotas covering a 120 day period. This comes to the RDdss manager as a fixed-width delimited text file. The RD1 file contains the data for the recruits soon to graduate from the MCRD. It also is a fixed-width delimited text file. The special assignments and reclassifications are the recruits identified with special abilities or talents and the Marines requiring reassignment to a different school class, respectively. This last data comes to the RDdss manager as a memo and not a text file.

The only control is the database (or Military Operational Data Store) used to store the data for the RDdss. Among other things, the MODS contains the informa-

tion specifying the eligibility requirements for each of the schools. This is why it is considered a control. It contains the data controlling or constraining the assignment of Marines to schools.

Besides the approved assignments that were mentioned earlier, the output consists of graphs, reports, and query results. Each graph was designed to provide insight into the current assignment result, plus provide a means for comparing different runs. The reports provide a print-out of information the RDdss manager might find useful. For instance, an approved assignment report and an unassigned Marine report are both available. The query results provide different views of the data in the MODS. For instance, one of the queries provides a listing of the fitness scores of all schools a Marine is eligible for.

Finally, the two mechanisms of the RDdss are the RDdss manager and the commercial-off-the-shelf (COTS) application AMPLPlus. Both of these entities are necessary to make the system work, and are not consumed by the process itself. This is why they are considered mechanisms. The RDdss manager is the one who points and clicks on the buttons of the RDdss, making it operate. AMPLPlus is the application containing the mathematical model used in making optimal Marine-to-school assignments. It interfaces with the CPLEX algorithm that actually performs the optimization.

III. MODELING THE RECRUIT DISTRIBUTION PROBLEM

We model the recruit distribution problem as an assignment problem. Various submodels are developed to compute input parameters and perform other preprocessing steps for the assignment model, which optimally assigns recruits to a school class. Following is a discussion of important points concerning the recruit distribution problem.

1. Schools and Classes: there are about 130 schools per fiscal year, which are broken down by classes (about 1800 classes per fiscal year). Classes of the same school, commencing on different dates, are identical. However, a class commencing soon after a recruit is available for training has greater utility than another class starting at a later date. On average, there are 600 classes in a run.
2. Program Enlisted For (PEF)
 - (a) 65-70% of enlistees enter the USMC under a guarantee (or PEF). The remaining recruits enter under an open contract (PEF=00).
 - (b) A recruit can be assigned only to schools associated with his or her PEF. Some PEF's are associated with one school, while others are associated with many schools.

A. OPTIMIZATION OBJECTIVES: FIT AND FILL

Our approach in the assignment model is to optimize by looking at both fit and fill. The importance of both these objectives was discussed in Chapter II, Section

B. The idea is to allow the RDdss manager to make a tradeoff between fit and fill.

The high-level objective function is:

- Maximize

$$K_1 \cdot \text{Fitness} - K_2 \cdot \text{Penalty}$$

Where K_1 is the fitness coefficient and K_2 is the fill coefficient.

By assigning different values to these coefficients, the RDdss manager is able to make trade-offs between fit and fill. This capability makes the assignment model flexible. It provides the RDdss manager a means for “gaming” the system.

B. MEASURING FITNESS

The first half of the high-level objective function deals with maximizing the fitness of Marines to schools. Therefore, we cover this topic now. Following is a discussion of important points concerning the measuring of fitness.

1. Marines are only eligible for assignment to schools corresponding to their PEF guarantees.
2. Associated with each school are specific mandatory properties, or minimum eligibility requirements (e.g. AGE > 18).
3. Eligibility of a Marine to a school is pre-determined by comparing a Marine’s attributes (e.g. Age = 20) to the school’s mandatory properties. A Marine is eligible only if he or she meets all the minimum school requirements.
4. Many schools also have desirable properties associated with them. This is a property which is desired in a Marine (e.g. Height > 65 inches), but are not prerequisites for attending the school. Since some desirable properties are more desirable than others, a means of distinguishing the properties is necessary. We have followed the Marine Corps practice of using 6 levels of desirability. Level 1 properties are the most desired, followed by level 2, etc.
5. Let $\mathcal{P}(s)$ denote the set of desirable properties for school s , and let $level(p)$ denote the desirability level of property p . Let $possesses_{(m,p)} = 1$ if Marine m possesses property p (0 otherwise). Further, let $Score_{level(p)} (> 0)$ be the score assigned for possessing a $level(p)$ property, where $Score_{level(p)}$ is calculated using an exponential function inversely proportional to $level(p)$. The scores for levels 1 through 6 are shown in Figure 5.
6. The total fitness score ($fit_{m,s}$) of Marine m to school s is composed of two parts: the score ($ManScore_{m,s}$) for possessing mandatory properties, and the score ($DesScore_{m,s}$) for possessing some or all of the desirable properties.

Our procedure for computing fitness scores is designed in a manner that, for each school, the average fitness score—with the average computed over all qualified Marines, i.e. Marines who have met the mandatory properties—is constant (i.e. 100). The reason we did this was to ensure each school

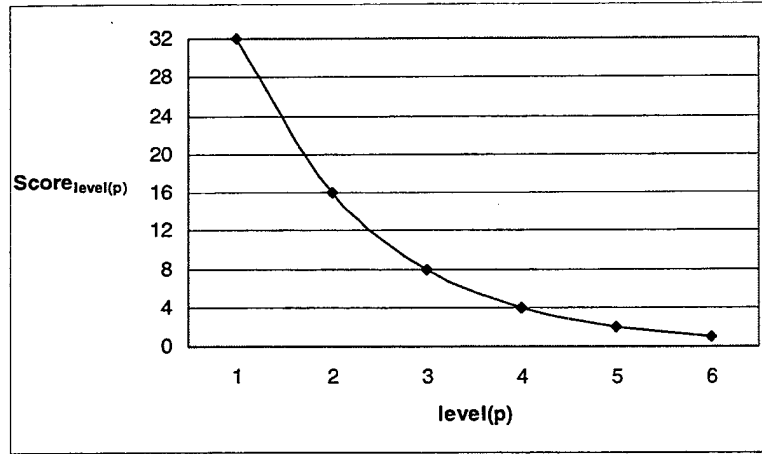


Figure 5. Exponential Function for Calculating $Score_{level(p)}$

received equal treatment in the model, regardless of the number of desirable properties specified for each school. Otherwise, all other things being equal, the school with the greatest number of desirable properties would receive the most assignments.

7. For each school s , percentage weights $ManWt_s$ and $DesWt_s$ are assigned, respectively, for the mandatory and desirable properties. For example, for school s_1 , a weight of 70% may be assigned to the mandatory properties and 30% for the desirable properties.
8. A Marine possessing all mandatory properties for school s is given an initial score $ManScore_{m,s} = ManWt_s \cdot 100$. Marines who do not possess all mandatory properties for a school are given an overall fitness score of 0 for that school.
9. Of all the Marines possessing the mandatory properties, those possessing desirable properties are awarded additional points, which are weighted by the level of the property. The Marine's score for desirable properties is computed as follows:
 - The absolute score $Abs_{m,s}$, for Marine m and school s , based on desirable properties is

$$Abs_{m,s} = \sum_{p \in \mathcal{P}(s)} Score_{level(p)} \cdot possesses(m, p)$$

The average of these absolute scores is computed over all qualified Marines. Let $AveAbs_s$ denote the average absolute score for school s .

- Each Marine's fitness score for desirable properties is then computed as a fraction of this average, and normalized by multiplying with the percentage weight for desirable properties. That is, $DesScore_{m,s}$ is a weighted relative score:

$$DesScore_{m,s} = \left(\frac{Abs_{m,s}}{AveAbs_s} \right) \cdot DesWt_s \cdot 100$$

The overall average of all these scores is, due to the above construction, $DesWt_s \cdot 100$.

10. The final fitness score, $fit_{m,s}$ is simply the sum $ManScore_{m,s} + DesScore_{m,s}$. It may be seen that, for each school, this number averages (over all the qualified Marines for that school) to 100.
11. As mentioned earlier, schools are broken down by classes that are identical, except for their start dates. Therefore, let $fit_{m,s} = fit_{m,c}$ for each class c of school s .

C. MEASURING FILL

The second half of the objective function deals with maximizing the fill of school seats. We now turn our attention to this topic.

As we discussed in Chapter II, school seats are paid for in advance. This is to guarantee the seat is available to the USMC. Consequently, every vacant spot is potentially a lost resource. Since the model is run about once every week, the biggest concern is the unfilled seats having report dates within the next seven days. Conversely, a seat having a report date in three months is not so critical.

To capture the essence of filling school seats with early report dates, we use a penalty function. The idea is simple. School seats having an early report date get a high penalty, and those having a late report date get a low penalty. A number of functions would have worked for this. We have graphed two candidates in Figure 6. The first is a linear function and the second a large-step function. Each of these was tried in the prototype. In our opinion the latter one is the best choice. It gives a lot of emphasis to the first week, plus it treats all schools having report dates within the same week equally. It is the function currently implemented in the rddss.

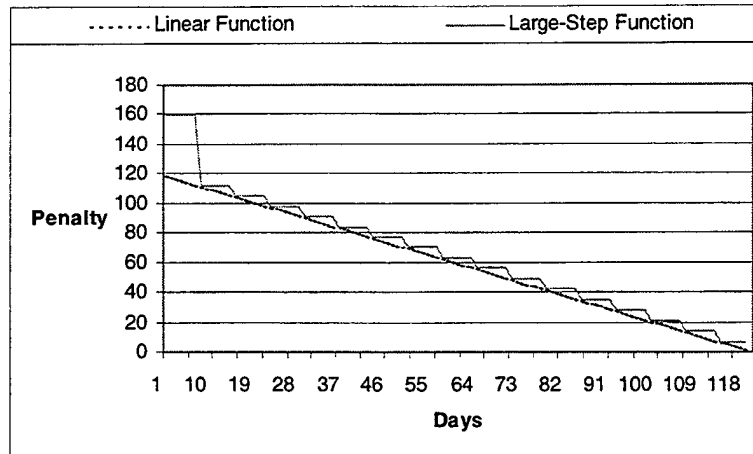


Figure 6. Two Candidate Penalty Functions

D. ASSIGNMENT MODEL

We are now ready to formally describe the assignment model. We will first specify the model's assumptions. Then, we will list the model's notation, followed by its objective function and constraints.

1. Assumptions

1. Each school is unique and identifies a single course of instruction leading to an MOS.
2. In any given run, a school may offer several classes that start on different dates. It is more desirable to fill seats in classes starting earlier. Such classes have a higher penalty per unit shortfall than similar classes starting at a later date.
3. All Marines available for assignment are graduating from the MCRD on the same date.
4. A Marine's eligibility for a school, as well as the fit for each school, is determined by the model preprocessor, taking into account the PEF guarantees, and the mandatory and desirable properties.
5. A Marine is assigned to a school corresponding to their PEF code, or not at all.
6. It is better to leave a Marine unassigned than to overfill a school.

7. The demand for the number of Marines to be trained in each school class is determined by looking at the BNA extract. This demand statement is already constrained by the capacity constraint at each school.

2. Notation

- Sets

\mathcal{M} : Marines

\mathcal{C} : Classes

- Exogenous Variables (Parameters in AMPL)

$fit_{m,c}$: the desirability of assigning Marine m to class c (note: the fitness score is zero for Marine-class pairs where either the Marine does not meet the classes mandatory properties, or where the class does not fulfill the Marine's PEF guarantee.)

$demand_c$: demand for Marines to be trained at class c

$penalty_c$: penalty for each unit of demand not met (higher value means it is more critical to fill the class)

- Decision Variables

$x_{m,c}$ (binary integer): 1, if Marine is assigned to the given class; 0 otherwise

3. Objective

Maximize the Total Utility: (Total Reward - Total Penalty)

$$TU = k_1 \left(\sum_{c \in \mathcal{C}} \sum_{m \in \mathcal{M}} fit_{m,c} \cdot x_{m,c} \right) - k_2 \left(\sum_{c \in \mathcal{C}} penalty_c \cdot (demand_c - \sum_{m \in \mathcal{M}} x_{m,c}) \right) \quad (\text{III.1})$$

4. Constraints

- assignmentLimit: a Marine is assignable to at most one school

$$\sum_{c \in \mathcal{C}} x_{m,c} \leq 1 \quad \forall m \quad (\text{III.2})$$

- eligibility: a Marine is only assignable to a class they are fit for (this prevents assigning Marines with a fitness score of zero)

$$x_{m,c} \leq fit_{m,c} \quad \forall m \quad \forall c \quad (\text{III.3})$$

- capacity: since penalties apply only if there is a positive shortfall, the objective function would be non-linear. To avoid that, we assume we will never oversupply Marines to schools, resulting in:

$$\sum_{m \in \mathcal{M}} x_{m,c} \leq demand_c \quad \forall c \quad (\text{III.4})$$

IV. A DECISION SUPPORT SYSTEM FOR RECRUIT DISTRIBUTION

A. ARCHITECTURE AND IMPLEMENTATION

Following is a discussion of the RDdss architecture and its implementation. First, the architectural components are discussed. Then, the steps taken to build the RDdss are described.

1. Architecture

The architecture of the RDdss is depicted in Figure 7. This illustration shows how the six major components of the system are related. We will describe this architecture by examining each component in the following order: switchboard, relational database, preprocessor, assignment model, solver, and analyzer.

The switchboard is one of two components providing an interface between the user and the RDdss. It is the mechanism by which the user controls the operation of the system. Figure 8 illustrates the main switchboard of the RDdss. It is displayed

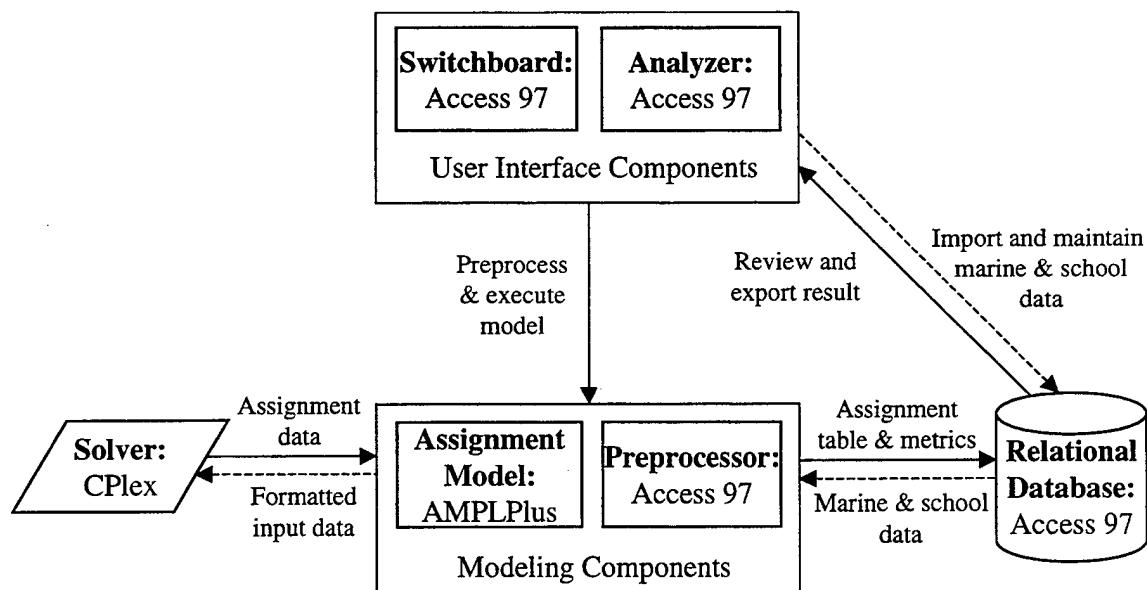


Figure 7. RDdss Architecture

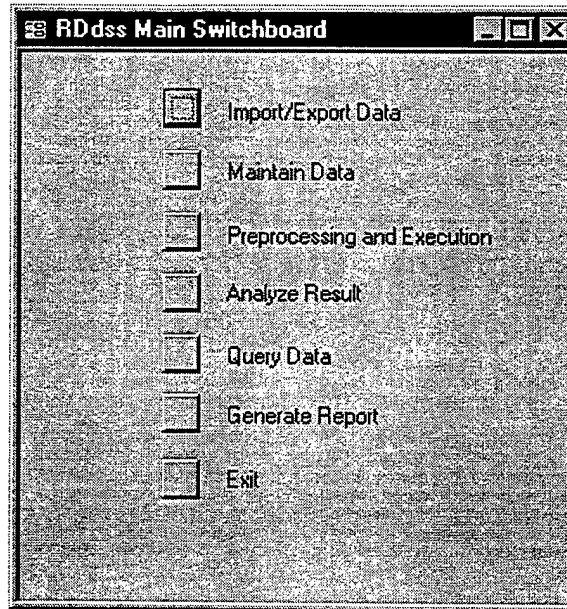


Figure 8: RDdss Main Switchboard

when the application is started. As this illustration depicts, the user can perform the following functions by simply pointing and clicking on the appropriate button: import and export data, maintain data, preprocessing and execution of the assignment model, analyze results, query data, and generate reports. All the RDdss switchboards were generated using Access 97.

The relational database technology was used in the RDdss [Ref. 8]. The actual table relationships are shown in Figure 9. These tables were developed in Access 97, which uses the relational technology. In addition to a few flat files used to communicate with the assignment model, all the data for the RDdss is stored in this database.

The preprocessor is one of the two modeling components. It consists of Access 97 Visual Basic for Applications (VBA) code that computes information necessary for the assignment model [Ref. 9]. It determines the demand of each class, the appropriate penalty associated with each class, and the Marine-to-class fitness matrix. The code for these assignment model inputs is found in Appendix D, lines 452-516 and 4305-4812.

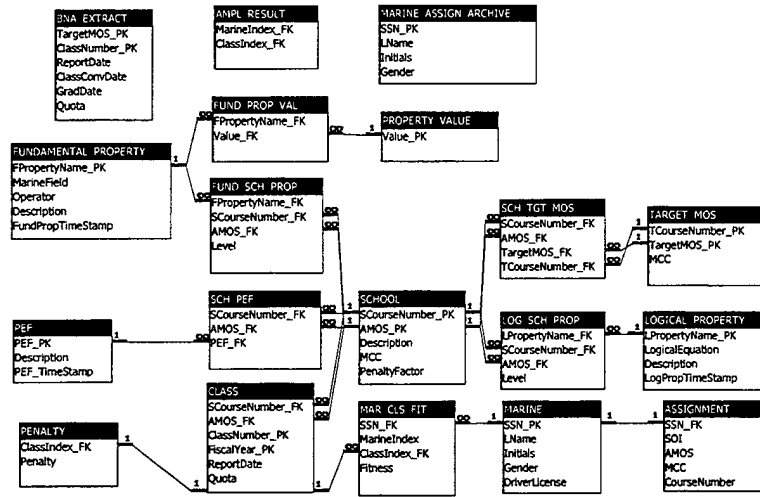


Figure 9. RDdss Relational Schema

The assignment model is the other component of the modeling components. It has been modeled in AMPL [Ref. 10], a language and a computing environment for expressing, solving, and analyzing mathematical programming problems (minimizing or maximizing a function of decision variables, subject to constraints on the variables). A complete discussion of the assignment model's math model is available in Chapter III, Section D.

The next component is the solver. It communicates directly with AMPL, the modeling language of AMPLPlus. As mentioned in Chapter II, it uses an algorithm called CPLEX, which is an optimization package for linear, network and integer programming. Working in conjunction with AMPLPlus, the solver finds an optimal solution.

The final component is the analyzer. It is the other component making up the user interface. This is where the RDdss manager analyzes assignment results, seeking insights for developing a "good" solution. Insight is developed mostly by the graphs and numerical information available in the analyzer. These were generated in Access 97 using its report generator and VBA code. The programming used in the analyzer is shown in Appendix D and includes lines 1-310, 2156-2214, and 3756-3827.

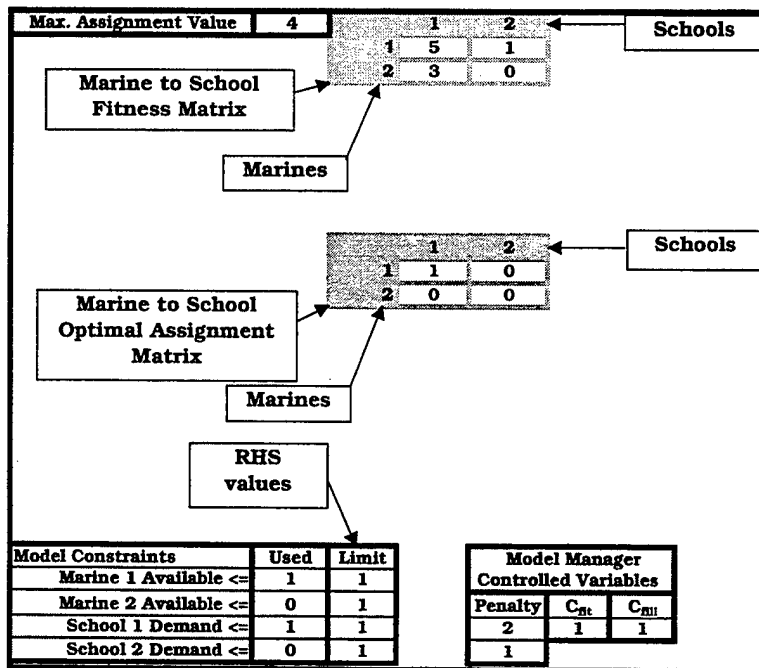


Figure 10. RDdss Demonstration Solver

2. Implementation

Building the RDdss was an iterative process. Some steps took numerous iterations, and some were straightforward. Below is a listing of the 14 major steps it took to build the RDdss.

1. Analyzing the current system was the first step. This included 1) interviewing USMC and DSAI personnel, 2) reviewing available documentation, and 3) operating the RDM.
2. After gathering the current system data, it was necessary to develop a logical understanding of the system. This was accomplished by modeling it with BPWin. Appendix B is the result of this step.
3. With this system understanding, it was possible to envision how the new system would make assignments. Using standard optimization techniques [Ref. 11], a small-scale demonstration solver was built using Excel 97. It is shown in Figure 10.
4. Combining the system understanding and the data requirements for the demonstration solver, it was possible to envision a data structure for the reengineered system. This led to the development of a third normal form relational schema for the RDdss [Ref. 3].

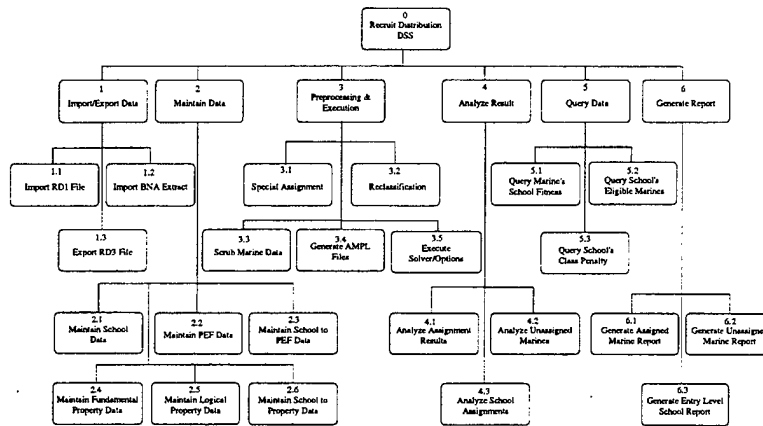


Figure 11. RDdss Decomposition Diagram

5. Using the relational schema as a model, the RDdss tables were built in Access 97. The results of this were seen earlier in Figure 9.
6. As the vision for the reengineered system came more into focus, an intuitive navigational scheme was conceived. This led to the development of a decomposition diagram for the RDdss. It is demonstrated in Figure 11.
7. Using the decomposition diagram as a model, the user interface was built for the RDdss in Access 97. The Main Switchboard was seen earlier in Figure 8.
8. With the switchboards now in place, VBA coding was commenced [Ref. 9]. This made each switchboard functional and useful.
9. Building upon the concept of the demonstration solver from step 3, a formal mathematical assignment model for the RDdss was developed. This was introduced in Chapter III, Section D.
10. The COTS applications, AMPLPlus and CPLEX were installed on the personal computer containing the RDdss.
11. With AMPLPlus now installed, it was possible to code the assignment model into the application. This made it possible to verify our design.
12. Manually creating input files for AMPLPlus is time consuming. So, the pre-processor was coded. This automated the generation of the class demand, class penalty, and Marine-to-class fitness matrix.
13. Further automation of the RDdss was accomplished by interfacing Access 97 and AMPLPlus. This required both VBA and AMPL coding. The end result was a seamless operation of the two applications.

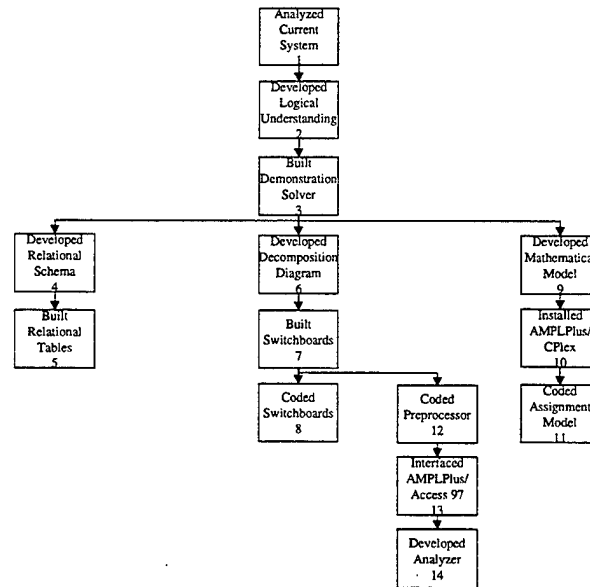


Figure 12. Actual RDdss Build Sequence

14. Finally, much thought was given to ways of gaining insight into the assignment results. This led to the development of the analyzer, which was discussed briefly in the previous section.

Some of these steps were performed in parallel. The actual order followed in building the RDdss are depicted in Figure 12.

B. USING THE RECRUIT DISTRIBUTION DECISION SUPPORT SYSTEM

In this section, we will discuss using the RDdss. Three aspects are covered, 1) setting up a run, 2) model execution, and 3) customizing a run and results. Each is discussed in the order given here.

1. Setting up a Run

Setting up a run of the RDdss is straightforward and easy to do. Figure 13 graphically illustrates the steps involved. Starting with step 1 on the graph and working through to step 3d(iii), we will discuss a run set-up.

The RDdss manager starts at the top of the Main Switchboard. Here, he or

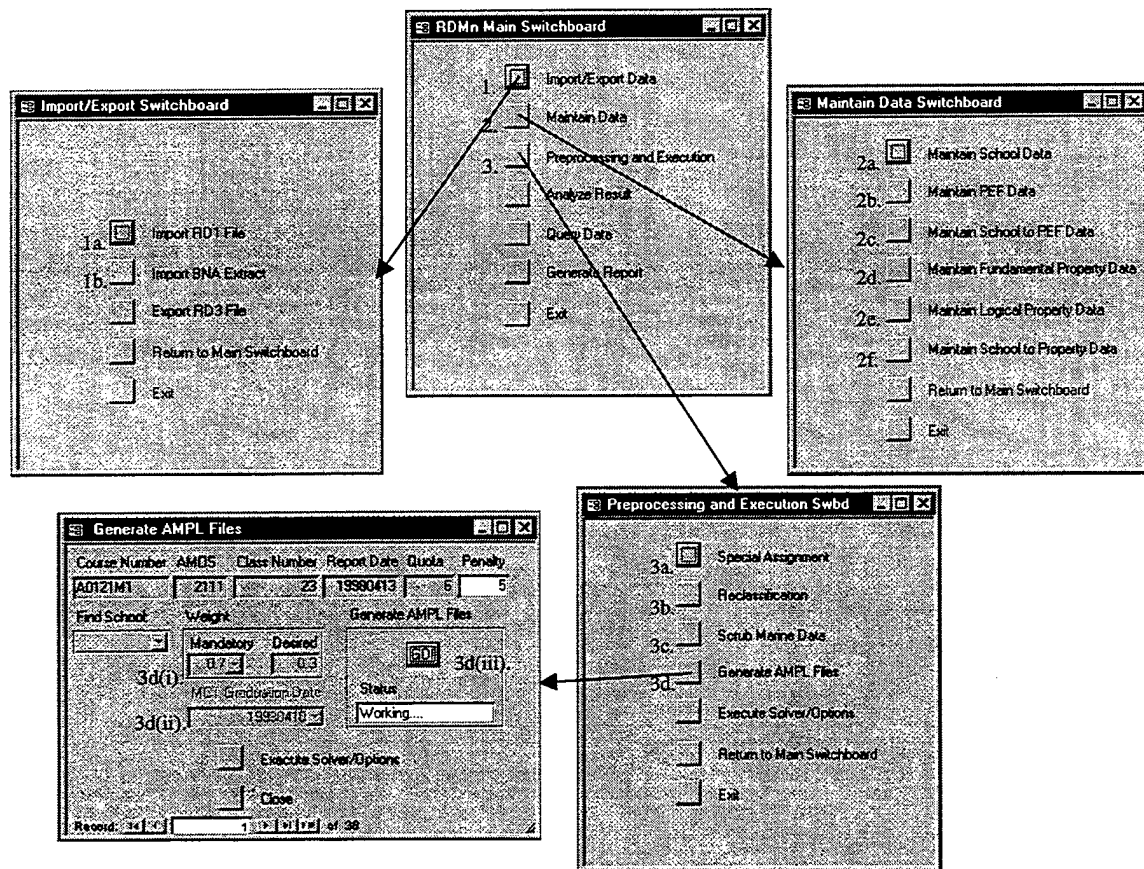


Figure 13. Setting up a Model Run

she clicks on the Import/Export Data button. This will cause the Main Switchboard to close and the Import/Export Switchboard to open. From this switchboard, the RDdss manager will import the RD1 file and the BNA extract into the RDdss MODS. This is easily accomplished by clicking on the corresponding buttons for importing these files. Finally, the RDdss manager will click on the “Return to Main Switchboard” button, causing the Import/Export Switchboard to close and the Main Switchboard to reopen.

The next step is data maintenance. This includes updating, editing, and/or deleting of school related information. The RDdss manager gets to the Maintain Data Switchboard by clicking on the appropriate button on the Main Switchboard.

From the Maintain Data Switchboard, the RDdss manager will start at the

top and work his or her way down to the bottom. The various buttons on this switchboard open forms where data maintenance is performed. For instance, the first button opens the Maintain School Data form, allowing the RDdss manager to create, edit, and/or delete a school. The other buttons allow similar operations with the PEFs, and fundamental and logical properties. After the RDdss manager has finished maintaining the data, he or she returns to the Main Switchboard by clicking on the "Return to Main Switchboard" button.

After data maintenance, the RDdss manager is ready to perform data preprocessing in preparation for a model run. By clicking on the Preprocessing and Execution button, the Main Switchboard closes and the Preprocessing and Execution Switchboard opens. From here, as on all other switchboards, the RDdss manager starts at the top and works down.

The first three preprocessing buttons directly effect the generation of the class demand, class penalty, and Marine-to-class fitness matrix. This is why they come before the fourth button, Generate AMPL Files. Preprocessing in the correct order will reduce the amount of time spent setting up the run, by minimizing the need to generate a new class demand, class penalty, and Marine-to-class fitness matrix.

The first preprocessing button concerns special assignments. Specially assigning a recruit affects the AMPL input in two ways. First, the respective member is not included in the Marine-to-class fitness matrix, since they already have an assignment. Second, the class demand for the class the recruit is assigned to must decrease by one. Otherwise, it may become overfilled.

The second preprocessing button concerns reclassification of a Marine. Reclassified Marines affect the AMPL input in two ways. First, the class demand of the class they are no longer assigned to must increase by one. This provides an opportunity to fill the empty seat during a subsequent run. Second, the class demand of the class the Marine has been reassigned to must decrease by one. Otherwise, it may become overfilled.

The third preprocessing button concerns data scrubbing, which also affects the AMPL input. It is an attempt to identify and correct errors relating to the Marines. An example is a Marine who has been given an unknown PEF. Since it is not identified by the RDdss, the member will receive a zero fitness score for every class. This will prevent the Marine from getting an assignment.

Following the completion of special assignments, reclassifications and data scrubbing, the RDdss manager is ready to generate the AMPL input files. After going to the Generate AMPL Files form, the first concern is specifying the weights for the mandatory and desired properties. This is specified by the RDdss manager at step 3d(i). What this does is indicate how important the mandatory properties are with respect to the desired properties. For instance, if the RDdss manager decided to give a weight of 1.0 to the Mandatory selection, the Desired selection would automatically fill in with 0.0. This would indicate the desired properties are of no consequence. The default for these two selections is 0.7 for Mandatory, and 0.3 for Desired.

Next, at step 3d(ii), the RDdss manager selects the MCT graduation date desired. This is selected from the drop down list. Then, the RDdss manager pushes the "GO!" button. The system will now generate the AMPL input files, completing the run set-up.

2. Model Execution

Execution of the assignment model takes place after the run set-up has been completed. Figure 14 graphically illustrates the steps taken to execute the model. A discussion of this is now given.

From the Main Switchboard, the RDdss manager points and clicks on the Preprocessing and Execution button. This takes him or her to the corresponding switchboard. From here, the Execute Solver/Options selection is made. This opens up the form where the model is executed from.

The Execute Solver/Options form provides the RDdss manager the ability to set the fit and fill coefficients for the objective function. The default values are 1 for

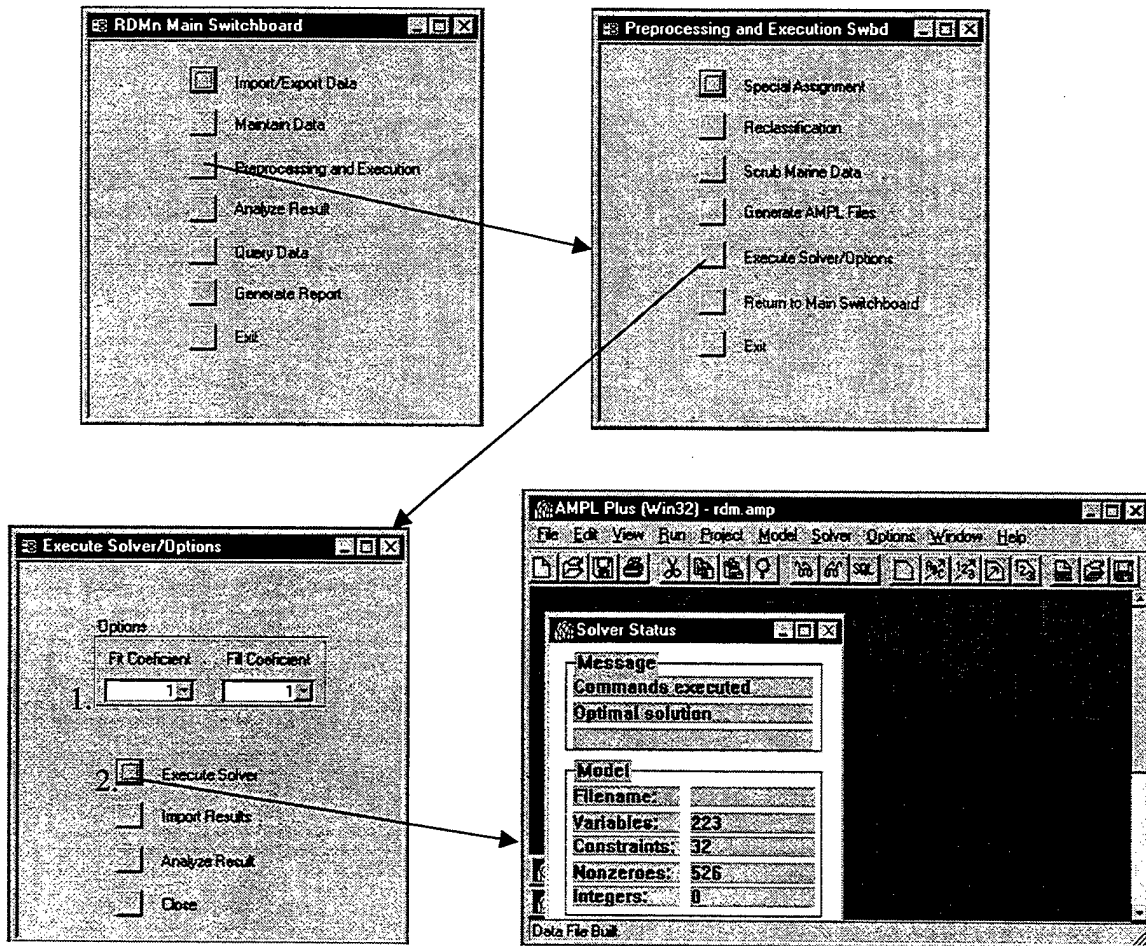


Figure 14. Model Execution

each coefficient. The RDdss manager can change either of these with a whole number from 0 - 100. The significance of these numbers is discussed in detail later in this chapter. Once the desired coefficients have been chosen, the Execute Solver button is pushed.

By selecting the Execute Solver button, AMPLPlus is launched. This application will run without user interaction. First, it will import the RDdss mathematical assignment model. Next, the specified fit and fill coefficients are retrieved. Finally, the preprocessed data generated during the run set-up are imported. Using this input data, AMPLPlus works in conjunction with the solver and generates an optimal assignment solution.

With the generation of an optimal assignment solution, the results are sent to an output file called "rdm.out." AMPLPlus, having completed its task, now closes. This indicates an optimal solution was found and is available for importing. So, the RDdss manager now imports the results by pushing the Import Results button on the Execute Solver/Option form.

3. Customizing a Run and Results

As mentioned in the previous subsection, the RDdss manager can set the fit and fill coefficients of the objective function. This is a key capability of the new system. It is the means by which the RDdss manager can customize a run and "game" the RDdss. This "gaming" of the new system is essential for determining a "good" assignment result.

For instance, if the fit coefficient is set to 0, and the fill coefficient is set to a number > 0 , say 1, the assignment model will maximize the fill of schools having early report dates. This has both advantages and disadvantages. One advantage is the average wait time for Marines to attend a school is minimized. Another is that the number of unfilled school seats, having a report date between the current and next scheduled model run, is minimized. However, a disadvantage is the average Marine-to-school fitness score is guaranteed to be the lowest score of all runs.

Conversely, if the fit coefficient is set to a number > 0 , say 1, and the fill coefficient is set to 0, the assignment model will maximize the average Marine-to-school fitness. An advantage here is the increased chance of each Marine successfully completing their ELS, establishing a pattern of success early in their career. However, one disadvantage is the wait time for a Marine to attend a school is guaranteed to be the longest. Also, the number of unfilled school seats, having a report date between the current and next scheduled model run, is guaranteed to be the greatest.

Neither of the above solutions is ideal. A better solution is some where between these two extremes. By changing the fit and fill coefficients of the objective function, the RDdss manager has a means for customizing each run to ensure a "good"

assignment result is found.

C. INNOVATIONS IN USER INTERACTION

A number of user interaction innovations have been made in the RDdss. Two of them have already been discussed. These were the intuitive navigational scheme using the switchboards, and the elimination of manually entering data already available in the system. One more note worthy innovation is now introduced. It is a date change innovation.

The date change innovation concerns changing the graduation date of the Marines in the downloaded RD1 file. This is necessary for generating correct assignment results. The date in the downloaded RD1 file is based on graduation from the MCRD. However, the model needs to know when the Marines are available for assignment. Availability is based on the MCT graduation date.

The old system solution to this problem took three steps. First, the RDM manager used a calendar to determine the MCT graduation date. This is normally 27 days following completion of MCRD, and is always on a Friday. So, the RDM manager will find the MCRD graduation date on the calendar and count 27 days. The Friday nearest to this day is the day desired. Second, this MCT graduation date, along with the RD1 file, is sent to another branch in the USMC where the date change is made. Third, the file is returned to the RDM manager, who imports the file into the RDM.

The reengineered system has streamlined this process. Within the RDdss, the RDdss manager makes the date change. It is accomplished with the push of a couple buttons.

Figure 15 shows how these changes are made. First, the RDdss manager selects the MCRD graduation date requiring a change. Once this is selected, a calendar appears. Its purpose is to help the RDdss manager correctly identify the MCT graduation date. Next, the "=" button is pushed. Its purpose is to add the number of

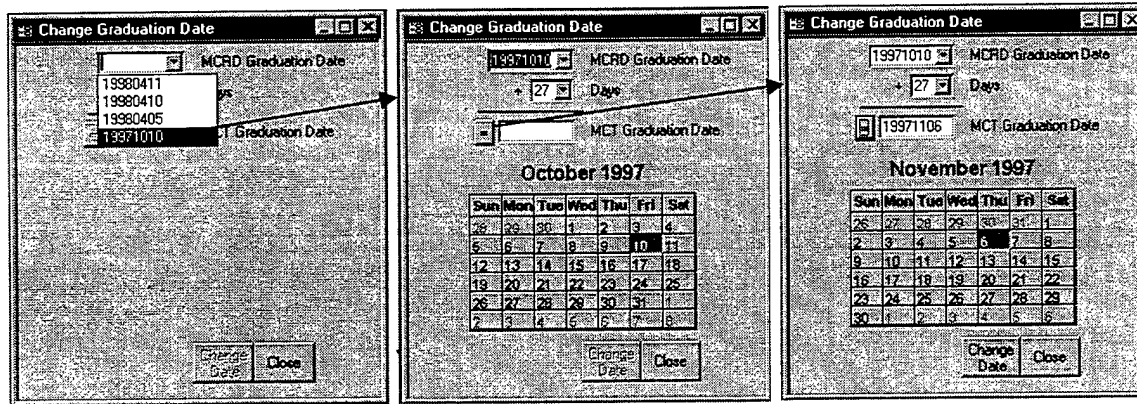


Figure 15. Changing a Graduation Date

days specified in the “Days” field (default is 27), to the MCRD graduation date. This updates the calendar as shown on the far right. When the correct date is found, the “Change Date” button is pushed. This makes the desired change to the data in the RDdss database.

D. INNOVATIONS IN ANALYZING MODEL RESULTS

As discussed earlier, the RDdss manager has the capability of “gaming” the system by using different fit and fill coefficients. However, having this ability is not enough. The RDdss manager still needs a means to compare one run against another.

It is hard, if not impossible, to estimate the quality of a solution simply by looking at a list of assignments. The actual assignment results depend on the values of the fit and fill coefficients, which cannot be set at the ideal levels without understanding the impact on solution quality. So, analyzing run results to get insight and develop a “good” solution is now discussed.

Four measures are considered particularly useful in evaluating alternative solutions.

1. Fitness premium. The Marine-to-school fitness function is defined in a way that for each school, the average fitness over all Marines *eligible* for that school is 100. So, for any school, if the average fitness computed over Marines *assigned*

to that school is greater than (or less than) 100, the difference represents a positive (or negative) premium.

2. Unfilled seats in first week. Since the model is run nearly every week, seats left unfilled in classes starting in the first week will remain unfilled. This represents wasted resources.
3. Average wait time for Marines. Experience has shown that the longer a Marine waits before attending their ELS, the greater the chance of disciplinary problems. Therefore, it is generally better to keep this wait period short.
4. Unassigned Marines. A purpose of the system is to assign each Marine to a school. Therefore, knowing the number of unassigned Marines is an important measure.

To provide the RDdss manager insight into the assignment results, these measures have been incorporated into output representations, including summary statistics and detailed graphs. Each output provides both numerical and visual information. The numbers are used for quantitative comparisons. The graphical information provides a big picture view for use in a qualitative comparison.

As an example, consider the two fitness graphs in Figure 16. Both of these were produced using an RD1 file of 344 Marines and an BNA extract with school data covering a 120 day period. Below each graph is amplifying and numerical information. The bar chart on the left has a fit coefficient of 0 and a fill coefficient of 1. Additionally, its fitness premium is 28 ($=128-100$). The bar chart on the right has a coefficient of 1 for both fit and fill. Its fitness premium is 38 ($=138-100$). The fit numbers are quantitatively comparable. The fitness premium for the run on the right is 36% ($=100\% \times [38-28]/28$) better than the run on the left.

The above numerical analysis is also supported by visually comparing the bar charts. Notice the large number of blips falling below the fitness score of 100 on the left graph. Each blip represents a Marine, and there are actually 44 of them. Conversely, notice the comparatively small number of blips below 100 on the right graph. There are 21 of them. The visual comparison of the two graphs reveals the

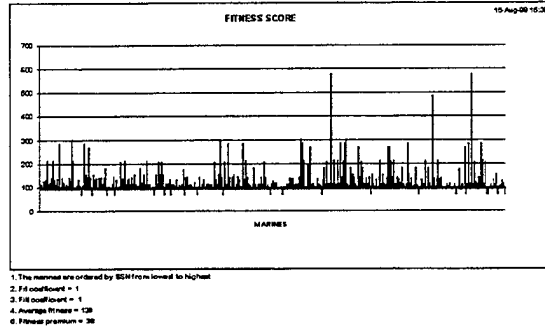
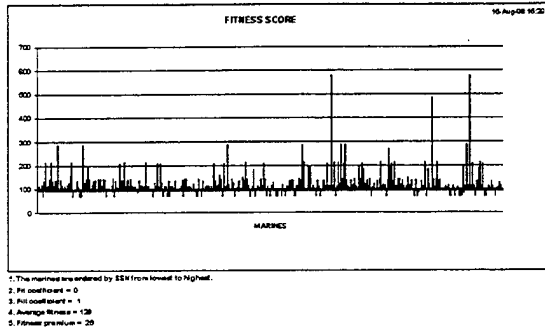


Figure 16. Two alternative results: Fit

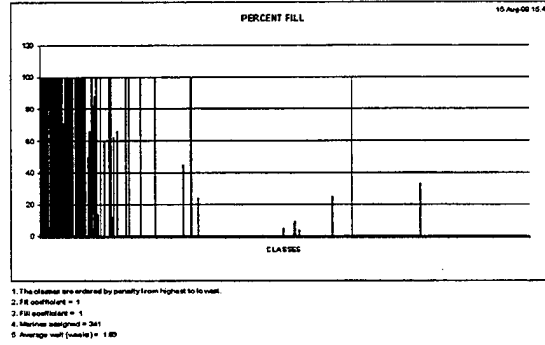
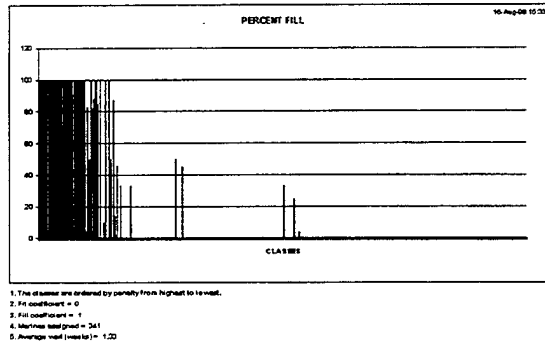


Figure 17. Two alternative results: Fill

qualitative difference. The graph on the right indicates a higher quality Marine-to-school fit. Of course, this supports the numerical comparison performed earlier.

As another example, consider the two fill graphs in Figure 17. These are based on the two model runs just described. In other words, the left graph has a fit coefficient of 0 and a fill coefficient of 1, and the right graph has a coefficient of 1 for both fit and fill.

Visually, these two graphs display an obvious difference. To appreciate how they are different requires an understanding of the significance of the blips. As note 1 of each graph indicates, "The classes are ordered by penalty from highest to lowest." Recall from Chapter III, this means the ordering is by class report date, starting with the earliest date. In other words, the bar chart on the left indicates almost every Marine will start school within the first three weeks. As discussed earlier, this is desirable. The bar chart on the right has its blips more spread out. So, by visually

comparing the two graphs, the left one has a higher quality with respect to minimizing wait time.

This is easily validated by the numbers given in note 5 of each graph. The average wait of a Marine from the left bar chart is 1.33 weeks, and the average wait from the other bar chart is 1.69 weeks. Now we quantitatively compare these values. The results from the left graph are 27% ($=100\% \times [1.69-1.33]/1.33$) better than the right.

In the following table, the results from three runs are given. The measures on the left are the four measures described earlier. Each run used the same 344 Marines and 120 days worth of school classes. The information for the first and third runs is from the two model runs discussed above. The second run was determined by “gaming” the system. In other words, we tried numerous combinations of the fit and fill coefficients until a “good” solution was found. Run 2 was the result. Understand though, we are not saying “best” solution. Only the USMC can determine what is best for them.

	Run 1 (fit=0; fill=1)	Run 2 (fit=1; fill=11)	Run 3 (fit=1; fill=1)
Unfilled Seats (1st week)	23	23	23
Unassigned Marines	3	3	3
Average Wait (weeks)	1.33	1.33	1.69
Fitness Premium	28	31	38

The reason we feel Run 2 is a “good” solution is based on a quantitative comparison of the measures and one additional piece of information. It is the fact the USMC feels it is more important to minimize wait time than to maximize the Marine-to-school fit. The remainder of this section explains how we decided Run 2 was a “good” solution.

Starting with the values for unfilled seats, it is obvious all three runs are equal. The same is true for the unassigned Marines measure. So, these two measures provide no insight for comparison purposes.

Concerning the average wait, the values of Runs 1 and 2 are equal. Further, their 1.33 value is numerically smaller than the 1.69 value of Run 3. This is good for Runs 1 and 2. As was proven earlier, a value of 1.33 is 27% better than 1.69.

Finally, there is the fitness premium measure. The first observation made is that Run 1 is a candidate for elimination. The reason is as follows. Run 2 has the same score as Run 1 for the first three measures in the table, but it shows an 11% ($=100\% \times [31-28]/28$) increase in the fitness premium. So, Run 1 is eliminated from the competition.

This leaves the last two runs. Run 3 shows a 23% ($=100\% \times [38-31]/31$) increase in fitness premium over Run 2. This creates a conflict. Run 3 has a 27% increase in average wait time, which is bad. Conversely, it has a 23% increase in fitness premium, which is good. To decide which solution is better, more information is necessary. This comes from the fact mentioned earlier, that the USMC feels it is more important to minimize wait time than to maximize the Marine-to-school fit.

We can now conclude Run 2 is a better solution than Run 3. Here is why. Run 2 minimizes average wait by 27% by giving up a 23% increase in Marine-to-school fit. This agrees with the USMC desire to more strongly emphasize a minimum waiting time. So, Run 2 is a better solution than Run 3. Since it also satisfies the known desires of the USMC, we feel it is a "good" solution as well.

E. OBJECTIVE COMPARISON OF OLD AND NEW SYSTEM SOLUTIONS

In the previous section, we described how a set of solutions generated by the RDdss are compared to determine a "good" solution. Using the same approach, we now demonstrate how to objectively compare the results from the old system with the new system. This provides the USMC a means of determining which system's solution is better, if either. Comparison of the new and old system results has been made easy by automating the importing of the RDM result into the RDdss.

We have summarized the important data for the old and new systems in the following table, based on the same 344 Marines and 120 days of school classes discussed earlier. Since the assignment algorithm for the RDM was designed to maximize fill, a run from the RDdss based strictly on fill has also been included. It is the run with a fit coefficient of 0, and a fill coefficient of 1. The last column, labeled "Run 2," is a "good" solution we found by gaming the RDdss.

	Run 1 (fit=0; fill=1)	RDM	Run 2 (fit=1; fill=10)
Unfilled Seats (1st week)	23	23	23
Unassigned Marines	3	3	3
Average Wait (weeks)	1.33	1.34	1.34
Fitness Premium	28	29	32

We start by objectively comparing the RDdss fill-based result with the RDM result. They are labeled "Run 1" and "RDM," respectively. As we shall prove, the RDdss has done a better job at filling early school seats than the RDM.

Of the four measures, only one provides any useful insight for this comparison. It is the average wait measure. The reason the other three do not apply is as follows. The first two measures have the same values, so they provide no insight. The last measure is not too meaningful, since Run 1 is strictly based on maximizing fill. In other words, it disregards fit because its fit coefficient is set to 0. This leaves the average wait measure.

A quantitative comparison of the average wait shows the RDdss solution is 1% ($100\% \times [1.34 - 1.33] / 1.33$) better than the RDM solution. This is explainable by considering the approach used in each model. The RDM assignment algorithm maximizes the fill of early school seats by looking at each school in a sequential fashion. In contrast, the RDdss conducts a global comparison of Marines and schools. In other words, it goes through numerous iterations until it finds an optimal fill solution. This shows that the RDM assignment algorithm is only closely approximating the optimized assignment result.

Finally, we “gamed” the new system until a “good” solution was found. This is Run 2. It has a fit coefficient of 1 and a fill coefficient of 10.

The objective comparison of the RDM and Run 2 solutions is straightforward. Since the first three measures are the same for both results, they provide no insights. This leaves only the fitness premium measure. A quantitative comparison of this measure shows Run 2 has a 10% ($100\% \times [32-29]/29$) better solution than RDM. This shows the benefit of “gaming” the RDdss.

F. SUMMARY

This relatively long chapter covered some important points related specifically to the RDdss. The architectural components of the new system were discussed first, followed by the steps taken to actually build this DSS. Then, from an operational point of view, using the RDdss was discussed. This included setting up a run, executing the model, and customizing a run for determining a “good” solution. Next, two innovations were discussed in great detail. These included the date change innovation and the analyzing model results innovation. Finally, a means of objectively comparing the solution from the old and new systems was discussed.

We truly believe the innovations and capabilities introduced by this prototype can greatly benefit the USMC’s Marine Enlisted Assignments branch. It can help the USMC to more effectively accomplish its mission as stated by the Commandant of the Marine Corp, “to put the right Marine in the right place at the right time with the right skills and quality of life.”

V. CONCLUSION

This concluding chapter has three short sections, followed by a final comment. In the first section, we reiterate the significant contributions made in the thesis. This is followed by a brief description of some lessons we learned. Then, some recommended improvements for the prototype are given. Following this last section, we comment on how this thesis work might benefit the other branches of the United States military.

A. SIGNIFICANT CONTRIBUTIONS

Four significant contributions beneficial to the USMC were made in this thesis. These were discussed in detail in the previous three chapters. A list and summary of the contributions is given here.

- Analysis and articulation of the recruit distribution process - by interviewing USMC and DSAI personnel, reviewing available documentation, and operating the RDM we were able to articulate an understanding of the recruit distribution process using IDEF process modeling. Additional data modeling was articulated in a third normal form relational schema.
- Development of a mathematical model - by analyzing the assignment process, criteria, and constraints, USMC policies and objectives, a mathematical model for the new system was developed.
- Fully functional prototype - an intuitive and easy to understand new system that seamlessly interfaces with the old system was built. It provides an objective means of comparing assignment results of both systems, and was developed using COTS software applications.
- Elimination of the proprietary solver and associated contractor lock-in - this was accomplished by replacing the proprietary solver with two COTS applications.

B. LESSONS LEARNED

Our original goal was to build an As-Is IDEF0 model of the RDM, about two months after starting the thesis. This worked out fine. The next goal was to take

another six weeks to develop a To-Be IDEF0 model [Ref. 1]. This did not work out, for two reasons. First, our understanding of the entire recruit distribution process was still not mature enough to build the To-Be model. Nearly half a year was spent just thinking about its many details. Second, it is nearly impossible to envision the new system without first knowing the capabilities and limitations of the application used to build the prototype. One extreme envisions the impossible, while the other barely touches the reengineered system's potential. *The lesson learned from this was that properly reengineering a process requires two things. A detailed understanding of the current process environment, and a good working knowledge of the capabilities and limitations of the application used to build the prototype.*

Getting feedback from the Marine Enlisted Assignments branch was not always easy. Correspondence normally occurred by e-mail. There was an inverse relationship between the number of questions asked and the number of answers given. As the one increased, the other decreased. *So, another lesson learned was that it is best to limit each correspondence to a question or two.*

While writing code, it is best to simultaneously document the code's purpose. Going back and trying to provide documentation at the end is too frustrating. This requires a considerable amount of discipline, but is well worth the effort. *The lesson learned here is document as you code.*

C. PROTOTYPE IMPROVEMENTS

A big part of this thesis involved the building and refinement of the prototype. The following topics are areas where the RDdss could use improvements.

1. Speed Improvements

a. Preprocessing Data

Of all the steps necessary to complete a run of the RDdss, the preprocessing step is the most time consuming. For the 344 Marines and approximately 500 school classes analyzed in this thesis, it takes about 30 minutes to preprocess

this data. This is about ten times longer than the assignment model takes to find an optimal solution. As the size of the preprocessing data is doubled, so is the time needed to preprocess it. Finding a way to speed up this process is desirable. One possibility is to find a way to perform all the preprocessing in main memory, with only two accesses to the hard drive. The first access is to get the necessary Marine and school data. The second is to write the preprocessed results back to the hard drive.

b. Analyzing Result

Compared to all other forms and switchboards in the RDdss, opening the "analyze results" forms is the most time intensive. With the 344 Marines and approximately 500 school classes, it takes just under a minute for any of these three to open. This feels like an eternity when you are in a hurry. The reason it takes so long is the creation of temporary tables, numerous calculations and comparisons, and retrieving of data from flat files. Finding a way to reduce this time is desirable. One possibility is to create another form which would contain only the four objective measures. Currently, the RDdss manager must go to two different forms for this information. Each containing information and displays in addition to the objective measures, which slow them down.

2. Multiple Solution Storage

Currently, the prototype will only save the results of one solution. This means the RDdss manager must either print-out hard copies of the graphs, or manually write down each solution's objective measures for comparison with other runs. An improvement here should allow the storage of at least three solutions. This would permit the RDdss manager to easily compare each solution. He or she could then eliminate one or two undesirable solutions, while continuing to search for a "good" one.

3. Administration Switchboard

To look at all the schools or properties in one display, the RDdss manager must open up the associated school or property table. A reason for doing this is system administration. For instance, if a supervisor is auditing the RDdss's property data, looking at all the data at one time is convenient. A simple way of including this feature is by adding another switchboard. The switchboard's buttons would have VBA code associated with them, causing the desired tables to open.

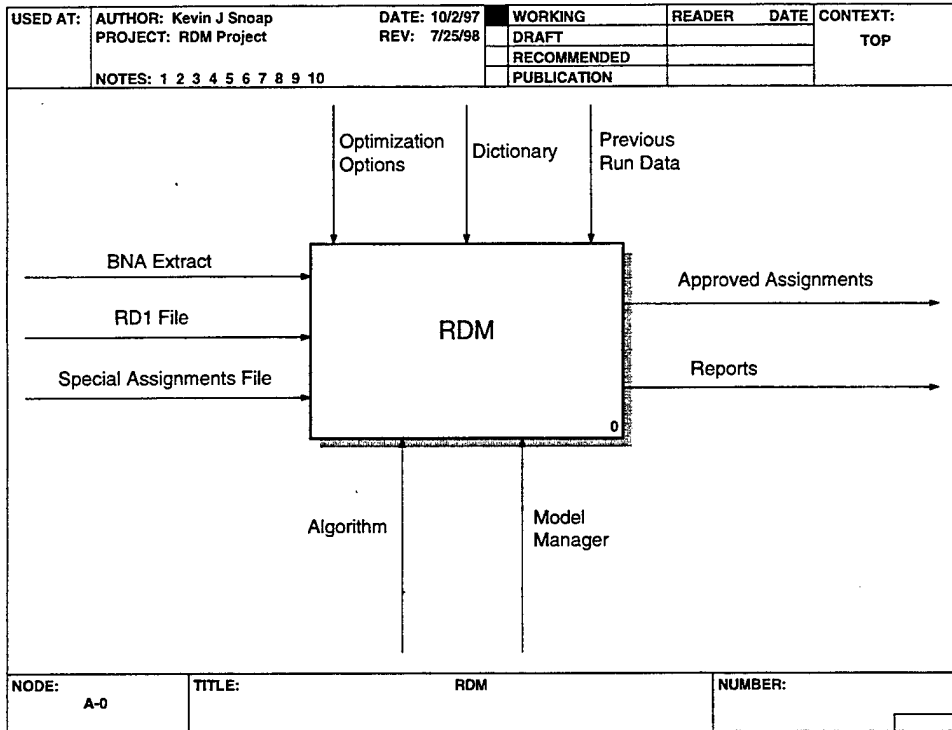
Any benefits derived from this thesis work were specifically aimed at the USMC. However, it does not have to end there. Like the USMC, other branches of the United States military must deal with the problem of how best to assign their service members to schools. Since all the services require their enlisted personnel to take the ASVAB test, the approach used in our prototype might prove an ideal solution. Further, our efforts may also contribute to the reengineering of other assignment models. So, this thesis work could actually benefit all branches of the United States military.

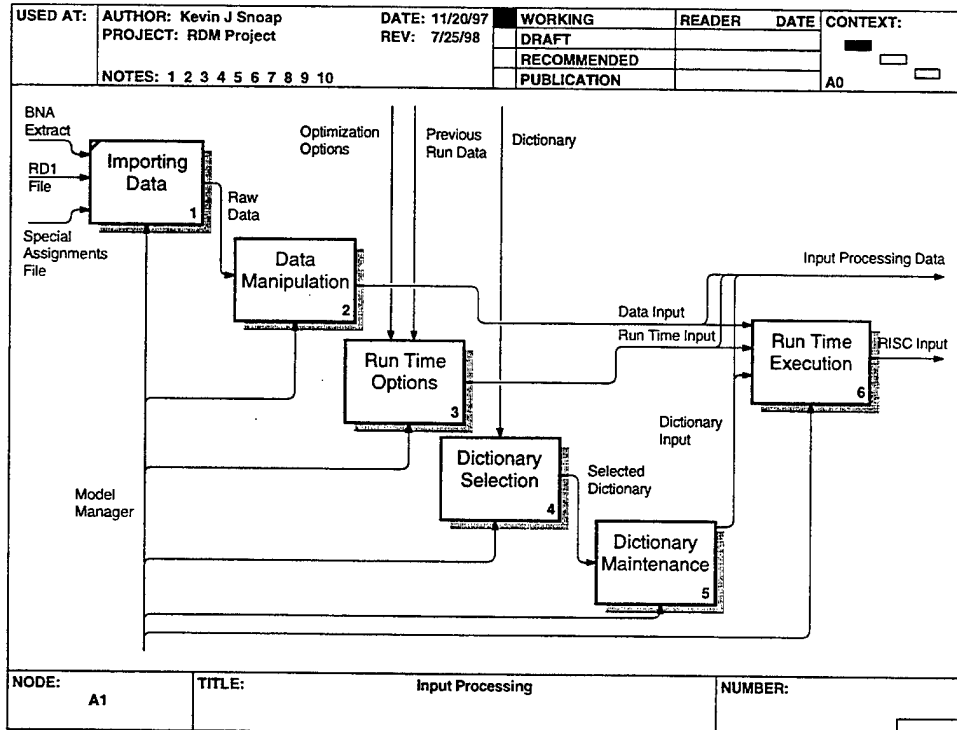
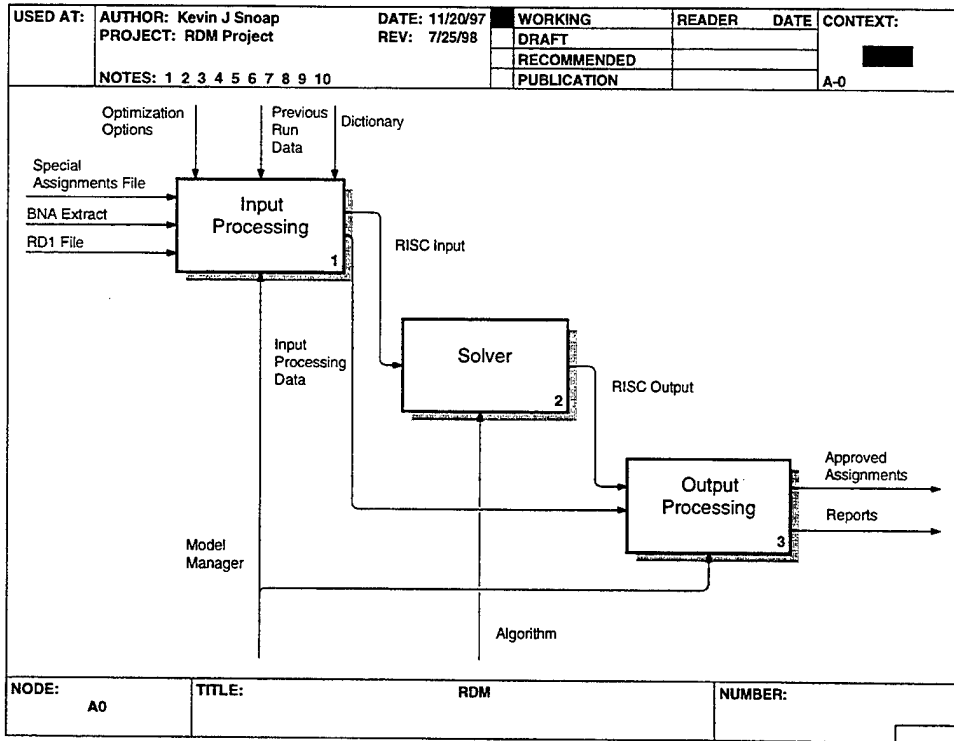
APPENDIX A. ACRONYMS

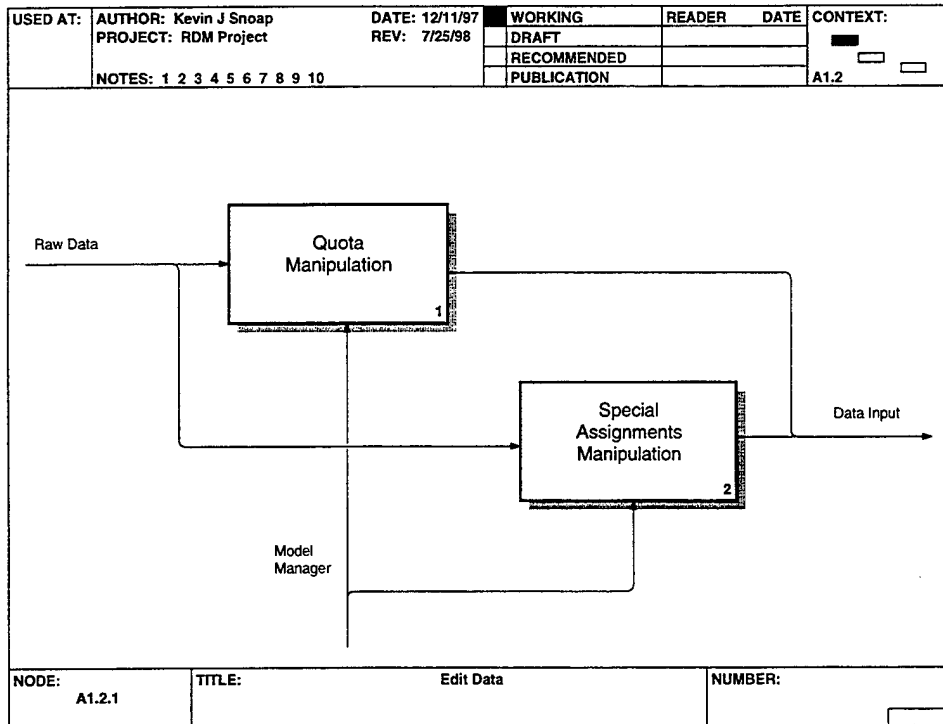
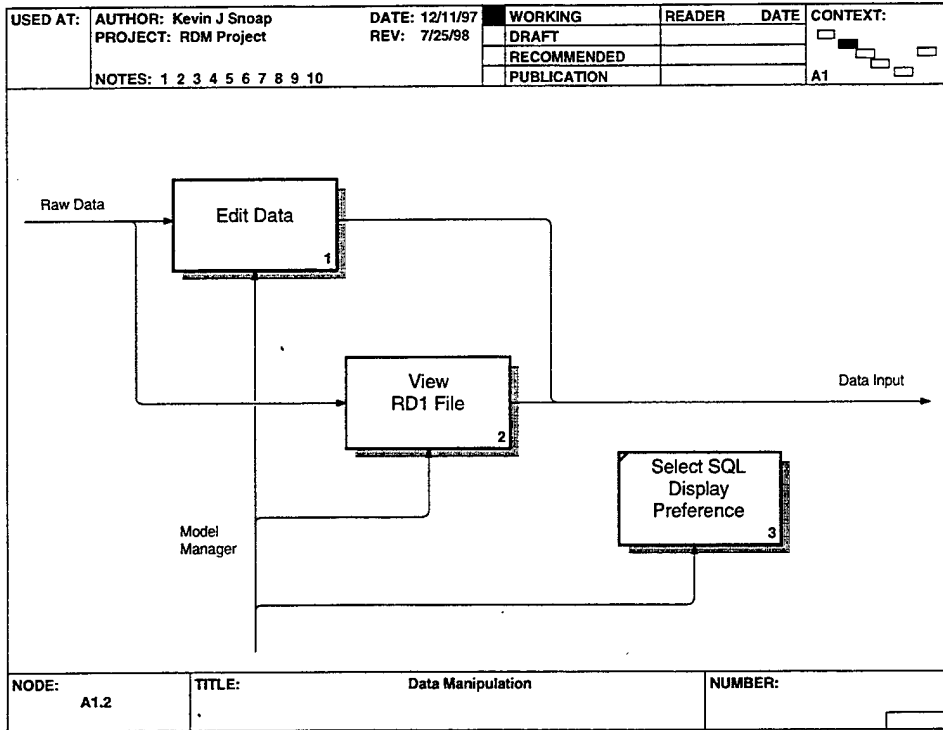
Acronym=>Meaning

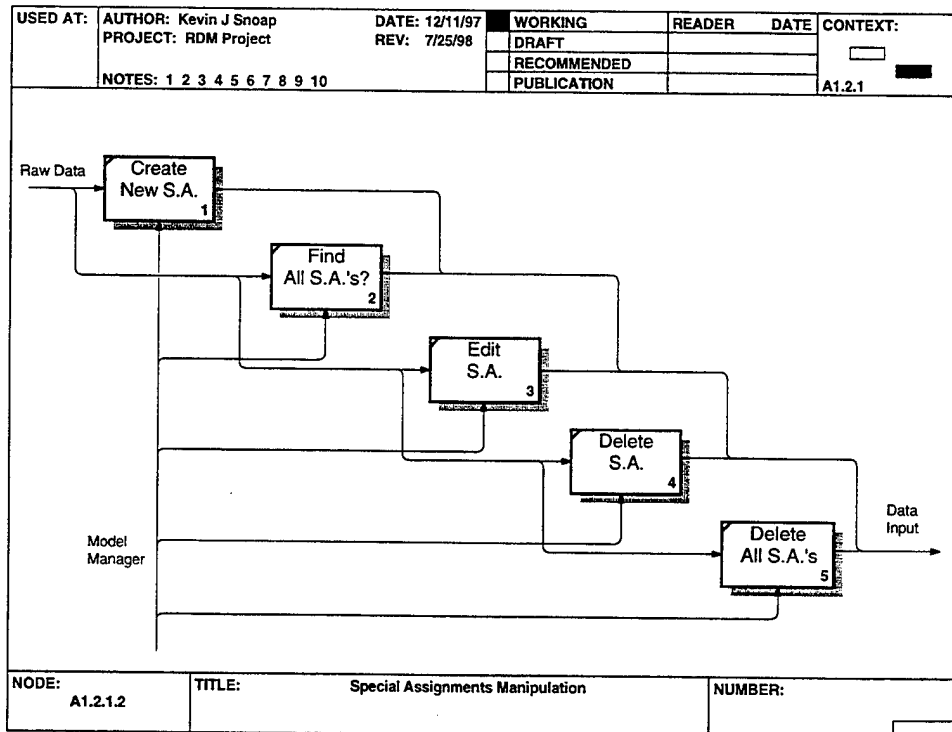
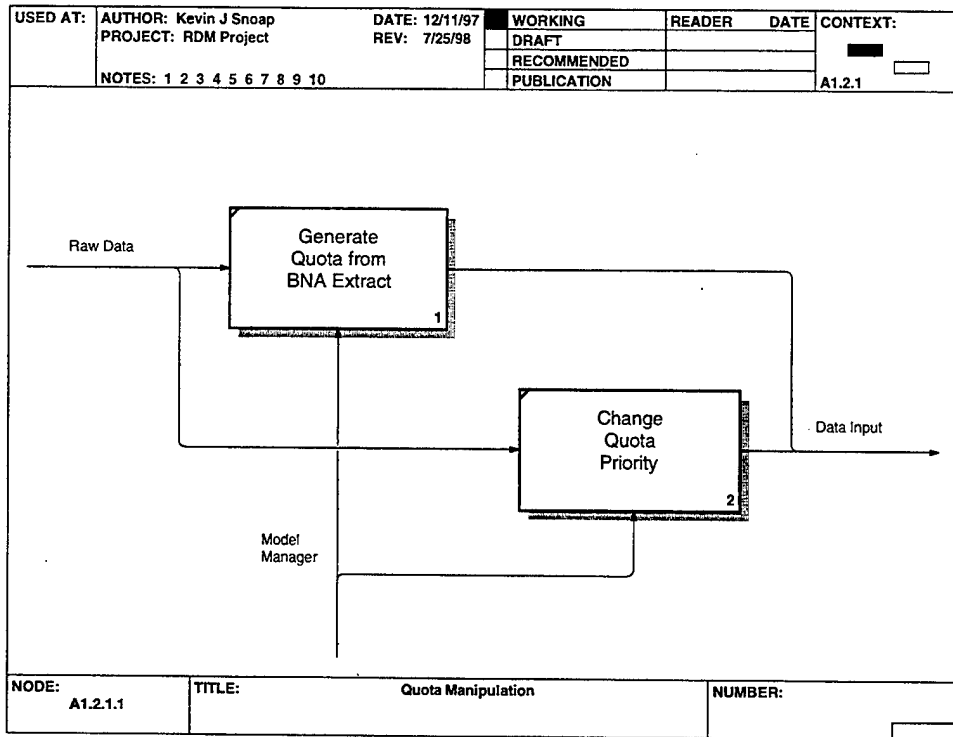
AMPL	A Modeling Language for Mathematical Programming
ARMS	Automated Recruit Management System
ASVAB	Armed Services Vocational Aptitude Battery
BNA	By Name Assignment System
BPWin	Business Process for Windows
COTS	Commercial-off-the-shelf
CPlex	Optimization Package for Complex Linear, Network and Integer Programming
DSAI	Decision Support Associates, Inc.
DSS	Decision Support System
ELS	Entry Level School
ICOM	Input, Control, Output, Mechanism
IDEF	Defense Institute Modeling
IDEF0	Business Process Modeling
MCRD	Marine Corps Recruit Depot
MCT	Marine Combat Training
MODS	Military Operational Data Store
MOS	Military Occupational Specialty
PEF	Program Enlisted For
RD1	Recruit Distribution Input File
RD3	Recruit Distribution Output File
RDdss	Recruit Distribution Decision Support System
RDM	Recruit Distribution Model
USMC	United States Marine Corps
VBA	Visual Basic for Applications

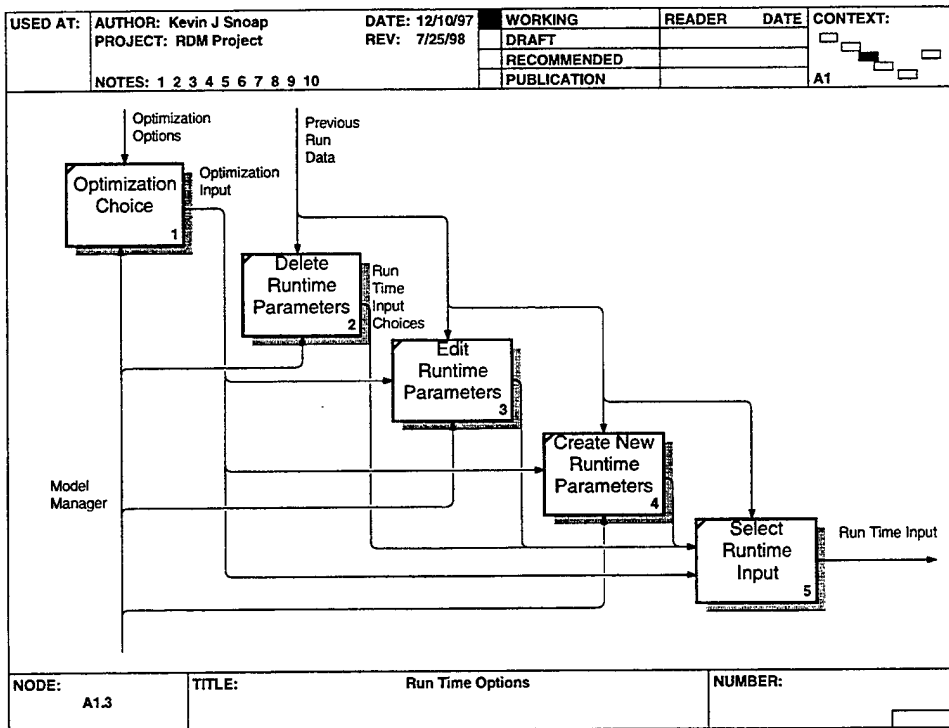
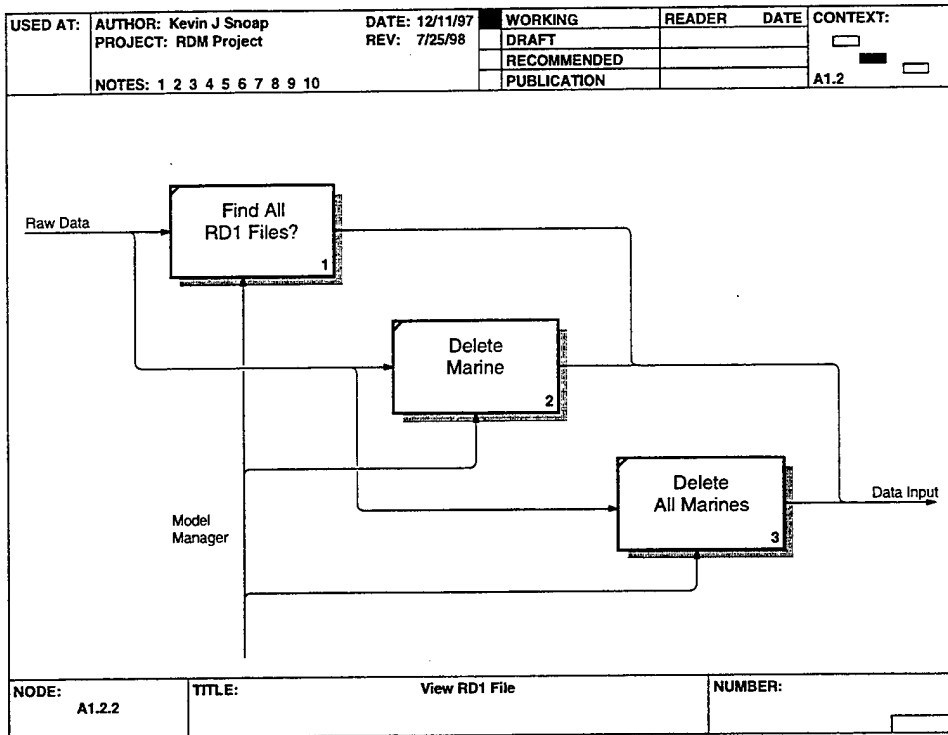
APPENDIX B. AS-IS RD BUSINESS PROCESS (IDEF0) MODEL

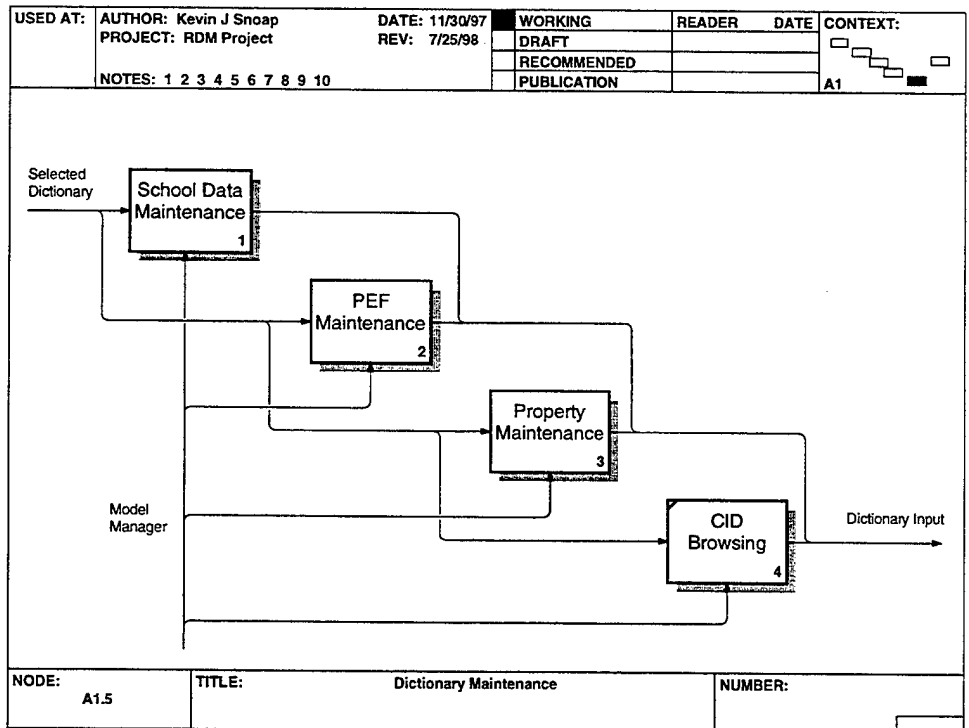
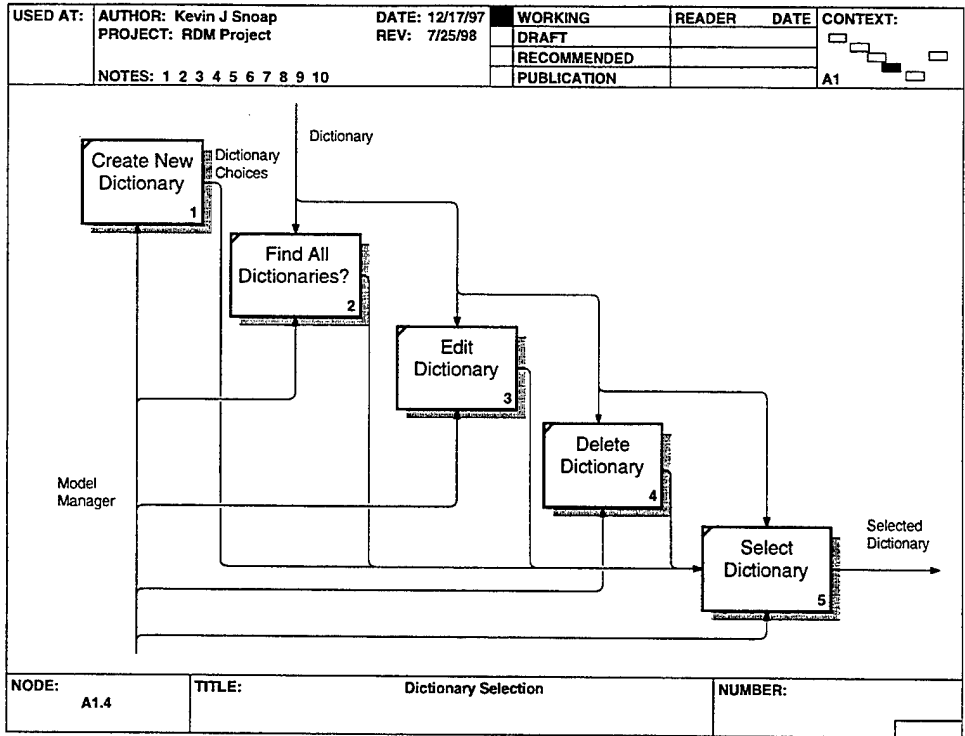


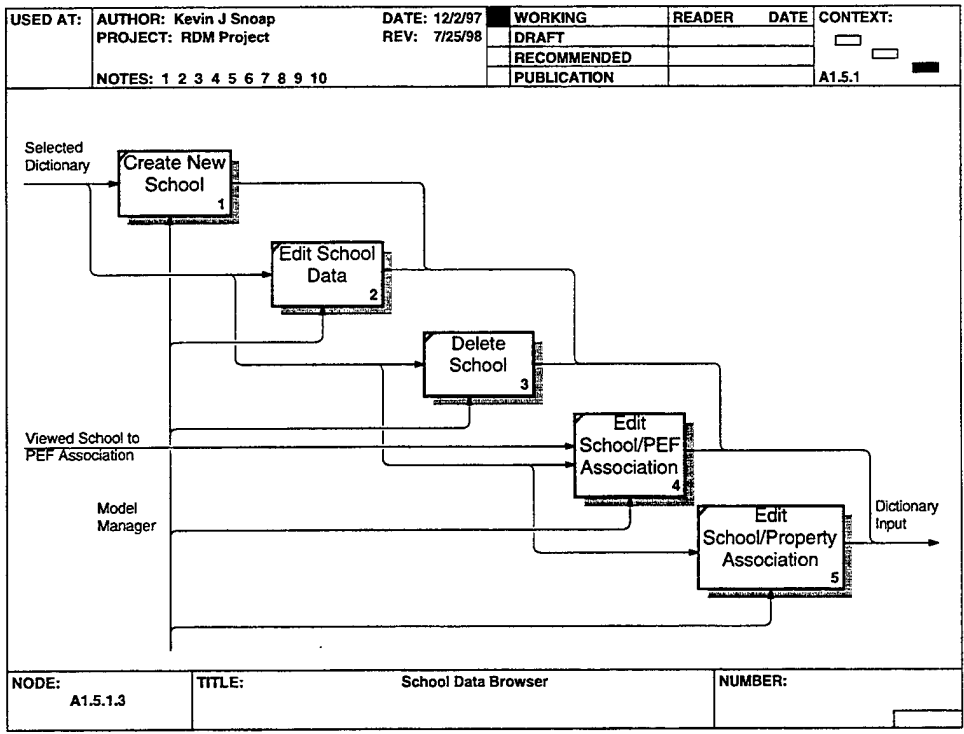
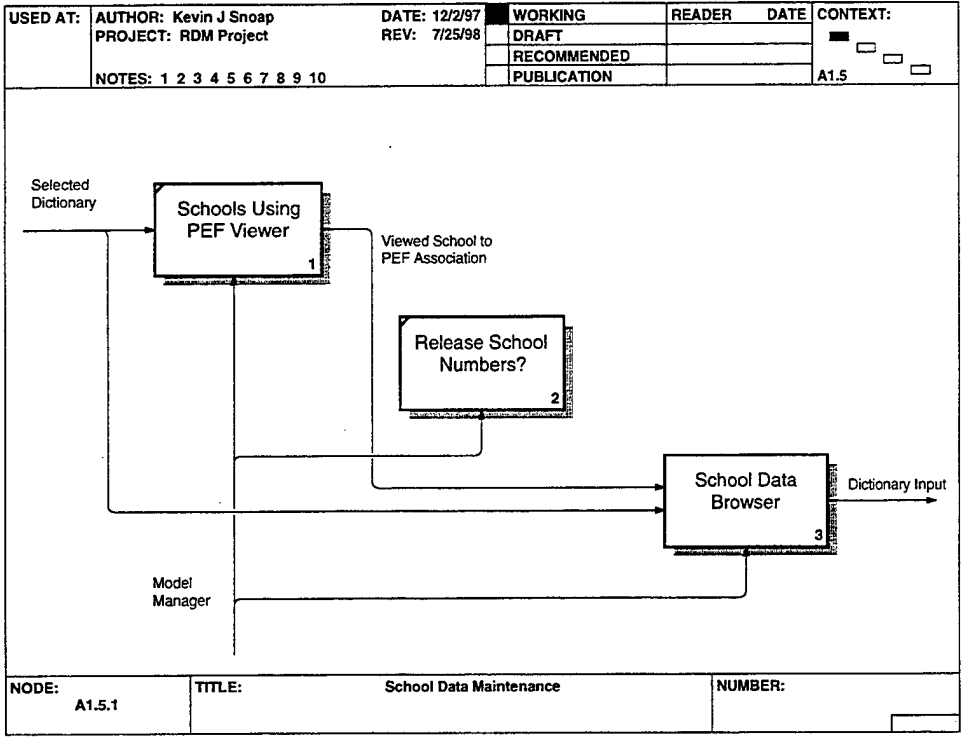


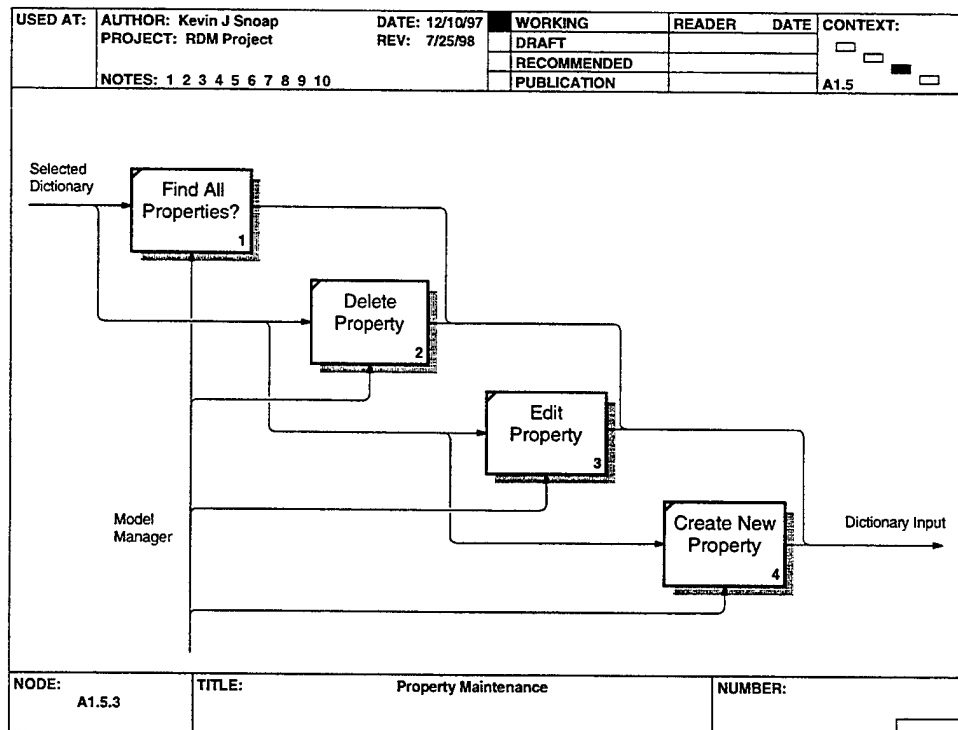
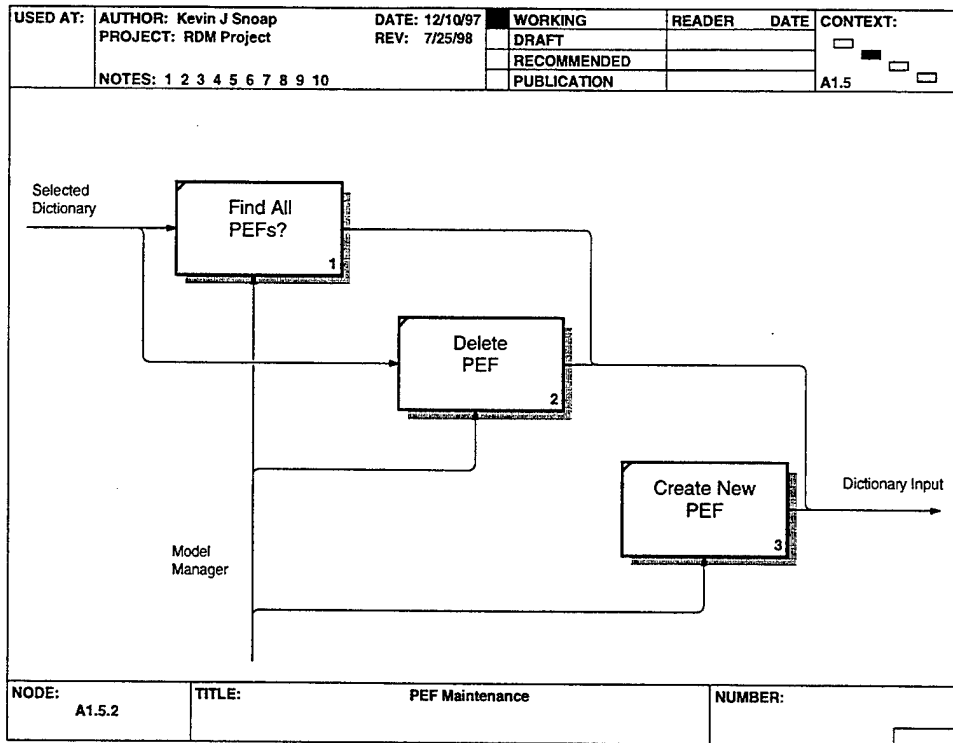


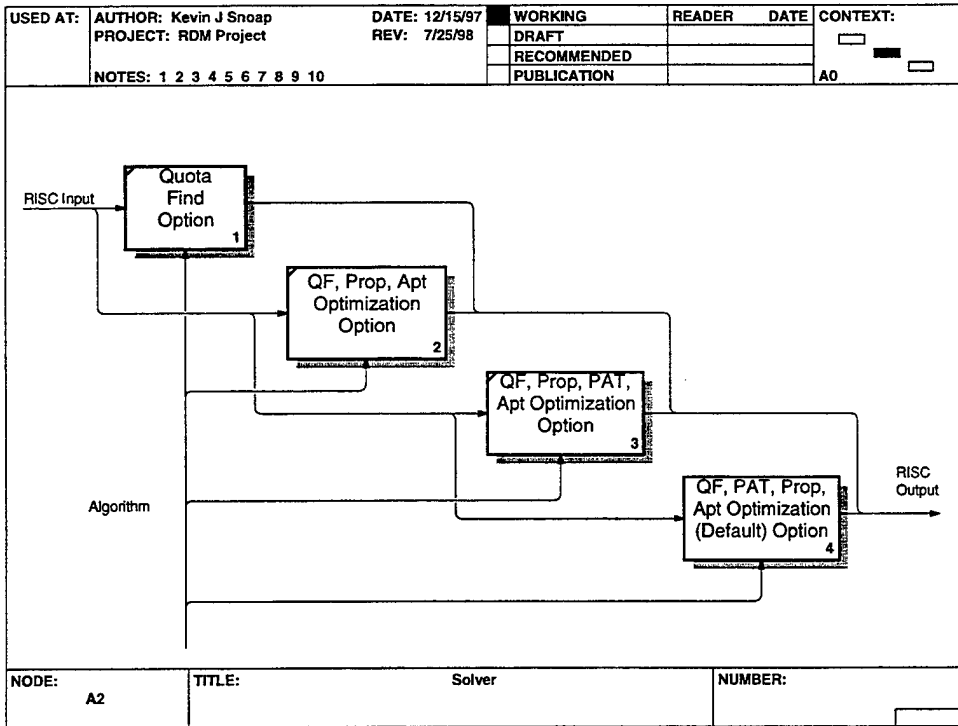
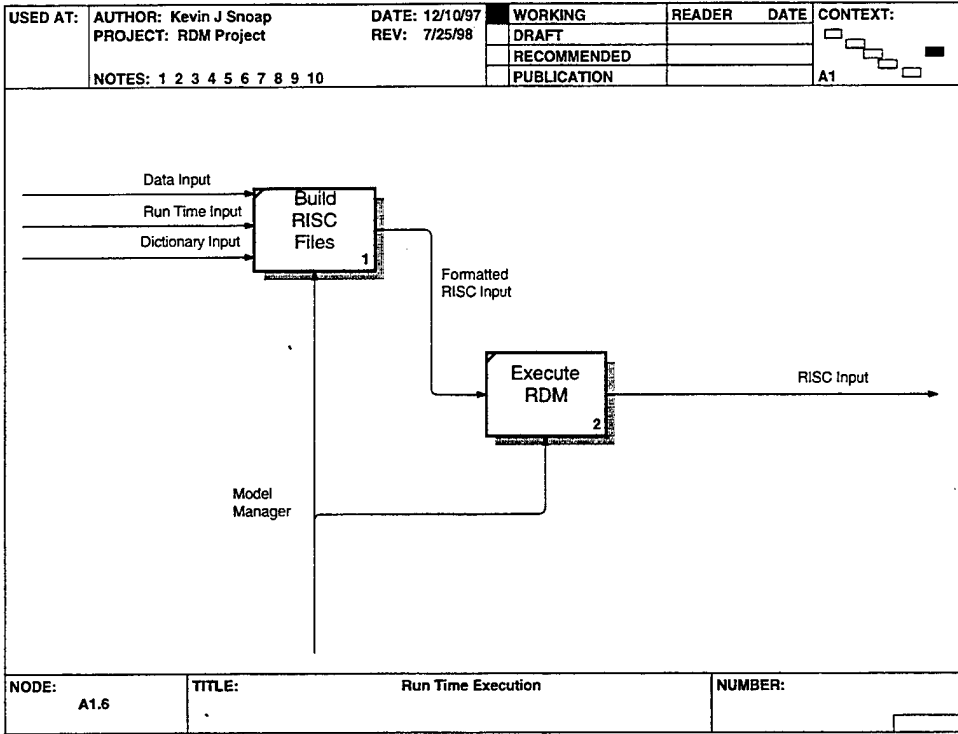


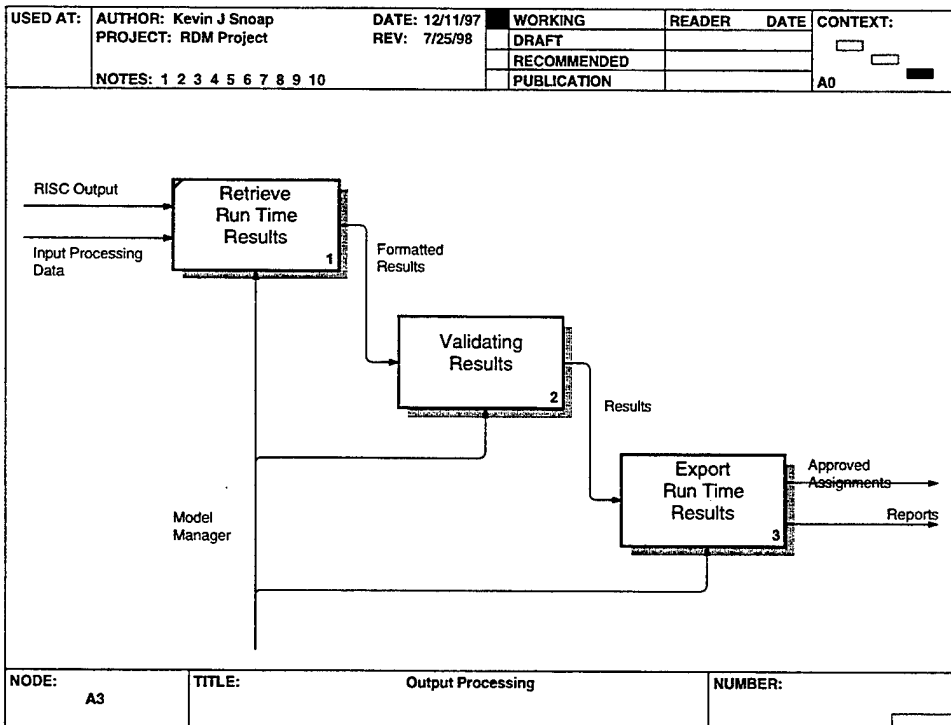
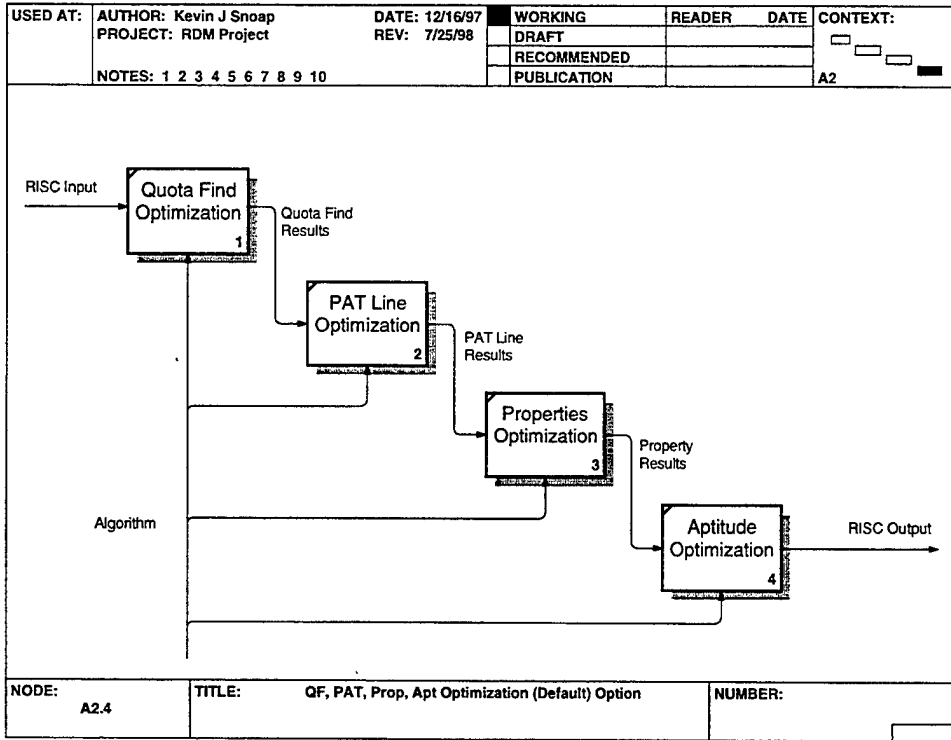


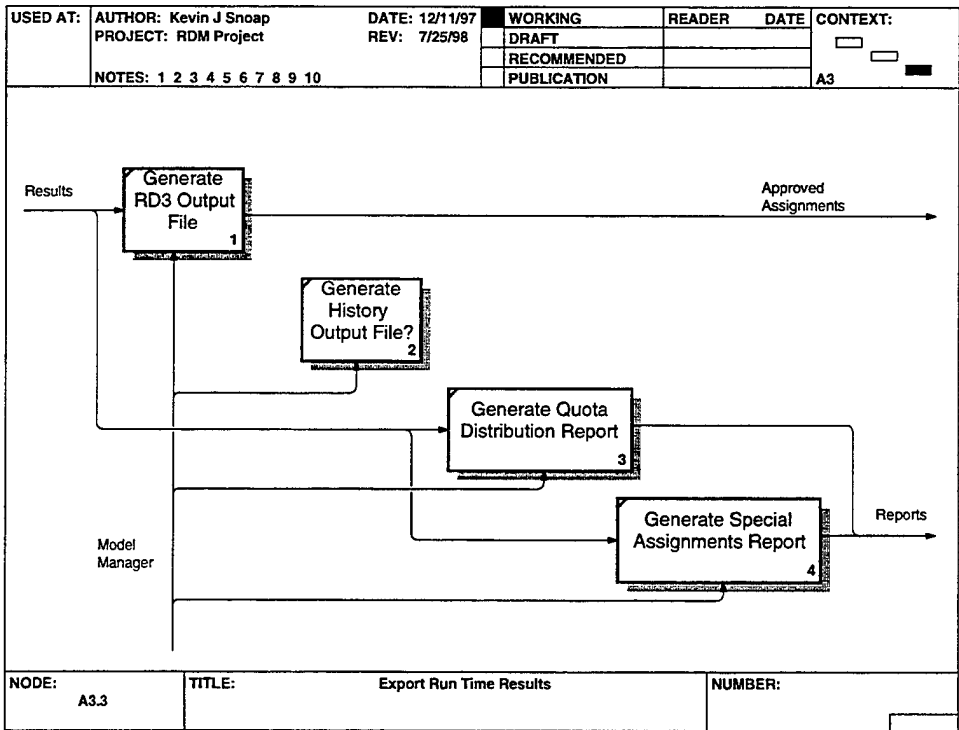
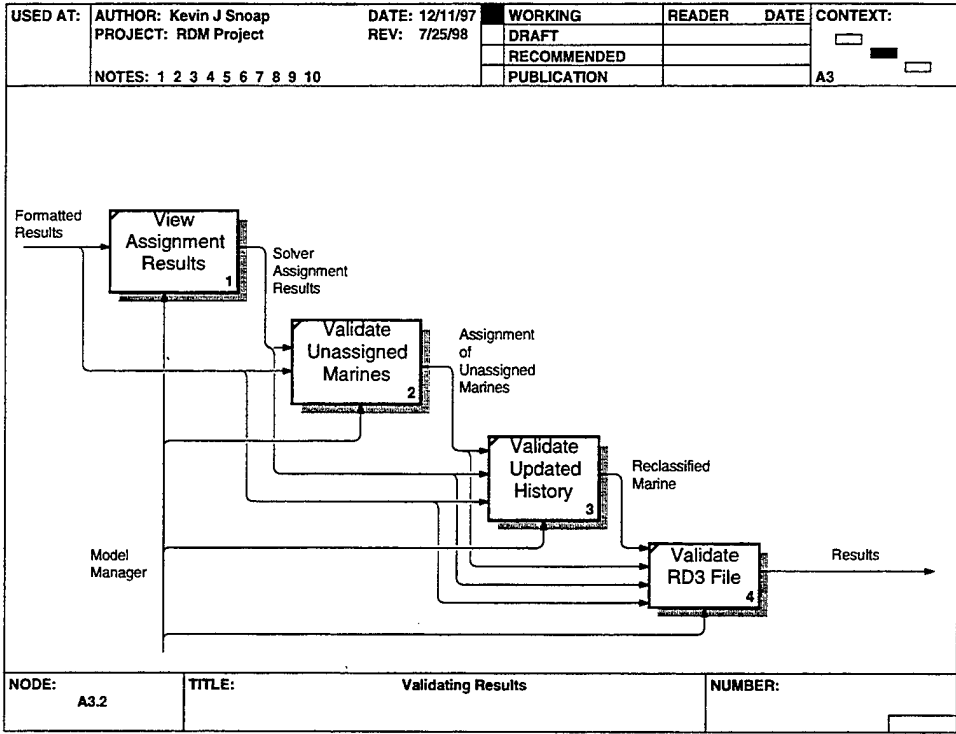




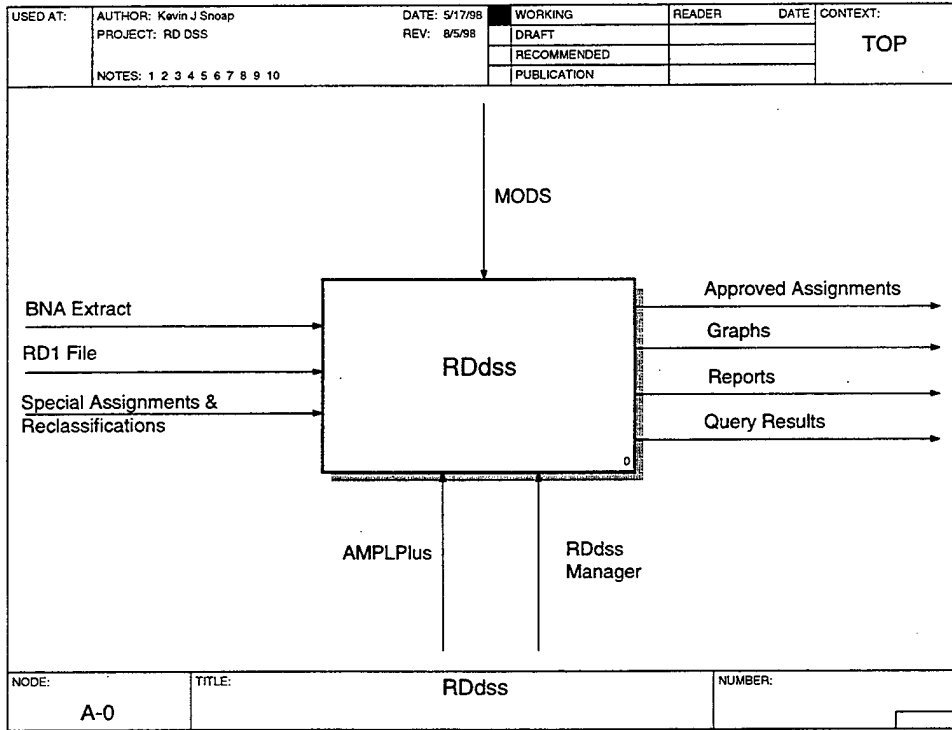


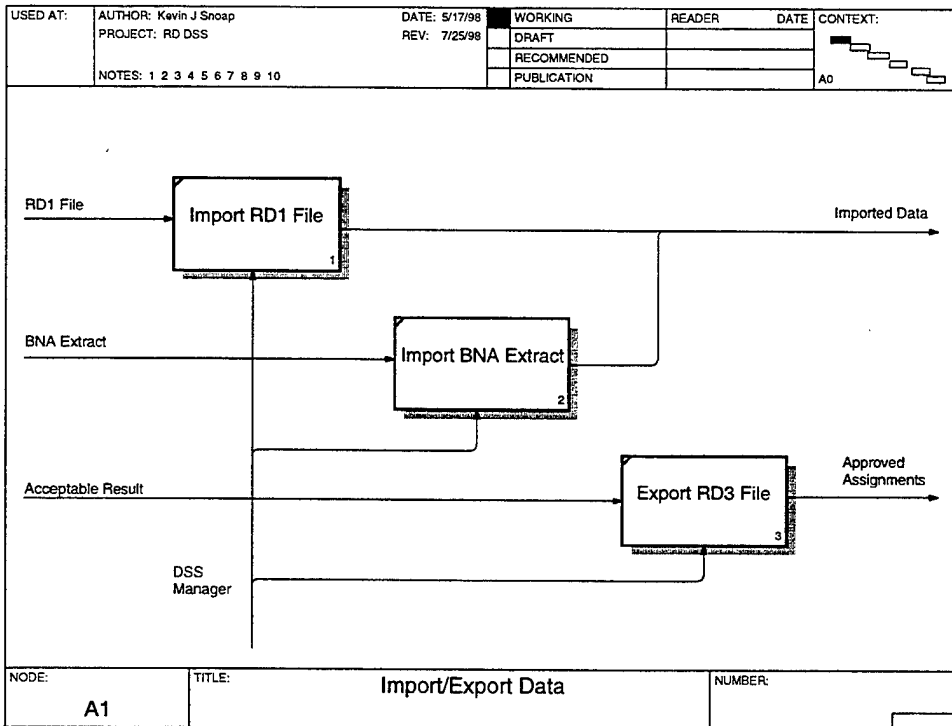
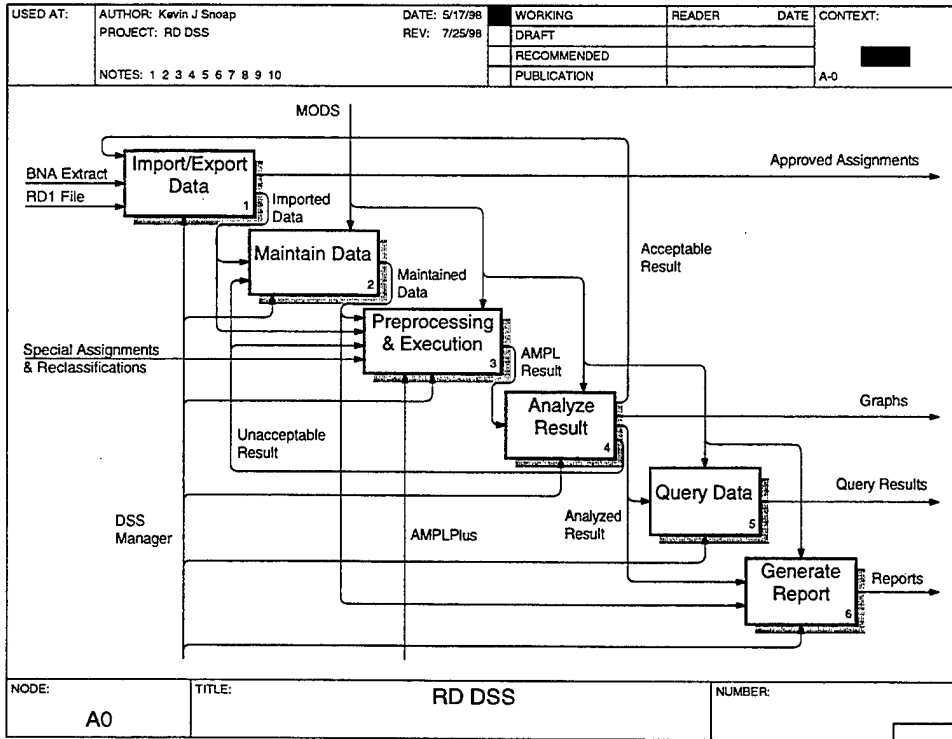


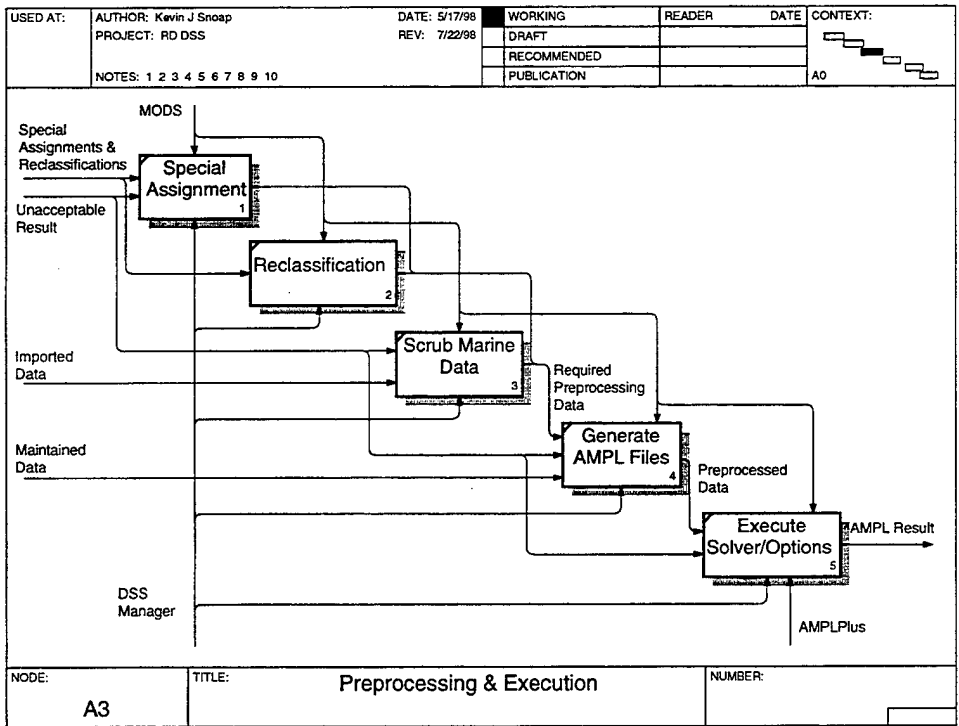
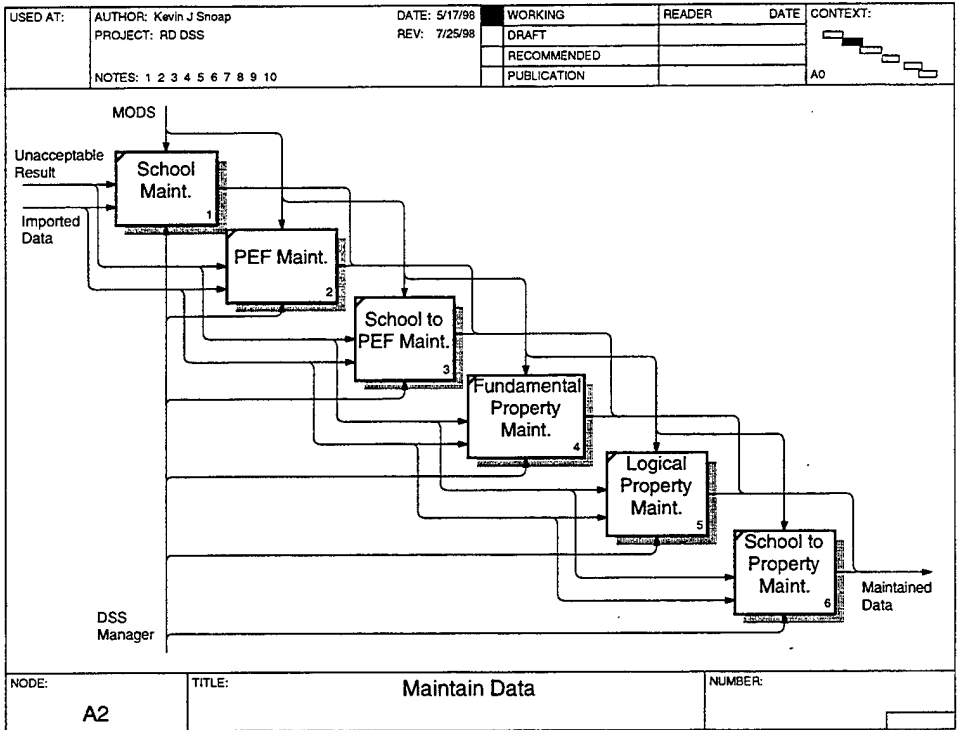


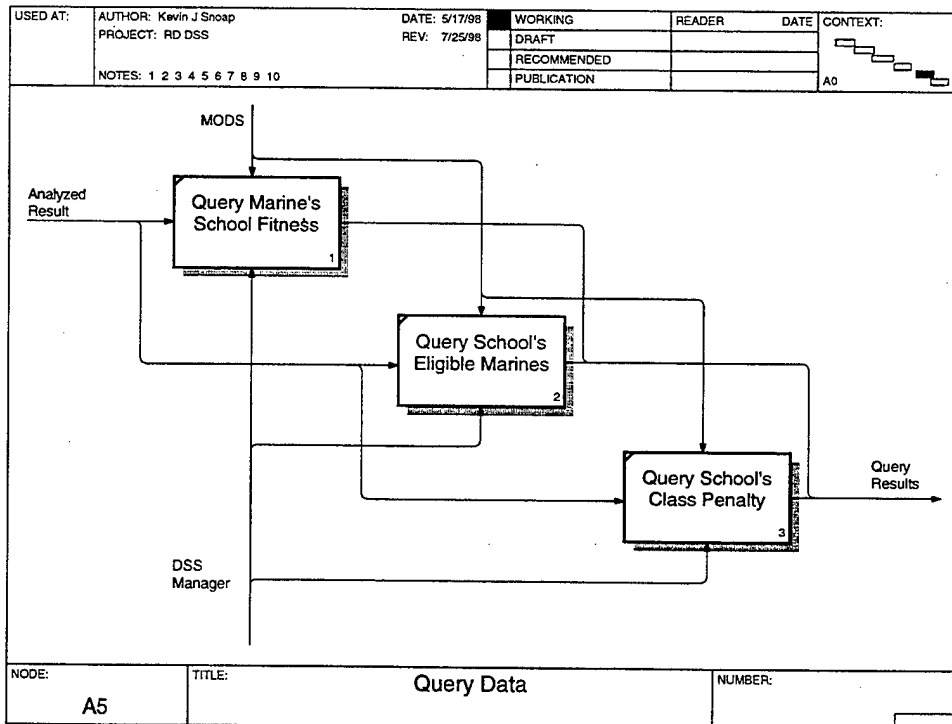
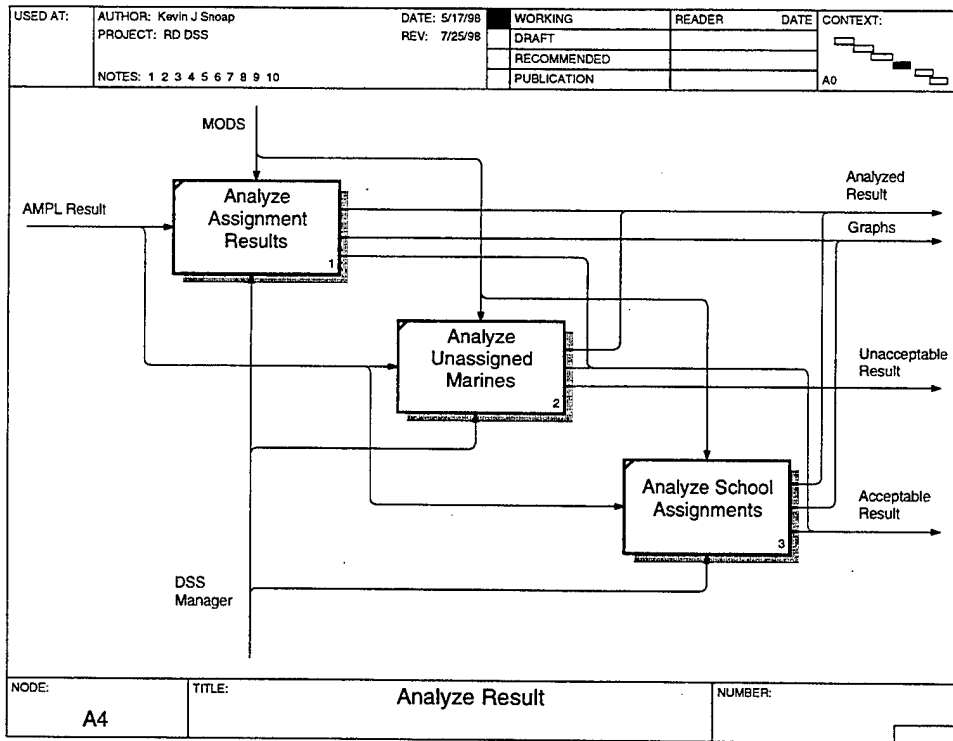


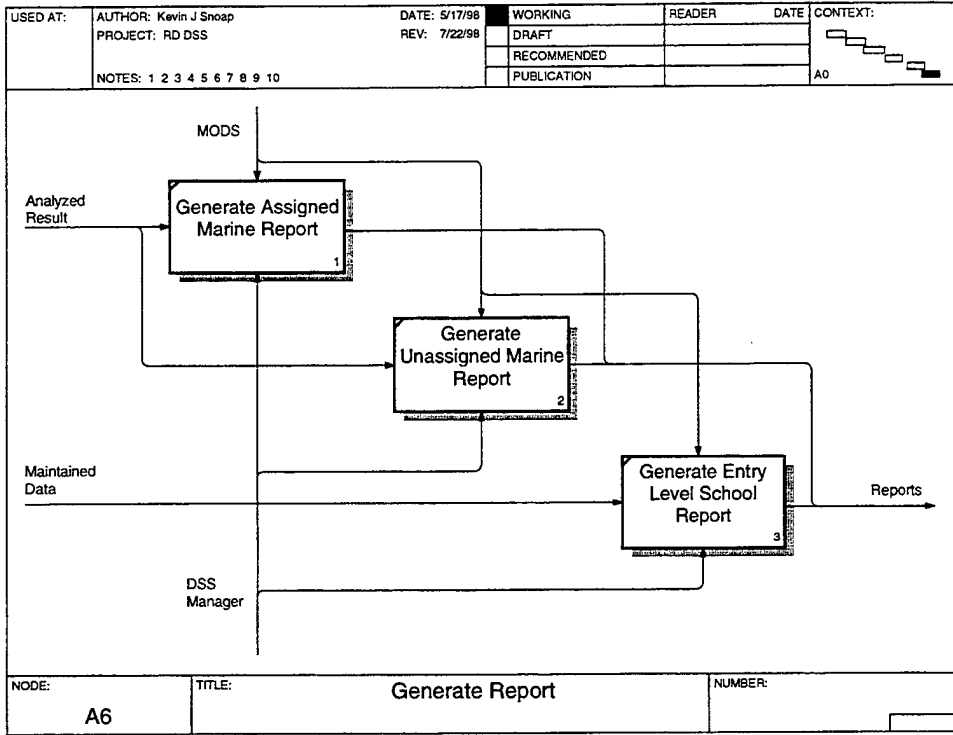
APPENDIX C. TO-BE RD BUSINESS PROCESS (IDEF0) MODEL











APPENDIX D. RDDSS VBA CODE

```

1 Form: frmAnalyzeResult
2 Code
3 1 Attribute VB_Name = "Form_frmAnalyzeResult"
4 2 Attribute VB_Creatable = True
5 3 Attribute VB_PredeclaredId = True
6 4 Attribute VB_Exposed = False
7 5 Option Compare Database
8 6 Option Explicit
9
10 8 Dim booSchoolAssignment As Boolean
11 9
12 10
13 11 Private Sub btnExit_Click()
14 12 On Error GoTo Err_btnExit_Click
15 13
16 14
17 15 DoCmd.Close
18 16
19 17 Exit_btnExit_Click:
20 18 Exit Sub
21 19
22 20 Err_btnExit_Click:
23 21 MsgBox Err.Description
24 22 Resume Exit_btnExit_Click
25 23
26 24
27 25 End Sub
28 26
29 27 Private Sub btnSchoolAssignments_Click()
30 28 On Error GoTo Err_btnSchoolAssignments_Click
31 29
32 30 Dim stDocName As String
33 31 Dim stLinkCriteria As String
34 32
35 33 ' Set the flag
36 34 booSchoolAssignment = True
37 35
38 36 ' Close the current form
39 37 DoCmd.Close
40 38 ' Open specified form
41 39 stDocName = "frmSchoolAssignments"
42 40 DoCmd.OpenForm stDocName, , , stLinkCriteria
43 41
44 42 Exit_btnSchoolAssignments_Click:
45 43 MsgBox Err.Description
46 44 Exit Sub
47 45
48 46 Err_btnSchoolAssignments_Click:
49 47 MsgBox Err.Description
50 48 Resume Exit_btnSchoolAssignments_Click
51 49
52 50 End Sub
53 51
54 52 Private Sub btnSolverAssignments_Click()
55 53 On Error GoTo Err_btnSolverAssignments_Click
56 54
57 55 Dim stDocName As String
58 56 Dim stLinkCriteria As String
59 57
60 58 ' Close the current form
61 59 DoCmd.Close
62 60 ' Open specified form
63 61 stDocName = "frmAssignmentsResult"
64 62 DoCmd.OpenForm stDocName, , , stLinkCriteria
65 63
66 64 Exit_btnSolverAssignments_Click:
67 65 Exit Sub
68 66
69 67 Err_btnSolverAssignments_Click:
70 68 MsgBox Err.Description
71 69 Resume Exit_btnSolverAssignments_Click
72 70
73 71 End Sub
74 72
75 73 Private Sub btnUnassignedMarines_Click()
76 74 On Error GoTo Err_btnUnassignedMarines_Click
77 75
78 76
79 77 Dim strSQL As String
80 78 Dim rec As Recordset
81 79 Dim db1 As Database
82 80
83 81 Set db1 = CurrentDb()
84 82
85 83 ' If the number of unassigned marines = 0, then disable the
86 84 unassigned marines button
87 85 strSQL = "SELECT Count(PEP) AS TotalUnassigned FROM
88 86 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
89 87 If rec.TotalUnassigned = 0 Then
90 88 MsgBox (" All Marines were assigned.")
91 89 Else
92 90 Dim stDocName As String
93 91 Dim stLinkCriteria As String
94 92
95 93 ' Close the current form
96 94 DoCmd.Close
97 95 ' Open specified form
98 96 stDocName = "frmUnassignedMarines"
99 97 DoCmd.OpenForm stDocName, , , stLinkCriteria
100 98
101 99 End If
102 100
103 101
104 102 Exit_btnUnassignedMarines_Click:
105 103 Exit Sub
106 104
107 105 Err_btnUnassignedMarines_Click:
108 106 MsgBox Err.Description
109 107 Resume Exit_btnUnassignedMarines_Click
110 108
111 109 End Sub
112 110
113 111 Private Sub Command8_Click()
114 112 On Error GoTo Err_Command8_Click
115 113
116 114 Dim stDocName As String
117 115 Dim stLinkCriteria As String
118 116
119 117 ' Close the current form
120 118 DoCmd.Close
121 119 ' Open specified form
122 120 stDocName = "frmIDM_Main_Switchboard"
123 121 DoCmd.OpenForm stDocName, , , stLinkCriteria
124 122
125 123 Exit_Command8_Click:
126 124 Exit Sub
127 125
128 126 Err_Command8_Click:
129 127 MsgBox Err.Description
130 128
131 129 Resume Exit_Command8_Click
132 130
133 131 End Sub
134 132
135 133 Private Sub Form_Deactivate()
136 134
137 135 If booSchoolAssignment Then
138 136 Dim strSQL As String
139 137 Dim1 As Integer, j As Integer
140 138 Dim rec As Recordset, rec1 As Recordset, rec2 As Recordset
141 139 Dim db1 As Database
142 140
143 141 Set db1 = CurrentDb()
144 142
145 143 ' Create a table to get the fit and fill weights used during the
146 144 Application.Session "Confirm Action Queries", False
147 145 ' Delete the table if it already exists
148 146 DeleteTable
149 147
150 148 db1.Execute "CREATE TABLE PERCENTAGE (CourseNumber TEXT, AMOS
151 149 TEXT, IntegerAssigned INTEGER, Percentage INTEGER),"
152 150 Application.Session "Confirm Action Queries", True
153 151
154 152 strSQL = "SELECT * FROM PERCENTAGE;"
155 153 Set rec = db1.OpenRecordset(strSQL, dbOpenDynaset)
156 154
157 155 strSQL = "SELECT * FROM qryTotalQuoteForRun ORDER BY
158 156 Set rec2 = db1.OpenRecordset(strSQL, dbOpenSnapshot)
159 157 strSQL = "SELECT * FROM qryTotalQuoteFilesForRun ORDER BY
160 158 SCourseNumber_FK,"
161 159 Set rec3 = db1.OpenRecordset(strSQL, dbOpenSnapshot)
162 160 ' Ensure there are values in the records.
163 161 If rec1.EOF = False Then
164 162 If rec2.EOF = False Then
165 163 rec1.MoveFirst
166 164 rec2.MoveLast
167 165 rec3.MoveFirst
168 166 ' Fill the PENALTY table with the quote values for the
169 167 For i = 1 To rec1.RecordCount
170 168 rec.AddNew
171 169 rec.CourseNumber = rec1.CourseNumber_FK
172 170 rec.AMOS = rec1.AMOS_FK
173 171 rec.Quote = rec1.SumOfQuote
174 172 rec.Update
175 173 rec1.MoveNext
176 174 Next i
177 175 rec.MoveFirst
178 176 ' Update the PENALTY table with the percentage of quote
179 177 For i = 1 To rec2.RecordCount
180 178 rec2.MoveFirst
181 179 For j = 1 To rec3.RecordCount
182 180 If rec3.CourseNumber = rec2.CourseNumber_FK Then
183 181 If rec.AMOS = rec3.AMOS_FK Then
184 182 rec.Edit
185 183 rec.Assigned = rec3.CountOfFacalYear_FK
186 184 rec.Percentage = Int((rec.Assigned /
187 185 rec.Quote) * 100)
188 186 rec.Update
189 187 Exit For
190 188 End If
191 189 End If
192 190 rec2.MoveNext
193 191 ' If no match was found, then the percent fill is
194 192 If i = rec2.RecordCount Then
195 193 rec.Edit
196 194 rec.Assigned = 0
197 195 rec.Percentage = 0
198 196 rec.Update
199 197 End If
200 198 Next i
201 199 Next j
202 200 End If
203 201 End If
204 202
205 203 rec.Close
206 204 rec1.Close
207 205 rec2.Close
208 206 db1.Close
209 207
210 208 ' Reset the flag
211 209 booSchoolAssignment = False
212 210 Else
213 211 DeleteTable
214 212 DeleteTable
215 213 End If
216 214
217 215 End Sub
218 216
219 217 End Sub
220 218
221 219
222 220
223 221
224 222
225 Form: frmAssignmentResult
226 Code
227 1 Attribute VB_Name = "Form_frmAssignmentResult"
228 2 Attribute VB_Creatable = True
229 3 Attribute VB_PredeclaredId = True
230 4 Attribute VB_Exposed = False
231 5 Option Compare Database
232 6 Option Explicit
233 7
234 8
235 9 Private Sub Form_Current()
236 10
237 11 Dim strSQL As String
238 12 Dim rec As Recordset
239 13 Dim db1 As Database
240 14
241 15 Set db1 = CurrentDb()
242 16
243 17 ' Calculate the high, average, and low fitness scores. Also, it
244 18 calculates the number of solver assignments.
245 19 strSQL = "SELECT Avg(Fitness) AS Average, Max(Fitness) AS Maximum,
246 20 Min(Fitness) AS Minimum, Count(Fitness) AS TotalAssigned FROM
247 21 qryAssignmentResult;"
248 22 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
249 23 Me.txtFitnessAvg = rec!Average
250 24 Me.txtFitnessMin = rec!Minimum
251 25 Me.txtFitnessMax = rec!Maximum
252 26 Me.txtSolverAssignments = rec!TotalAssigned
253 27
254 28 ' Calculate the number of manual assignments, and the total number
255 29 of assignments.
256 30 strSQL = "SELECT Count(AssignmentType) AS TotalManualAssigned FROM
257 31 qryManualAssignedMarines;"
258 32 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
259 33 Me.txtManualAssignments = rec!TotalManualAssigned
260 34
261 35
262 36
263 37
264 38
265 39
266 40
267 41
268 42
269 43
270 44
271 45
272 46
273 47
274 48
275 49
276 50
277 51
278 52
279 53
280 54
281 55
282 56
283 57
284 58
285 59
286 60
287 61
288 62
289 63
290 64
291 65
292 66
293 67
294 68
295 69
296 70
297 71
298 72
299 73
300 74
301 75
302 76
303 77
304 78
305 79
306 80
307 81
308 82
309 83
310 84
311 85
312 86
313 87
314 88
315 89
316 90
317 91
318 92
319 93
320 94
321 95
322 96
323 97
324 98
325 99
326 100
327 101
328 102
329 103
330 104
331 105
332 106
333 107
334 108
335 109
336 110
337 111
338 112
339 113
340 114
341 115
342 116
343 117
344 118
345 119
346 120
347 121
348 122
349 123
350 124
351 125
352 126
353 127
354 128
355 129
356 130
357 131
358 132
359 133
360 134
361 135
362 136
363 137
364 138
365 139
366 140
367 141
368 142
369 143
370 144
371 145
372 146
373 147
374 148
375 149
376 150
377 151
378 152
379 153
380 154
381 155
382 156
383 157
384 158
385 159
386 160
387 161
388 162
389 163
390 164
391 165
392 166
393 167
394 168
395 169
396 170
397 171
398 172
399 173
400 174
401 175
402 176
403 177
404 178
405 179
406 180
407 181
408 182
409 183
410 184
411 185
412 186
413 187
414 188
415 189
416 190
417 191
418 192
419 193
420 194
421 195
422 196
423 197
424 198
425 199
426 200
427 201
428 202
429 203
430 204
431 205
432 206
433 207
434 208
435 209
436 210
437 211
438 212
439 213
440 214
441 215
442 216
443 217
444 218
445 219
446 220
447 221
448 222
449 223
450 224
451 225
452 226
453 227
454 228
455 229
456 230
457 231
458 232
459 233
460 234
461 235
462 236
463 237
464 238
465 239
466 240
467 241
468 242
469 243
470 244
471 245
472 246
473 247
474 248
475 249
476 250
477 251
478 252
479 253
480 254
481 255
482 256
483 257
484 258
485 259
486 260
487 261
488 262
489 263
490 264
491 265
492 266
493 267
494 268
495 269
496 270
497 271
498 272
499 273
500 274
501 275
502 276
503 277
504 278
505 279
506 280
507 281
508 282
509 283
510 284
511 285
512 286
513 287
514 288
515 289
516 290
517 291
518 292
519 293
520 294
521 295
522 296
523 297
524 298
525 299
526 300
527 301
528 302
529 303
530 304
531 305
532 306
533 307
534 308
535 309
536 310
537 311
538 312
539 313
540 314
541 315
542 316
543 317
544 318
545 319
546 320
547 321
548 322
549 323
550 324
551 325
552 326
553 327
554 328
555 329
556 330
557 331
558 332
559 333
560 334
561 335
562 336
563 337
564 338
565 339
566 340
567 341
568 342
569 343
570 344
571 345
572 346
573 347
574 348
575 349
576 350
577 351
578 352
579 353
580 354
581 355
582 356
583 357
584 358
585 359
586 360
587 361
588 362
589 363
590 364
591 365
592 366
593 367
594 368
595 369
596 370
597 371
598 372
599 373
600 374
601 375
602 376
603 377
604 378
605 379
606 380
607 381
608 382
609 383
610 384
611 385
612 386
613 387
614 388
615 389
616 390
617 391
618 392
619 393
620 394
621 395
622 396
623 397
624 398
625 399
626 400
627 401
628 402
629 403
630 404
631 405
632 406
633 407
634 408
635 409
636 410
637 411
638 412
639 413
640 414
641 415
642 416
643 417
644 418
645 419
646 420
647 421
648 422
649 423
650 424
651 425
652 426
653 427
654 428
655 429
656 430
657 431
658 432
659 433
660 434
661 435
662 436
663 437
664 438
665 439
666 440
667 441
668 442
669 443
670 444
671 445
672 446
673 447
674 448
675 449
676 450
677 451
678 452
679 453
680 454
681 455
682 456
683 457
684 458
685 459
686 460
687 461
688 462
689 463
690 464
691 465
692 466
693 467
694 468
695 469
696 470
697 471
698 472
699 473
700 474
701 475
702 476
703 477
704 478
705 479
706 480
707 481
708 482
709 483
710 484
711 485
712 486
713 487
714 488
715 489
716 490
717 491
718 492
719 493
720 494
721 495
722 496
723 497
724 498
725 499
726 500
727 501
728 502
729 503
730 504
731 505
732 506
733 507
734 508
735 509
736 510
737 511
738 512
739 513
740 514
741 515
742 516
743 517
744 518
745 519
746 520
747 521
748 522
749 523
750 524
751 525
752 526
753 527
754 528
755 529
756 530
757 531
758 532
759 533
760 534
761 535
762 536
763 537
764 538
765 539
766 540
767 541
768 542
769 543
770 544
771 545
772 546
773 547
774 548
775 549
776 550
777 551
778 552
779 553
780 554
781 555
782 556
783 557
784 558
785 559
786 560
787 561
788 562
789 563
790 564
791 565
792 566
793 567
794 568
795 569
796 570
797 571
798 572
799 573
800 574
801 575
802 576
803 577
804 578
805 579
806 580
807 581
808 582
809 583
810 584
811 585
812 586
813 587
814 588
815 589
816 590
817 591
818 592
819 593
820 594
821 595
822 596
823 597
824 598
825 599
826 600
827 601
828 602
829 603
830 604
831 605
832 606
833 607
834 608
835 609
836 610
837 611
838 612
839 613
840 614
841 615
842 616
843 617
844 618
845 619
846 620
847 621
848 622
849 623
850 624
851 625
852 626
853 627
854 628
855 629
856 630
857 631
858 632
859 633
860 634
861 635
862 636
863 637
864 638
865 639
866 640
867 641
868 642
869 643
870 644
871 645
872 646
873 647
874 648
875 649
876 650
877 651
878 652
879 653
880 654
881 655
882 656
883 657
884 658
885 659
886 660
887 661
888 662
889 663
890 664
891 665
892 666
893 667
894 668
895 669
896 670
897 671
898 672
899 673
900 674
901 675
902 676
903 677
904 678
905 679
906 680
907 681
908 682
909 683
910 684
911 685
912 686
913 687
914 688
915 689
916 690
917 691
918 692
919 693
920 694
921 695
922 696
923 697
924 698
925 699
926 700
927 701
928 702
929 703
930 704
931 705
932 706
933 707
934 708
935 709
936 710
937 711
938 712
939 713
940 714
941 715
942 716
943 717
944 718
945 719
946 720
947 721
948 722
949 723
950 724
951 725
952 726
953 727
954 728
955 729
956 730
957 731
958 732
959 733
960 734
961 735
962 736
963 737
964 738
965 739
966 740
967 741
968 742
969 743
970 744
971 745
972 746
973 747
974 748
975 749
976 750
977 751
978 752
979 753
980 754
981 755
982 756
983 757
984 758
985 759
986 760
987 761
988 762
989 763
990 764
991 765
992 766
993 767
994 768
995 769
996 770
997 771
998 772
999 773
1000 774

```

```

261 Me.tblSolveAssignments
262 30
263 31 ' Calculates the number of unassigned marines, and the total number
264 of assignable marines.
265 32 strSQL = "SELECT Count(PER) AS TotalUnassigned FROM
266 33 Sel rec = db1.OpenRecords(strSQL, dbOpenSnapshot)
267 34 Me.tblTotalUnassigned = rec.TotalUnassigned
268 35 Me.tblTotalAssignable = rec.TotalUnassigned +
269 36
270 37 ' Create a table to get the fit and fill weights used during the run
271 38 Application.SetOption "Confirm Action Queries", False
272 39 db1.Execute "CREATE TABLE CONSTANTS (ConstantName TEXT, IntValue
273 40 DoCmd.TransferText acImportField, "Form_Data Export Specification",
274 41 CONSTANTS, "C:\tbl\Output\Temp\ConstsForm.dat"
275 42 Application.SetOption "Confirm Action Queries", True
276 43
277 43 strSQL = "SELECT IntValue FROM CONSTANTS;"
278 44 Sel rec = db1.OpenRecords(strSQL, dbOpenSnapshot)
279 45 Me.tblFitWeight = rec.IntValue
280 46 rec.MoveNext
281 47 rec.MoveNext
282 48 Me.tblFillWeight = rec.IntValue
283 49 rec.Close
284 50
285 51 ' Remove the created table
286 52 db1.Execute "DROP TABLE CONSTANTS;"
287 53 db1.Close
288 54
289 55
290 56 End Sub
291 57 Private Sub Close_Click()
292 58 On Error GoTo Err_Close_Click
293 59
294 60 Dim stDocName As String
295 61 Dim stLinkCriteria As String
296 62
297 63 ' Close the current form
298 64 DoCmd.Close
299 65 ' Open specified form
300 66 stDocName = "frmAnalysisResult"
301 67 DoCmd.OpenForm stDocName, , , stLinkCriteria
302 68
303 69 Exit_Close_Click
304 70 Exit Sub
305 71
306 72 Err_Close_Click
307 73 MsgBox Err.Description
308 74 Resume Exit_Close_Click
309 75
310 76 End Sub
311
312 Form: frmChangeGradDate
313 Code
314 1 Attribute VB_Name = "Form_frmChangeGradDate"
315 2 Attribute VB_Creatable = True
316 3 Attribute VB_PredeclaredId = True
317 4 Attribute VB_Exposed = False
318 5 Option Compare Database
319 6 Option Explicit
320 7
321 8 Private Sub btnChangeDate_Click()
322 9
323 10 ' Change the specified graduation date in the MARINE table
324 11 Application.SetOption "Confirm Action Queries", False
325 12 DoCmd.OpenQuery "qryChangeGradDate"
326 13 Application.SetOption "Confirm Action Queries", True
327 14
328 15 ' Update the drop down list for MCRD Grad Date
329 16 Me.cmbMCRDGradDate.RowSource = "SELECT DISTINCTROW
330 17 [qryMCTGradDate].[GradDate] FROM [qryMCTGradDate];"
331 17
332 18 ' Force user to refresh data before performing another date change
333 19 Me.txtMCRDGradDate.Enabled = False
334 20 Me.tblMCTGradDate.Enabled = False
335 21 Me.tblMCTGradDate.Enabled = False
336 22
337 23 ' Provide a message indicating the change was made
338 24 MsgBox "Date change was successful!"
339 25
340 26 End Sub
341 27
342 28 Private Sub btnEquals_Click()
343 29
344 30 Dim varDate As Variant
345 31 Dim inDays As Integer
346 32
347 33 inDays = Me.cmbDays
348 34 varDate = Me.cmbMCRDGradDate
349 35
350 36 If IsNull(varDate) = False And IsNull(inDays) = False Then
351 37
352 38 ' Format this date for use in the DateAdd function
353 39 varDate = Format(varDate, "####/##/##")
354 40
355 41 ' Add 120 days to this date
356 42 varDate = DateAdd("d", inDays, varDate)
357 43
358 44 ' Change calendar to this date
359 45 Me.txtCalendar.Value = varDate
360 46
361 47 ' Convert the new date back into the original format
362 48 varDate = Format(varDate, "yyyymmdd")
363 49
364 50 ' Place the new date in the "upper date bound" text box on the
365 51 calling form.
366 51 Me.tblMCTGradDate = varDate
367 52
368 53 ' Allows user to change the graduation date in the MARINE table
369 54 Me.tblMCTGradDate.Enabled = True
370 55 End If
371 56
372 57 End Sub
373 58
374 59 Private Sub cmbMCRDGradDate_AfterUpdate()
375 60
376 61 Dim varDate As Variant
377 62
378 63 varDate = Me.cmbMCRDGradDate
379 64 Me.tblMCTGradDate.Value = ""
380 65
381 66 If IsNull(varDate) = False Then
382 67
383 68 ' Format this date for use in the DateAdd function
384 69 varDate = Format(varDate, "####/##/##")
385 70
386 71 ' Change calendar to this date
387 72 Me.txtCalendar.Value = varDate
388 73 Me.txtCalendar.Visible = True
389 74 Me.cmbDays.Enabled = True
390 75 Me.tblMCTGradDate.Enabled = True

```

```

391 76 Me.tblMCTGradDate.Enabled = True
392 77
393 78 End If
394 79
395 80 End Sub
396 81
397 82 Private Sub Form_Current()
398 83
399 84 Me.txtCalendar.Visible = False
400 85 Me.cmbDays.Enabled = False
401 86 Me.tblMCTGradDate.Enabled = False
402 87 Me.tblMCTGradDate.Enabled = False
403 88 Me.tblMCTGradDate.Enabled = False
404 89
405 90 End Sub
406 91 Private Sub Close_Click()
407 92 On Error GoTo Err_Close_Click
408 93
409 94 Dim stDocName As String
410 95 Dim stLinkCriteria As String
411 96
412 97 ' Close the current form
413 98 DoCmd.Close
414 99 ' Open specified form
415 100 stDocName = "frmImportExportSwitchboard"
416 101 DoCmd.OpenForm stDocName, , , stLinkCriteria
417 102
418 103 Exit_Close_Click
419 104 Exit Sub
420 105
421 106 Err_Close_Click
422 107 MsgBox Err.Description
423 108 Resume Exit_Close_Click
424 109
425 110 End Sub
426
427 Form: frmClass
428 Code
429 1 Attribute VB_Name = "Form_frmClass"
430 2 Attribute VB_Creatable = True
431 3 Attribute VB_PredeclaredId = True
432 4 Attribute VB_Exposed = False
433 5 Option Compare Database
434 6 Option Explicit
435 7
436 8 Private Sub btnFindSchool_Click()
437 9 On Error GoTo Err_btnFindSchool_Click
438 10
439 11
440 12 Screen.PreviousControl.SetFocus
441 13 DoCmd.DoMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
442 14
443 15 Exit_btnFindSchool_Click
444 16 Exit Sub
445 17
446 18 Err_btnFindSchool_Click
447 19 MsgBox Err.Description
448 20 Resume Exit_btnFindSchool_Click
449 21
450 22 End Sub
451
452 Form: frmClassQuotaPenaltyAndFit
453 Code
454 1 Attribute VB_Name = "Form_frmClassQuotaPenaltyAndFit"
455 2 Attribute VB_Creatable = True
456 3 Attribute VB_PredeclaredId = True
457 4 Attribute VB_Exposed = False
458 5 Option Compare Database
459 6 Option Explicit
460 7
461 8 Private Sub btnGo_Click()
462 9
463 10 Dim varUDateBound
464 11 Dim lngHour As Long, lngMin As Long, lngSec As Long
465 12 Dim lngStart As Long, lngEnd As Long
466 13 Dim strSQL As String
467 14
468 15 ' Starts the timer for displaying the time taken for this subroutine
469 16 lngStart = Timer
470 17
471 18
472 19 ' Prevents the user from going back to change the MCTGradDate until a
473 20 fitness file is
474 20 ' made, or the reset button is pushed.
475 21 Me.tblMCTGradDate.Enabled = False
476 22 varUDateBound = Me.UDateBound
477 23
478 24 ' Generate the quota and penalty files
479 25 Quota_Penalty (varUDateBound)
480 26
481 27 ' Update the quota/penalty records on this form
482 28 strSQL = "SELECT DISTINCTROW ([CLASS].[CourseNumber_FK] &
483 29 '[CLASS].[AMICS_FK].[CLASS].[ClassNumber_FK].[CLASS].[ReportDate] &
484 30 '[CLASS].[Quota].[CLASS].[ClassIndex].[PENALTY].[ClassIndex_FK] &
485 31 '[PENALTY].[Penalty] FROM ([CLASS] INNER JOIN [PENALTY] ON " &
486 32 "[CLASS].[ClassIndex] = [PENALTY].[ClassIndex_FK]);"
487 33 Me.RecordSource = strSQL
488 34
489 35 ' update status of status text box
490 36 Me.txtStatus = "Working..."
491 37
492 38 ' Call the Fitness procedure
493 39 Fitness
494 40
495 41 ' update status of status text box
496 42 Me.txtStatus = "Files generated!"
497 43 DoCmd.Beep
498 44
499 45 ' Records the stop time for running this subroutine
500 46 lngEnd = Timer
501 47 ' Determine total secs
502 48 lngSec = lngEnd - lngStart
503 49 ' Determine total hours
504 50 lngHour = lngSec \ 3600
505 51 ' Determine total mins
506 52 lngMin = (lngSec - lngHour * 3600) \ 60
507 53 ' Determine the secs remaining
508 54 lngSec = (lngSec - (lngMin * 60) - (lngHour * 3600)) \ 1
509 55 MsgBox "This operation took " & lngHour & " hour(s), " & lngMin & "
510 56 mIn(s), " &
511 57 " and " & lngSec & " secs.", vbOKOnly, "Recruit Distribution Model"
512 57
513 58 Me.btnPrepareAndExecuteSolver.SetFocus
514 59
515 60 End Sub
516 61 End Sub
517 62
518 63 Private Sub btnGo_MouseDown(Button As Integer, Shift As Integer, X As
519 64 Y As Single)
520 65

```

```

521 65 Me!Status = "Working..."
522 67
523 68 End Sub
524 69
525 70 Private Sub btnPrepareAndExecuteSolve_Click()
526 71 On Error GoTo Err_btnPrepareAndExecuteSolve_Click
527 72
528 73 Dim stDocName As String
529 74 Dim stLinkCriteria As String
530 75
531 76 ' Close the current form
532 77 DoCmd.Close
533 78 ' Open specified form
534 79 stDocName = "frmPrepareAndExecuteSolve"
535 80 DoCmd.OpenForm stDocName, , stLinkCriteria
536 81
537 82 Exit_btnPrepareAndExecuteSolve_Click:
538 83 Exit Sub
539 84
540 85 Err_btnPrepareAndExecuteSolve_Click:
541 86 MsgBox Err.Description
542 87 Resume Exit_btnPrepareAndExecuteSolve_Click
543 88
544 89 End Sub
545 90
546 91 Private Sub btnReturnPrevious_Click()
547 92 On Error GoTo Err_btnReturnPrevious_Click
548 93
549 94 Dim stDocName As String
550 95 Dim stLinkCriteria As String
551 96
552 97 ' Close the current form
553 98 DoCmd.Close
554 99 ' Open specified form
555 100 stDocName = "frmProcessingAndExecutionSwitchboard"
556 101 DoCmd.OpenForm stDocName, , stLinkCriteria
557 102
558 103 Exit_btnReturnPrevious_Click:
559 104 Exit Sub
560 105
561 106 Err_btnReturnPrevious_Click:
562 107 MsgBox Err.Description
563 108 Resume Exit_btnReturnPrevious_Click
564 109
565 110 End Sub
566 111
567 112 Private Sub cmbCourseNumberFind_AfterUpdate()
568 113
569 114 Dim R As Recordset
570 115 Set R = Me.RecordsetClone
571 116 R.FindFirst "[SCourseNumber_PK] = " & Chr(34) &
572 117 Me[cmbCourseNumberFind] & Chr(34)
573 118 Me[cmbCourseNumberFind] = Null
574 119 Me.Penalty.SetFocus
575 120
576 121 End Sub
577 122
578 123 Private Sub Form_Current()
579 124
580 125 ' Prevents the user from creating the AMPL files until a MCT
581 126 ' graduation date is entered
582 127 If IsNull(Me.MCTGradDate.Value) Then
583 128 Me!btnGo.Enabled = False
584 129 Me!Status.Enabled = False
585 130
586 131 ' Allows the user to create the quota and penalty files
587 132 Else
588 133 Me!btnGo.Enabled = True
589 134
590 135 End If
591 136
592 137 ' Update the status of the status text box
593 138 Me!Status = ""
594 139
595 140 End Sub
596 141 Private Sub MCTGradDate_AfterUpdate()
597 142
598 143 Dim varDate As Variant
599 144
600 145 varDate = Me.MCTGradDate
601 146
602 147 If IsNull(varDate) = False Then
603 148
604 149 Me!btnGo.Enabled = True
605 150 Me!Status.Enabled = True
606 151
607 152 ' Format this date for use in the DateAdd function
608 153 varDate = Format(varDate, "###/##/##")
609 154
610 155 ' Add 120 days to this date
611 156 varDate = DateAdd("d", 120, varDate)
612 157
613 158 ' Convert the new date back into the original format
614 159 varDate = Format(varDate, "yyyy/mm/dd")
615 160
616 161 ' Place the new date in the "upper date bound" text box on the
617 162 ' calling form.
618 163 Me.UDateBound = varDate
619 164
620 165 Else
621 166 Me!btnGo.Enabled = False
622 167 Me!Status.Enabled = False
623 168 End If
624 169
625 170 End Sub
626 171
627 172
628 173
629 Form: frmFind
630 Code
631 1 Attribute VB_Name = "Form_frmFind"
632 2 Attribute VB_Creatable = True
633 3 Attribute VB_PredeclaredId = True
634 4 Attribute VB_Exposed = False
635 5 Option Compare Database
636 6 Option Explicit
637 7
638 8 Private Sub btnFind_Click()
639 9 On Error GoTo Err_btnFind_Click
640 10
641 11 ' Screen PreviousControl.SetFocus
642 12 Form!frmSchoolToPropertyMenu.SetFocus
643 13 DoCmd.FindRecord Me.cmbCourseNumber, , True, , True, acAll
644 14 DoCmd.FindMenuItem acFormBar, acEditMenu, 10, , acMenuVer70
645 15
646 16 Exit_btnFind_Click:
647 17 Exit Sub
648 18
649 19 Err_btnFind_Click:
650 20 MsgBox Err.Description

```

```

651 21 Resume Exit_btnFind_Click
652 22 End Sub
653 23
654 24
655 Form: frmFundamentalProperty
656 Code
657 1 Attribute VB_Name = "Form_frmFundamentalProperty"
658 2 Attribute VB_Creatable = True
659 3 Attribute VB_PredeclaredId = True
660 4 Attribute VB_Exposed = False
661 5 Option Compare Database
662 6 Option Explicit
663 7
664 8 Private Sub Command15_Click()
665 9
666 10 MsgBox "This is a button, no information, 'Testing'"
667 11
668 12 End Sub
669 13
670 14 Private Sub btnClose_Click()
671 15 On Error GoTo Err_btnClose_Click
672 16
673 17 DoCmd.Close
674 18 DoCmd.OpenForm "frmMaintenanceSwitchboard"
675 19
676 20 Exit Sub
677 21
678 22 Exit_btnClose_Click:
679 23 Exit Sub
680 24
681 25 Err_btnClose_Click:
682 26 MsgBox Err.Description
683 27 Resume Exit_btnClose_Click
684 28
685 29 End Sub
686 30
687 31 Private Sub cmbPropertyNameFind_AfterUpdate()
688 32
689 33 Dim R As Recordset
690 34 Set R = Me.RecordsetClone
691 35 R.FindFirst "[FPropertyName_PK] = " & Chr(34) &
692 36 Me[cmbPropertyNameFind] & Chr(34)
693 37 Me.Bookmark = R.Bookmark
694 38 Me[cmbPropertyNameFind] = Null
695 39 Me.FPropertyName_PK.SetFocus
696 40
697 41 End Sub
698 42
699 43 Private Sub Command11_Click()
700 44 ' Forces user to enter date for property name and description
701 45 If IsNull(Me.FPropertyName_PK.Value) Or IsNull(Me.Description.Value)
702 46 Me!MarineField.Enabled = False
703 47 Me!Operator.Enabled = False
704 48 Me!Label10.Visible = False
705 49 Me!subfrmFundamentalPropertyList.Visible = False
706 50 Me!subfrmFundamentalPropertyList.Visible = False
707 51 Me!cmbPotentialValue.Visible = False
708 52 Me!cmbPotentialValue.Visible = False
709 53 MsgBox "You must provide a 'Property Name' and 'Description'"
710 54 ' before building the fundamental equation."
711 55
712 56 ' Allows the user to enter values for the fundamental equation
713 57 Else
714 58 Me!MarineField.Enabled = True
715 59 Me!Operator.Enabled = True
716 60 Me!Label10.Visible = True
717 61 Me!Form!frmFundamentalProperty.Operator.Value = "in" Then
718 62 Me!subfrmFundamentalPropertyList.Visible = False
719 63 Me!subfrmFundamentalPropertyList.Visible = True
720 64
721 65 Else
722 66 Me!cmbPotentialValue.Visible = True
723 67 Me!subfrmFundamentalPropertyList.Visible = True
724 68 Me!subfrmFundamentalPropertyList.Visible = False
725 69 End If
726 70 End If
727 71
728 72 End Sub
729 73 Private Sub Description_AfterUpdate()
730 74
731 75 Dim strSQL As String
732 76
733 77 ' Forces user to enter date for property name and description
734 78 If IsNull(Me.FPropertyName_PK.Value) Or IsNull(Me.Description.Value)
735 79 Me!MarineField.Enabled = False
736 80 Me!Operator.Enabled = False
737 81 Me!Label10.Visible = False
738 82 Me!subfrmFundamentalPropertyList.Visible = False
739 83 Me!subfrmFundamentalPropertyList.Visible = False
740 84
741 85 ' Allows the user to enter values for the fundamental equation
742 86 Else
743 87 Me!MarineField.Enabled = True
744 88 Me!Operator.Enabled = True
745 89 Me!Label10.Visible = True
746 90 Me!Form!frmFundamentalProperty.Operator.Value = "in" Then
747 91 strSQL = "SELECT DISTINCT " & Me!MarineField & " FROM
748 92 Form!frmFundamentalProperty!subfrmFundamentalPropertyList.Value_PK.RowSo
749 93 urce = strSQL
750 94 Me!subfrmFundamentalPropertyList.Visible = False
751 95 Me!subfrmFundamentalPropertyList.Visible = True
752 96
753 97 Else
754 98 strSQL = "SELECT DISTINCT " & Me!MarineField & " FROM
755 99 Form!frmFundamentalProperty!subfrmFundamentalProperty!cmbValue_PK.RowSou
756 100 rce = strSQL
757 101 Me!subfrmFundamentalPropertyList.Visible = True
758 102 Me!subfrmFundamentalPropertyList.Visible = False
759 103 End If
760 104
761 105 End Sub
762 106
763 107 Private Sub Form_BeforeUpdate(Cancel As Integer)
764 108
765 109 [FundPropTimeStamp] = Now
766 110
767 111 End Sub
768 112
769 113 Private Sub Form_Current()
770 114 Dim strSQL As String
771 115
772 116 strSQL = "SELECT MarineField, Operator FROM FUNDAMENTAL_PROPERTY
773 117 WHERE FPropertyName_PK = " & FundPropName
774 118
775 119 End Sub
776 120
777 121
778 122
779 123
780 124

```

```

781 117 ' Forces user to enter data for property name and description
782 118 If IsNull(Me.FPropertyName_PK.Value) Or IsNull(Me.Description.Value)
783 119 Me.MarineField.Enabled = False
784 120 Me.Operator.Enabled = False
785 121 Me.LabID.Visible = False
786 122 Me.SubfrmFundamentalsProperty.Visible = False
787 123 Me.SubfrmFundamentalsPropertyList.Visible = False
788 124
789 125 ' Allows the user to enter values for the fundamental equation
790 126 Else
791 127 Me.MarineField.Enabled = True
792 128 Me.Operator.Enabled = True
793 129 Me.LabID.Visible = True
794 130 If Forms!frmFundamentalsProperty.Operator.Value = "In" Then
795 131 ' Update the Potential Value List
796 132 Me.SubfrmFundamentalsProperty.Visible = False
797 133 Me.SubfrmFundamentalsPropertyList.Visible = True
798 134 strSQL = "SELECT DISTINCT * & Me.MarineField & * FROM
799 135
800 Forms!frmFundamentalsProperty!SubfrmFundamentalsPropertyList!Value_FK.RowSo
801 urce = strSQL
802 136 Else
803 137 ' Update the Potential Value List
804 138 strSQL = "SELECT DISTINCT * & Me.MarineField & * FROM
805 139
806 Forms!frmFundamentalsProperty!SubfrmFundamentalsProperty!cmbValue_FK.RowSou
807 rce = strSQL
808 140 Me.SubfrmFundamentalsProperty.Visible = True
809 141 Me.SubfrmFundamentalsPropertyList.Visible = False
810 142 End If
811 143
812 144 ' Set focus to the property name. This prevents problem
813 145 associated with entering the "Operator" object.
814 146 FPropertyName_PK.SetFocus
815 147 End If
816 148
817 149
818 150 End Sub
819 151
820 152 Private Sub FPropertyName_PK_AfterUpdate()
821 153
822 154 Dim strSQL As String
823 155
824 156 ' Forces user to enter data for property name and description
825 157 If IsNull(Me.FPropertyName_PK.Value) Or IsNull(Me.Description.Value)
826 158 Me.MarineField.Enabled = False
827 159 Me.Operator.Enabled = False
828 160 Me.LabID.Visible = False
829 161 Me.SubfrmFundamentalsProperty.Visible = False
830 162 Me.SubfrmFundamentalsPropertyList.Visible = False
831 163
832 164 ' Allows the user to enter values for the fundamental equation
833 165 Else
834 166 Me.MarineField.Enabled = True
835 167 Me.Operator.Enabled = True
836 168 Me.LabID.Visible = True
837 169 If Forms!frmFundamentalsProperty.Operator.Value = "In" Then
838 170 ' Update the Potential Value List
839 171 strSQL = "SELECT DISTINCT * & Me.MarineField & * FROM
840 172
841 Forms!frmFundamentalsProperty!SubfrmFundamentalsPropertyList!Value_FK.RowSo
842 urce = strSQL
843 173 Me.SubfrmFundamentalsProperty.Visible = False
844 174 Me.SubfrmFundamentalsPropertyList.Visible = True
845 175 Else
846 176 strSQL = "SELECT DISTINCT * & Me.MarineField & * FROM
847 177
848 Forms!frmFundamentalsProperty!SubfrmFundamentalsProperty!cmbValue_FK.RowSou
849 rce = strSQL
850 178 Me.SubfrmFundamentalsProperty.Visible = True
851 179 Me.SubfrmFundamentalsPropertyList.Visible = False
852 180 End If
853 181 End If
854 182
855 183 End Sub
856 184
857 185 Private Sub MarineField_AfterUpdate()
858 186
859 187 Dim strSQL As String
860 188
861 189 ' Update the Potential Value list
862 190 strSQL = "SELECT DISTINCT * & Me.MarineField & * FROM MARINE;"
863 191
864 192
865 Forms!frmFundamentalsProperty!SubfrmFundamentalsPropertyList!Value_FK.RowSo
866 urce = strSQL
867 193
868 Forms!frmFundamentalsProperty!SubfrmFundamentalsProperty!cmbValue_FK.RowSou
869 rce = strSQL
870 194 End Sub
871 195
872 196 Private Sub Operator_AfterUpdate()
873 197
874 198 If Forms!frmFundamentalsProperty.Operator.Value = "In" Then
875 199 Me.SubfrmFundamentalsProperty.Visible = False
876 200 Me.SubfrmFundamentalsPropertyList.Visible = True
877 201 Else
878 202 Me.SubfrmFundamentalsProperty.Visible = True
879 203 Me.SubfrmFundamentalsPropertyList.Visible = False
880 204 End If
881 205
882 206 End Sub
883 207 Private Sub btnDelete_Click()
884 208 On Error GoTo Err_btnDelete_Click
885 209
886 210
887 211 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
888 212 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
889 213
890 214 Exit Sub
891 215
892 216
893 217 Err_btnDelete_Click
894 218 MsgBox Err.Description
895 219 Resume Exit_btnDelete_Click
896 220
897 221 End Sub
898 222 Private Sub Command14_Click()
899 223 On Error GoTo Err_Command14_Click
900 224
901 225 Dim stDiaStr As String
902 226 Dim PrevCI As Control
903 227 Const ERR_OBJNOTEXIST = 2467
904 228 Const ERR_OBJNOTSET = 91
905 229
906 230 Set PrevCI = Screen.PreviousControl
907 231
908 232 If TypeOf PrevCI Is TextBox Then
909 233 stDiaStr = IIf(VarType(PrevCI) > V_NULL, PrevCI, "")
910
911 234 ElseIf TypeOf PrevCI Is ListBox Then
912 235 stDiaStr = IIf(VarType(PrevCI) > V_NULL, PrevCI, "")
913 236 ElseIf TypeOf PrevCI Is ComboBox Then
914 237 stDiaStr = IIf(VarType(PrevCI) > V_NULL, PrevCI, "")
915 238 Else
916 239 stDiaStr = ""
917 240 End If
918 241
919 242 Application.Run "Utility.mdb_AutoDia", stDiaStr
920 243
921 244 Exit Command14_Click
922 245 Exit Sub
923 246
924 247 Err_Command14_Click
925 248 If Err = ERR_OBJNOTEXIST Or Err = ERR_OBJNOTSET Then
926 249 Resume Next
927 250 End If
928 251 MsgBox Err.Description
929 252 Resume Exit_Command14_Click
930 253
931 254 End Sub
932 255
933 256 Private Sub Operator_Enter()
934 257
935 258 If Me.Operator.Value = "In" Then
936 259 MsgBox "The 'In' operator is not changeable. It is removed by
937 260 deleting the property. ", vbInformation, "Unauthorized Action"
938 261 subfrmFundamentalsPropertyList.SetFocus
939 262 End If
940 263 End Sub
941 264
942 265
943 Form: frmGenerateReport
944 266 Code
945 267 1 Attribute VB_Name = "Form_frmGenerateReport"
946 268 2 Attribute VB_Creatable = True
947 269 3 Attribute VB_PredeclaredId = True
948 270 4 Attribute VB_Exposed = False
949 271 5 Option Compare Database
950 272 6 Option Explicit
951 273
952 274 Private Sub btnAssignmentReport_Click()
953 275
954 276 DoCmd.OpenReport "rptAssignedMarines", acViewPreview
955 277
956 278 End Sub
957 279
958 280 Private Sub btnEntryLevelSchool_Click()
959 281
960 282 DoCmd.OpenReport "rptEntryLevelSchool", acViewPreview
961 283
962 284 End Sub
963 285
964 286 Private Sub btnExit_Click()
965 287 On Error GoTo Err_btnExit_Click
966 288
967 289 DoCmd.Close
970 290
971 291 Exit btnExit_Click
972 292 Exit Sub
973 293
974 294 Err_btnExit_Click
975 295 MsgBox Err.Description
976 296 Resume Exit_btnExit_Click
977 297
978 298 End Sub
979 299 Private Sub btnUnassignedMarineReport_Click()
980 300
981 301 DoCmd.OpenReport "rptUnassignedMarines", acViewPreview
982 302
983 303 End Sub
984 304
985 305 End Sub
986 306
987 307 Private Sub Command8_Click()
988 308 On Error GoTo Err_Command8_Click
989 309
990 310 Dim stDocName As String
991 311 Dim stLinkCriteria As String
992 312
993 313 ' Close the current form
994 314 DoCmd.Close
995 315 ' Open specified form
996 316 stDocName = "frmForm_Main_Switchboard"
997 317 DoCmd.OpenForm stDocName, , stLinkCriteria
998 318
999 319 Exit Command8_Click
1000 320 Exit Sub
1001 321
1002 322 Err_Command8_Click
1003 323 MsgBox Err.Description
1004 324 Resume Exit_Command8_Click
1005 325
1006 326 End Sub
1007 327
1008 Form: frmImportExportSwitchboard
1009 328 Code
1010 329 1 Attribute VB_Name = "Form_frmImportExportSwitchboard"
1011 330 2 Attribute VB_Creatable = True
1012 331 3 Attribute VB_PredeclaredId = True
1013 332 4 Attribute VB_Exposed = False
1014 333 5 Option Compare Database
1015 334 6 Option Explicit
1016 335
1017 336 Private Sub btnExit_Click()
1018 337 On Error GoTo Err_btnExit_Click
1019 338
1020 339 DoCmd.Close
1021 340
1022 341 Exit btnExit_Click
1023 342 Exit Sub
1024 343
1025 344 Err_btnExit_Click
1026 345 MsgBox Err.Description
1027 346 Resume Exit_btnExit_Click
1028 347
1029 348 End Sub
1030 349
1031 350 End Sub
1032 351
1033 352 Private Sub btnExportR03_Click()
1034 353
1035 354 Application.SetOption "Confirm Action Queries", False
1036 355 DoCmd.SubWarnings (False)
1037 356 ' Exports assignments to R03 bit file
1038 357 ExportR03
1039 358 ' Archives data from the Marine and assignment tables, having report
1040 359 dates less than the current date

```

```

1041 31 ArchiveData
1042 32 Application.SetOption "Confirm Action Queries", True
1043 33 DoCmd.SetWarnings (True)
1044 34
1045 35 End Sub
1046
1047 37
1048 38 Private Sub Command2_Click()
1049 39
1050 40 Dim stDocName As String
1051 41 Dim stLinkCriteria As String
1052 42
1053 43 Application.SetOption "Confirm Action Queries", False
1054 44 DoCmd.SetWarnings (False)
1055 45 'Imports the RDT file
1056 46 ImportRDT
1057 47 Application.SetOption "Confirm Action Queries", True
1058 48 DoCmd.SetWarnings (True)
1059 49
1060 50 ' Close the current form
1061 51 DoCmd.Close
1062 52 ' Open specified form
1063 53 stDocName = "frmChangeGradData"
1064 54 DoCmd.OpenForm stDocName, , stLinkCriteria
1065 55
1066 56 End Sub
1067 57
1068 58 Private Sub Command4_Click()
1069 59
1070 60 'Imports BNA.tbl file
1071 61 Application.SetOption "Confirm Action Queries", False
1072 62 DoCmd.SetWarnings (False)
1073 63 DoCmd.RunSQL "DELETE FROM BNA_EXTRACT;"
1074 64 ImportBNA
1075 65 Application.SetOption "Confirm Action Queries", True
1076 66 DoCmd.SetWarnings (True)
1077 67
1078 68 End Sub
1079 69
1080 70 Private Sub Command8_Click()
1081 71
1082 72 On Error GoTo Err_Command8_Click
1083 73
1084 74 Dim stDocName As String
1085 75 Dim stLinkCriteria As String
1086 76
1087 77 ' Close the current form
1088 78 DoCmd.Close
1089 79 ' Open specified form
1090 80 stDocName = "frmRDM_Main_Switchboard"
1091 81 DoCmd.OpenForm stDocName, , stLinkCriteria
1092 82
1093 83 Exit_Command8_Click
1094 84 Exit Sub
1095 85
1096 86 Err_Command8_Click
1097 87 MsgBox Err.Description
1098 88 Resume Exit_Command8_Click
1099 89
1100 90 End Sub
1101
1102 Form: frmLogicalProperty
1103 Code
1104 1 Attribute VB_Name = "Form_frmLogicalProperty"
1105 2 Attribute VB_Creatable = True
1106 3 Attribute VB_PredeclaredId = True
1107 4 Attribute VB_Exposed = False
1108 5 Option Compare Database
1109 6 Option Explicit
1110 7
1111 8 Private Sub btnClose_Click()
1112 9 On Error GoTo Err_btnClose_Click
1113 10
1114 11
1115 12 DoCmd.Close
1116 13 DoCmd.OpenForm "frmMaintenanceSwitchboard"
1117 14
1118 15 Exit_btnClose_Click
1119 16 Exit Sub
1120 17
1121 18 Err_btnClose_Click
1122 19 MsgBox Err.Description
1123 20 Resume Exit_btnClose_Click
1124 21
1125 22 End Sub
1126 23
1127 24 Private Sub cmbFundProp_Click()
1128 25
1129 26 ' Adds the selected fundamental property to the logical equation
1130 27 Me.LogicalEquation.Value = Me.LogicalEquation.Value &
1131 28 Me.cmbFundProp.Value & " "
1132 29
1133 30 End Sub
1134 31 Private Sub cmbLogOper_Click()
1135 32
1136 33 ' Adds the selected logical operator to the logical equation
1137 34 Me.LogicalEquation.Value = Me.LogicalEquation.Value &
1138 35 Me.cmbLogOper.Value & " "
1139 36
1140 37 End Sub
1141 38
1142 39 Private Sub cmbPropertyNameFind_AfterUpdate()
1143 40
1144 41 Dim R As Recordset
1145 42 Set R = Me.RecordsetClone
1146 43 R.FindFirst "[!PropertyName_PK] = " & Chr(34) &
1147 44 Me.cmbPropertyNameFind & Chr(34)
1148 45 Me.Bookmark = R.Bookmark
1149 46
1150 47 Me!cmbPropertyNameFind = Null
1151 48
1152 49 End Sub
1153 50
1154 51 Private Sub Command10_Click()
1155 52
1156 53 ' Removes the last entry in the logical equation
1157 54 Dim stInStrName, strUndo As String ' input name
1158 55 Dim lnSpacePos As Integer ' position of the space
1159 56 Dim i As Integer ' loop index
1160 57 Dim booResult As Boolean
1161 58 Dim lngConversion As Long
1162 59 Dim varConv As Variant
1163 60
1164 61 'get the logical expression
1165 62 strInStrName = Me.LogicalEquation.Value
1166 63
1167 64 'Check for "Null" value
1168 65 If strInStrName <> "Null" Then
1169 66
1170 67
1171 68
1172 69
1173 70
1174 71
1175 72
1176 73
1177 74
1178 75
1179 76
1180 77
1181 78
1182 79
1183 80
1184 81
1185 82
1186 83
1187 84
1188 85
1189 86
1190 87
1191 88
1192 89
1193 90
1194 91
1195 92
1196 93
1197 94
1198 95
1199 96
1200 97
1201 98
1202 99
1203 100
1204 101
1205 102
1206 103
1207 104
1208 105
1209 106
1210 107
1211 108
1212 109
1213 110
1214 111
1215 112
1216 113
1217 114
1218 115
1219 116
1220 117
1221 118
1222 119
1223 120
1224 121
1225 122
1226 123
1227 124
1228 125
1229 126
1230 127
1231 128
1232 129
1233 130
1234 131
1235 132
1236 133
1237 134
1238 135
1239 136
1240 137
1241 138
1242 139
1243 140
1244 141
1245 142
1246 143
1247 144
1248 145
1249 146
1250 147
1251 148
1252 149
1253 150
1254 151
1255 152
1256 153
1257 154
1258 155
1259 156
1260 157
1261 158
1262 159
1263 160
1264 161
1265 162
1266 163
1267 164
1268 165
1269 166
1270 167
1271 168
1272 169
1273 170
1274 171
1275 172
1276 173
1277 174
1278 175
1279 176
1280 177
1281 178
1282 179
1283 180
1284 181
1285 182
1286 183
1287 184
1288 185
1289 186
1290 187
1291 188
1292 189
1293 190
1294 191
1295 192
1296 193
1297 194
1298 195
1299 196
1300 197
1301 198
1302 199
1303 200
1304 201
1305 202
1306 203
1307 204
1308 205
1309 206
1310 207
1311 208
1312 209
1313 210
1314 211
1315 212
1316 213
1317 214
1318 215
1319 216
1320 217
1321 218
1322 219
1323 220
1324 221
1325 222
1326 223
1327 224
1328 225
1329 226
1330 227
1331 228
1332 229
1333 230
1334 231
1335 232
1336 233
1337 234
1338 235
1339 236
1340 237
1341 238
1342 239
1343 240
1344 241
1345 242
1346 243
1347 244
1348 245
1349 246
1350 247
1351 248
1352 249
1353 250
1354 251
1355 252
1356 253
1357 254
1358 255
1359 256
1360 257
1361 258
1362 259
1363 260
1364 261
1365 262
1366 263
1367 264
1368 265
1369 266
1370 267
1371 268
1372 269
1373 270
1374 271
1375 272
1376 273
1377 274
1378 275
1379 276
1380 277
1381 278
1382 279
1383 280
1384 281
1385 282
1386 283
1387 284
1388 285
1389 286
1390 287
1391 288
1392 289
1393 290
1394 291
1395 292
1396 293
1397 294
1398 295
1399 296
1400 297
1401 298
1402 299
1403 300
1404 301
1405 302
1406 303
1407 304
1408 305
1409 306
1410 307
1411 308
1412 309
1413 310
1414 311
1415 312
1416 313
1417 314
1418 315
1419 316
1420 317
1421 318
1422 319
1423 320
1424 321
1425 322
1426 323
1427 324
1428 325
1429 326
1430 327
1431 328
1432 329
1433 330
1434 331
1435 332
1436 333
1437 334
1438 335
1439 336
1440 337
1441 338
1442 339
1443 340
1444 341
1445 342
1446 343
1447 344
1448 345
1449 346
1450 347
1451 348
1452 349
1453 350
1454 351
1455 352
1456 353
1457 354
1458 355
1459 356
1460 357
1461 358
1462 359
1463 360
1464 361
1465 362
1466 363
1467 364
1468 365
1469 366
1470 367
1471 368
1472 369
1473 370
1474 371
1475 372
1476 373
1477 374
1478 375
1479 376
1480 377
1481 378
1482 379
1483 380
1484 381
1485 382
1486 383
1487 384
1488 385
1489 386
1490 387
1491 388
1492 389
1493 390
1494 391
1495 392
1496 393
1497 394
1498 395
1499 396
1500 397
1501 398
1502 399
1503 400
1504 401
1505 402
1506 403
1507 404
1508 405
1509 406
1510 407
1511 408
1512 409
1513 410
1514 411
1515 412
1516 413
1517 414
1518 415
1519 416
1520 417
1521 418
1522 419
1523 420
1524 421
1525 422
1526 423
1527 424
1528 425
1529 426
1530 427
1531 428
1532 429
1533 430
1534 431
1535 432
1536 433
1537 434
1538 435
1539 436
1540 437
1541 438
1542 439
1543 440
1544 441
1545 442
1546 443
1547 444
1548 445
1549 446
1550 447
1551 448
1552 449
1553 450
1554 451
1555 452
1556 453
1557 454
1558 455
1559 456
1560 457
1561 458
1562 459
1563 460
1564 461
1565 462
1566 463
1567 464
1568 465
1569 466
1570 467
1571 468
1572 469
1573 470
1574 471
1575 472
1576 473
1577 474
1578 475
1579 476
1580 477
1581 478
1582 479
1583 480
1584 481
1585 482
1586 483
1587 484
1588 485
1589 486
1590 487
1591 488
1592 489
1593 490
1594 491
1595 492
1596 493
1597 494
1598 495
1599 496
1600 497
1601 498
1602 499
1603 500
1604 501
1605 502
1606 503
1607 504
1608 505
1609 506
1610 507
1611 508
1612 509
1613 510
1614 511
1615 512
1616 513
1617 514
1618 515
1619 516
1620 517
1621 518
1622 519
1623 520
1624 521
1625 522
1626 523
1627 524
1628 525
1629 526
1630 527
1631 528
1632 529
1633 530
1634 531
1635 532
1636 533
1637 534
1638 535
1639 536
1640 537
1641 538
1642 539
1643 540
1644 541
1645 542
1646 543
1647 544
1648 545
1649 546
1650 547
1651 548
1652 549
1653 550
1654 551
1655 552
1656 553
1657 554
1658 555
1659 556
1660 557
1661 558
1662 559
1663 560
1664 561
1665 562
1666 563
1667 564
1668 565
1669 566
1670 567
1671 568
1672 569
1673 570
1674 571
1675 572
1676 573
1677 574
1678 575
1679 576
1680 577
1681 578
1682 579
1683 580
1684 581
1685 582
1686 583
1687 584
1688 585
1689 586
1690 587
1691 588
1692 589
1693 590
1694 591
1695 592
1696 593
1697 594
1698 595
1699 596
1700 597
1701 598
1702 599
1703 600
1704 601
1705 602
1706 603
1707 604
1708 605
1709 606
1710 607
1711 608
1712 609
1713 610
1714 611
1715 612
1716 613
1717 614
1718 615
1719 616
1720 617
1721 618
1722 619
1723 620
1724 621
1725 622
1726 623
1727 624
1728 625
1729 626
1730 627
1731 628
1732 629
1733 630
1734 631
1735 632
1736 633
1737 634
1738 635
1739 636
1740 637
1741 638
1742 639
1743 640
1744 641
1745 642
1746 643
1747 644
1748 645
1749 646
1750 647
1751 648
1752 649
1753 650
1754 651
1755 652
1756 653
1757 654
1758 655
1759 656
1760 657
1761 658
1762 659
1763 660
1764 661
1765 662
1766 663
1767 664
1768 665
1769 666
1770 667
1771 668
1772 669
1773 670
1774 671
1775 672
1776 673
1777 674
1778 675
1779 676
1780 677
1781 678
1782 679
1783 680
1784 681
1785 682
1786 683
1787 684
1788 685
1789 686
1790 687
1791 688
1792 689
1793 690
1794 691
1795 692
1796 693
1797 694
1798 695
1799 696
1800 697
1801 698
1802 699
1803 700
1804 701
1805 702
1806 703
1807 704
1808 705
1809 706
1810 707
1811 708
1812 709
1813 710
1814 711
1815 712
1816 713
1817 714
1818 715
1819 716
1820 717
1821 718
1822 719
1823 720
1824 721
1825 722
1826 723
1827 724
1828 725
1829 726
1830 727
1831 728
1832 729
1833 730
1834 731
1835 732
1836 733
1837 734
1838 735
1839 736
1840 737
1841 738
1842 739
1843 740
1844 741
1845 742
1846 743
1847 744
1848 745
1849 746
1850 747
1851 748
1852 749
1853 750
1854 751
1855 752
1856 753
1857 754
1858 755
1859 756
1860 757
1861 758
1862 759
1863 760
1864 761
1865 762
1866 763
1867 764
1868 765
1869 766
1870 767
1871 768
1872 769
1873 770
1874 771
1875 772
1876 773
1877 774
1878 775
1879 776
1880 777
1881 778
1882 779
1883 780
1884 781
1885 782
1886 783
1887 784
1888 785
1889 786
1890 787
1891 788
1892 789
1893 790
1894 791
1895 792
1896 793
1897 794
1898 795
1899 796
1900 797
1901 798
1902 799
1903 800
1904 801
1905 802
1906 803
1907 804
1908 805
1909 806
1910 807
1911 808
1912 809
1913 810
1914 811
1915 812
1916 813
1917 814
1918 815
1919 816
1920 817
1921 818
1922 819
1923 820
1924 821
1925 822
1926 823
1927 824
1928 825
1929 826
1930 827
1931 828
1932 829
1933 830
1934 831
1935 832
1936 833
1937 834
1938 835
1939 836
1940 837
1941 838
1942 839
1943 840
1944 841
1945 842
1946 843
1947 844
1948 845
1949 846
1950 847
1951 848
1952 849
1953 850
1954 851
1955 852
1956 853
1957 854
1958 855
1959 856
1960 857
1961 858
1962 859
1963 860
1964 861
1965 862
1966 863
1967 864
1968 865
1969 866
1970 867
1971 868
1972 869
1973 870
1974 871
1975 872
1976 873
1977 874
1978 875
1979 876
1980 877
1981 878
1982 879
1983 880
1984 881
1985 882
1986 883
1987 884
1988 885
1989 886
1990 887
1991 888
1992 889
1993 890
1994 891
1995 892
1996 893
1997 894
1998 895
1999 896
2000 897
2001 898
2002 899
2003 900
2004 901
2005 902
2006 903
2007 904
2008 905
2009 906
2010 907
2011 908
2012 909
2013 910
2014 911
2015 912
2016 913
2017 914
2018 915
2019 916
2020 917
2021 918
2022 919
2023 920
2024 921
2025 922
2026 923
2027 924
2028 925
2029 926
2030 927
2031 928
2032 929
2033 930
2034 931
2035 932
2036 933
2037 934
2038 935
2039 936
2040 937
2041 938
2042 939
2043 940
2044 941
2045 942
2046 943
2047 944
2048 945
2049 946
2050 947
2051 948
2052 949
2053 950
2054 951
2055 952
2056 953
2057 954
2058 955
2059 956
2060 957
2061 958
2062 959
2063 960
2064 961
2065 962
2066 963
2067 964
2068 965
2069 966
2070 967
2071 968
2072 969
2073 970
2074 971
2075 972
2076 973
2077 974
2078 975
2079 976
2080 977
2081 978
2082 979
2083 980
2084 981
2085 982
2086 983
2087 984
2088 985
2089 986
2090 987
2091 988
2092 989
2093 990
2094 991
2095 992
2096 993
2097 994
2098 995
2099 996
2100 997
2101 998
2102 999
2103 1000
2104 1001
2105 1002
2106 1003
2107 1004
2108 1005
2109 1006
2110 1007
2111 1008
2112 1009
2113 1010
2114 1011
2115 1012
2116 1013
2117 1014
2118 1015
2119 1016
2120 1017
2121 1018
2122 1019
2123 1020
2124 1021
2125 1022
2126 1023
2127 1024
2128 1025
2129 1026
2130 1027
2131 1028
2132 1029
2133 1030
2134 1031
2135 1032
2136 1033
2137 1034
2138 1035
2139 1036
2140 1037
2141 1038
2142 1039
2143 1040
2144 1041
2145 1042
2146 1043
2147 1044
2148 1045
2149 1046
2150 1047
2151 1048
2152 1049
2153 1050
2154 1051
2155 1052
2156 1053
2157 1054
2158 1055
2159 1056
2160 1057
2161 1058
2162 1059
2163 1060
2164 1061
2165 1062
2166 1063
2167 1064
2168 1065
2169 1066
2170 1067
2171 1068
2172 1069
2173 1070
2174 1071
2175 1072
2176 1073
2177 1074
2178 1075
2179 1076
2180 1077
2181 1078
2182 1079
2183 1080
2184 1081
2185 1082
2186 1083
2187 1084
2188 1085
2189 1086
2190 1087
2191 1088
2192 1089
2193 1090
2194 1091
2195 1092
2196 1093
2197 1094
2198 1095
2199 1096
2200 1097
2201 1098
2202 1099
2203 1100
2204 1101
2205 1102
2206 1103
2207 1104
2208 1105
2209 1106
2210 1107
2211 1108
2212 1109
2213 1110
2214 1111
2215 1112
2216 1113
2217 1114
2218 1115
2219 1116
2220 1117
2221 1118
2222 1119
2223 1120
2224 1121
2225 1122
2226 1123
2227 1124
2228 1125
2229 1126
2230 1127
2231 1128
2232 1129
2233 1130
2234 1131
2235 1132
2236 1133
2237 1134
2238 1135
2239 1136
2240 1137
2241 1138
2242 1139
2243 1140
2244 1141
2245 1142
2246 1143
2247 1144
2248 1145
2249 1146
2250 1147
2251 1148
2252 1149
2253 1150
2254 1151
2255 1152
2256 1153
2257 1154
2258 1155
2259 1156
2260 1157
2261 1158
2262 1159
2263 1160
2264 1161
2265 1162
2266 1163
2267 1164
2268 1165
2269 1166
2270 1167
2271 1168
2272 1169
2273 1170
2274 1171
2275 1172
2276 1173
2277 1174
2278 1175
2279 1176
2280 1177
2281 1178
2282 1179
2283 1180
2284 1181
2285 1182
2286 1183
2287 1184
2288 1185
2289 1186
2290 1187
2291 1188
2292 1189
2293 1190
2294 1191
2295 1192
2296 1193
2297 1194
2298 1195
2299 1196
2300 1197
2301 1198
2302 1199
2303 1200
2304 1201
2305 1202
2306 1203
2307 1204
2308 1205
2309 1206
2310 1207
2311 1208
2312 1209
2313 1210
2314 1211
2315 1212
2316 1213
2317 1214
2318 1215
2319 1216
2320 1217
2321 1218
2322 1219
2323 1220
2324 1221
2325 1222
2326 1223
2327 1224
2328 1225
2329 1226
2330 1227
2331 1228
2332 1229
2333 1230
2334 1231
2335 1232
2336 1233
2337 1234
2338 1235
2339 1236

```

```

1301 188 Me!Command9.Enabled = True
1302 189 Me!Command9.Enabled = True
1303 190 Me!Command10.Enabled = True
1304 191 Me!cmbFundProp.Enabled = True
1305 192 Me!cmbLogOper.Enabled = True
1306 193 Me!LogicalEquation.Enabled = True
1307 194 Me!LogicalEquation.Value = "("
1308 195
1309 196 ' Allows the user to enter values into the logical equation
1310 197 Else
1311 198 Me!Command8.Enabled = True
1312 199 Me!Command9.Enabled = True
1313 200 Me!Command10.Enabled = True
1314 201 Me!cmbFundProp.Enabled = True
1315 202 Me!cmbLogOper.Enabled = True
1316 203 Me!LogicalEquation.Enabled = True
1317 204 End If
1318 205
1319 206 End Sub
1320 207
1321 208 Private Sub Form_BeforeUpdate(Cancel As Integer)
1322 209
1323 210 (LogPropTimeStamp) = Now
1324 211
1325 212 End Sub
1326 213
1327 214 Private Sub Form_Current()
1328 215
1329 216 ' Forces user to enter data for property name and description
1330 217 If IsNull(Me.LPropertyName_PK.Value) Or IsNull(Me.Description.Value)
1331 218 Me!Command8.Enabled = False
1332 219 Me!Command9.Enabled = False
1333 220 Me!Command10.Enabled = False
1334 221 Me!cmbFundProp.Enabled = False
1335 222 Me!cmbLogOper.Enabled = False
1336 223 Me!LogicalEquation.Enabled = False
1337 224
1338 225 ' Enters a "(" be the first entry in the logical equation if it is
1339 226 ElseIf IsNull(Me.LogicalEquation.Value) Then
1340 227 Me!LogicalEquation.Value = "("
1341 228
1342 229 ' Allows the user to enter values into the logical equation
1343 230 Else
1344 231 Me!Command8.Enabled = True
1345 232 Me!Command9.Enabled = True
1346 233 Me!Command10.Enabled = True
1347 234 Me!cmbFundProp.Enabled = True
1348 235 Me!cmbLogOper.Enabled = True
1349 236 Me!LogicalEquation.Enabled = True
1350 237 End If
1351 238
1352 239 End Sub
1353 240 Private Sub LogicalEquation_Enter()
1354 241
1355 242 ' Prevents the user from modifying the logical equation directly
1356 243 MsgBox "The logical equation is only modifiable by the above logical
1357 244 equation builder"
1358 245 DoCmd.GoToControl "LPropertyName_PK"
1359 246
1360 246 End Sub
1361 247
1362 248 Private Sub LPropertyName_PK_AfterUpdate()
1363 249
1364 250 ' Forces user to enter data for property name and description
1365 251 If IsNull(Me.LPropertyName_PK.Value) Or IsNull(Me.Description.Value)
1366 252 Me!Command8.Enabled = False
1367 253 Me!Command9.Enabled = False
1368 254 Me!Command10.Enabled = False
1369 255 Me!cmbFundProp.Enabled = False
1370 256 Me!cmbLogOper.Enabled = False
1371 257 Me!LogicalEquation.Enabled = False
1372 258
1373 259 ' Enters a "(" be the first entry in the logical equation if it is
1374 260 ElseIf IsNull(Me.LogicalEquation.Value) Then
1375 261 Me!Command8.Enabled = True
1376 262 Me!Command9.Enabled = True
1377 263 Me!Command10.Enabled = True
1378 264 Me!cmbFundProp.Enabled = True
1379 265 Me!cmbLogOper.Enabled = True
1380 266 Me!LogicalEquation.Enabled = True
1381 267 Me!LogicalEquation.Value = "("
1382 268
1383 269 ' Allows the user to enter values into the logical equation
1384 270 Else
1385 271 Me!Command8.Enabled = True
1386 272 Me!Command9.Enabled = True
1387 273 Me!Command10.Enabled = True
1388 274 Me!cmbFundProp.Enabled = True
1389 275 Me!cmbLogOper.Enabled = True
1390 276 Me!LogicalEquation.Enabled = True
1391 277 End If
1392 278
1393 279 End Sub
1394
1395 Form: frmMaintenanceSwitchboard
1396 Code
1397 1 Attribute VB_Name = "Form_frmMaintenanceSwitchboard"
1398 2 Attribute VB_Creatable = True
1399 3 Attribute VB_PredeclaredId = True
1400 4 Attribute VB_Exposed = False
1401 5 Option Compare Database
1402 6 Option Explicit
1403 7
1404 8 Private Sub btnExit_Click()
1405 9 On Error GoTo Err_btnExit_Click
1406 10
1407 11
1408 12 DoCmd.Close
1409 13
1410 14 Exit_btnExit_Click
1411 15 Exit Sub
1412 16
1413 17 Err_btnExit_Click
1414 18 MsgBox Err.Description
1415 19 Resume Exit_btnExit_Click
1416 20
1417 21
1418 22 End Sub
1419 23
1420 24 Private Sub Command0_Click()
1421 25 On Error GoTo Err_Command0_Click
1422 26
1423 27 Dim stDocName As String
1424 28 Dim stLinkCriteria As String
1425 29
1426 30 ' Close the current form
1427 31 DoCmd.Close
1428 32 ' Open specified form
1429 33 stDocName = "frmSchoolToPEF_Maint"
1430 34 DoCmd.OpenForm stDocName, , , stLinkCriteria
1431 35
1432 36 Exit_Command0_Click
1433 37 Exit Sub
1434 38
1435 39 Err_Command0_Click
1436 40 MsgBox Err.Description
1437 41 Resume Exit_Command0_Click
1438 42
1439 43 End Sub
1440 44
1441 45 Private Sub Command12_Click()
1442 46 On Error GoTo Err_Command12_Click
1443 47
1444 48 Dim stDocName As String
1445 49 Dim stLinkCriteria As String
1446 50
1447 51 ' Close the current form
1448 52 DoCmd.Close
1449 53 ' Open specified form
1450 54 stDocName = "frmSchool"
1451 55 DoCmd.OpenForm stDocName, , , stLinkCriteria
1452 56
1453 57 Exit_Command12_Click
1454 58 Exit Sub
1455 59
1456 60 Err_Command12_Click
1457 61 MsgBox Err.Description
1458 62 Resume Exit_Command12_Click
1459 63
1460 64 End Sub
1461 65
1462 66 Private Sub Command14_Click()
1463 67 On Error GoTo Err_Command14_Click
1464 68
1465 69 Dim stDocName As String
1466 70 Dim stLinkCriteria As String
1467 71
1468 72 ' Close the current form
1469 73 DoCmd.Close
1470 74 ' Open specified form
1471 75 stDocName = "frmPEF"
1472 76 DoCmd.OpenForm stDocName, , , stLinkCriteria
1473 77
1474 78 Exit_Command14_Click
1475 79 Exit Sub
1476 80
1477 81 Err_Command14_Click
1478 82 MsgBox Err.Description
1479 83 Resume Exit_Command14_Click
1480 84
1481 85 End Sub
1482 86
1483 87 Private Sub Command2_Click()
1484 88 On Error GoTo Err_Command2_Click
1485 89
1486 90 Dim stDocName As String
1487 91 Dim stLinkCriteria As String
1488 92
1489 93 ' Close the current form
1490 94 DoCmd.Close
1491 95 ' Open specified form
1492 96 stDocName = "frmFundamentalProperty"
1493 97 DoCmd.OpenForm stDocName, , , stLinkCriteria
1494 98
1495 99 Exit_Command2_Click
1496 100 Exit Sub
1497 101
1498 102 Err_Command2_Click
1499 103 MsgBox Err.Description
1500 104 Resume Exit_Command2_Click
1501 105
1502 106 End Sub
1503 107
1504 108 Private Sub Command4_Click()
1505 109 On Error GoTo Err_Command4_Click
1506 110
1507 111 Dim stDocName As String
1508 112 Dim stLinkCriteria As String
1509 113
1510 114 ' Close the current form
1511 115 DoCmd.Close
1512 116 ' Open specified form
1513 117 stDocName = "frmLogicalProperty"
1514 118 DoCmd.OpenForm stDocName, , , stLinkCriteria
1515 119
1516 120 Exit_Command4_Click
1517 121 Exit Sub
1518 122
1519 123 Err_Command4_Click
1520 124 MsgBox Err.Description
1521 125 Resume Exit_Command4_Click
1522 126
1523 127 End Sub
1524 128
1525 129 Private Sub Command6_Click()
1526 130 On Error GoTo Err_Command6_Click
1527 131
1528 132 Dim stDocName As String
1529 133 Dim stLinkCriteria As String
1530 134
1531 135 ' Close the current form
1532 136 DoCmd.Close
1533 137 ' Open specified form
1534 138 stDocName = "frmSchoolToPropertyMaint"
1535 139 DoCmd.OpenForm stDocName, , , stLinkCriteria
1536 140
1537 141 Exit_Command6_Click
1538 142 Exit Sub
1539 143
1540 144 Err_Command6_Click
1541 145 MsgBox Err.Description
1542 146 Resume Exit_Command6_Click
1543 147
1544 148 End Sub
1545 149
1546 150 Private Sub Command8_Click()
1547 151 On Error GoTo Err_Command8_Click
1548 152
1549 153 Dim stDocName As String
1550 154 Dim stLinkCriteria As String
1551 155
1552 156 ' Close the current form
1553 157 DoCmd.Close
1554 158 ' Open specified form
1555 159 stDocName = "frmFDN_Maint_Switchboard"
1556 160 DoCmd.OpenForm stDocName, , , stLinkCriteria
1557 161
1558 162 Exit_Command8_Click
1559 163 Exit Sub
1560 164

```

```

1561 165 Err_Command8_Click
1562 166 MsgBox Err.Description
1563 167 Resume Ext_Command8_Click
1564 168
1565 169 End Sub
1566
1567 Form: frmPEF
1568 Code
1569 1 Attribute VB_Name = "Form_frmPEF"
1570 2 Attribute VB_Creatable = True
1571 3 Attribute VB_Procedureid = True
1572 4 Attribute VB_Exposed = False
1573 5 Option Compare Database
1574 6 Option Explicit
1575 7
1576 8 Private Sub cmbPeFnd_AfterUpdate()
1577 9
1578 10 Dim R As Recordset
1579 11 Set R = Me.RecordsetClone
1580 12 R.FindFirst "[PEF_PK] = " & Chr(34) & Me[cmbPeFnd] & Chr(34)
1581 13 Me.Bookmark = R.Bookmark
1582 14 Me[cmbPeFnd] = Null
1583 15 Me.PEF_PK.SetFocus
1584 16
1585 17 End Sub
1586 18
1587 19 Private Sub Form_BeforeUpdate(Cancel As Integer)
1588 20
1589 21 [PEF_TimeStamp] = Now
1590 22
1591 23 End Sub
1592 24
1593 25 Private Sub Form_Current()
1594 26
1595 27 Dim strSQL As String
1596 28
1597 29 'Update the PEF List
1598 30 strSQL = "SELECT DISTINCTROW [MARINE].[PEF]FROM MARINE LEFT JOIN PEF
1599 31 ON [MARINE].[PEF] = [PEF].[PEF_PK]WHERE ([PEF].[PEF_PK] Is Null)ORDER BY
1600 32 Me.PEF_PK.RowSource = strSQL
1601 33
1602 33 End Sub
1603 34
1604 35 Private Sub PEF_PK_AfterUpdate()
1605 36
1606 37 Dim strSQL As String
1607 38
1608 39 'Update the PEF List
1609 40 strSQL = "SELECT DISTINCTROW [MARINE].[PEF]FROM MARINE LEFT JOIN PEF
1610 41 ON [MARINE].[PEF] = [PEF].[PEF_PK]WHERE ([PEF].[PEF_PK] Is Null)ORDER BY
1611 42 Me.PEF_PK.RowSource = strSQL
1612 43
1613 43 End Sub
1614 44 Private Sub btnDelete_Click()
1615 45 On Error GoTo Err_btnDelete_Click
1616 46
1617 47 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 8, acMenuVer70
1618 48 DoCmd.DeleteMenuItem acFormBar, acEditMenu, 6, acMenuVer70
1619 49
1620 50
1621 51 Exit btnDelete_Click
1622 52 Exit Sub
1623 53
1624 54 Err_btnDelete_Click
1625 55 MsgBox Err.Description
1626 56 Resume Ext_btnDelete_Click
1627 57
1628 58 End Sub
1629 59 Private Sub btnClose_Click()
1630 60 On Error GoTo Err_btnClose_Click
1631 61
1632 62
1633 63 DoCmd.Close
1634 64 DoCmd.OpenForm "frmMaintenanceSwitchboard"
1635 65
1636 66 Exit btnClose_Click
1637 67 Exit Sub
1638 68
1639 69 Err_btnClose_Click
1640 70 MsgBox Err.Description
1641 71 Resume Ext_btnClose_Click
1642 72
1643 73 End Sub
1644
1645 Form: frmPrepareAndExecuteSoler
1646 Code
1647 1 Attribute VB_Name = "Form_frmPrepareAndExecuteSoler"
1648 2 Attribute VB_Creatable = True
1649 3 Attribute VB_Procedureid = True
1650 4 Attribute VB_Exposed = False
1651 5 Option Compare Database
1652 6 Option Explicit
1653 7
1654 8 Private Sub btnAMPL_Click()
1655 9 On Error GoTo Err_btnAMPL_Click
1656 10
1657 11 AmpData
1658 12
1659 13 Dim stAppName As String
1660 14
1661 15 stAppName = "C:\ToBeRdm\Amp\rdm\ampwin.exe"
1662 16 stAppName = "C:\ToBeRdm\Amp\rdm\amp.bat"
1663 17 Call Shell(stAppName, 0)
1664 18
1665 19 MsgBox "Enter into the command line: 'include rdm.run'"
1666 20
1667 21 Exit btnAMPL_Click
1668 22 Exit Sub
1669 23
1670 24 Err_btnAMPL_Click
1671 25 MsgBox Err.Description
1672 26 Resume Ext_btnAMPL_Click
1673 27
1674 28 End Sub
1675 29
1676 30 Private Sub btnAmplPlus_Click()
1677 31 On Error GoTo Err_btnAmplPlus_Click
1678 32
1679 33 AmpData
1680 34
1681 35 Dim stAppName As String
1682 36
1683 37 stAppName = "C:\AMPLPLUS\AMPLPLUS.EXE rdm.ampl"
1684 38 Call Shell(stAppName, 1)
1685 39
1686 40 Exit btnAmplPlus_Click
1687 41 Exit Sub
1688 42
1689 43 Err_btnAmplPlus_Click
1690 44 MsgBox Err.Description

```

```

1691 45 Resume Ext_btnAmplPlus_Click
1692 46
1693 47 End Sub
1694 48
1695 49
1696 50 Private Sub btnReturnPrevious_Click()
1697 51 On Error GoTo Err_btnReturnPrevious_Click
1698 52
1699 53
1700 54 Dim stDocName As String
1701 55 Dim stLinkCriteria As String
1702 56
1703 57 'Close the current form
1704 58 DoCmd.Close
1705 59 'Open specified form
1706 60 stDocName = "frmPreprocessing&ExecutionSwitchboard"
1707 61 DoCmd.OpenForm stDocName, , stLinkCriteria
1708 62
1709 63 Exit btnReturnPrevious_Click
1710 64 Exit Sub
1711 65
1712 66 Err_btnReturnPrevious_Click
1713 67 MsgBox Err.Description
1714 68 Resume Ext_btnReturnPrevious_Click
1715 69
1716 70
1717 71 End Sub
1718 72
1719 73 Private Sub Command4_Click()
1720 74 Application.SetOption "Confirm Action Queries", False
1721 75 DoCmd.SetWarnings (False)
1722 76 DoCmd.RunSQL "DELETE FROM AMPL_RESULT;"
1723 77 Amp_Result
1724 78 Application.SetOption "Confirm Action Queries", True
1725 79 DoCmd.SetWarnings (True)
1726 80
1727 81 End Sub
1728 82
1729 83 Private Sub Command7_Click()
1730 84 On Error GoTo Err_Command7_Click
1731 85
1732 86 Dim stDocName As String
1733 87 Dim stLinkCriteria As String
1734 88
1735 89 'Close the current form
1736 90 DoCmd.Close
1737 91 'Open specified form
1738 92 stDocName = "frmAnalyzeResult"
1739 93 DoCmd.OpenForm stDocName, , stLinkCriteria
1740 94
1741 95 Exit Command7_Click
1742 96 Exit Sub
1743 97
1744 98 Err_Command7_Click
1745 99 MsgBox Err.Description
1746 100 Resume Ext_Command7_Click
1747 101
1748 102 End Sub
1749
1750 Form: frmPreprocessing&ExecutionSwitchboard
1751 Code
1752 1 Attribute VB_Name = "Form_frmPreprocessing&ExecutionSwitchboard"
1753 2 Attribute VB_Creatable = True
1754 3 Attribute VB_Procedureid = True
1755 4 Attribute VB_Exposed = False
1756 5 Option Compare Database
1757 6 Option Explicit
1758 7
1759 8 Private Sub btnExt_Click()
1760 9 On Error GoTo Err_btnExt_Click
1761 10
1762 11
1763 12 DoCmd.Close
1764 13
1765 14 Exit btnExt_Click
1766 15 Exit Sub
1767 16
1768 17 Err_btnExt_Click
1769 18 MsgBox Err.Description
1770 19 Resume Ext_btnExt_Click
1771 20
1772 21
1773 22 End Sub
1774 23
1775 24 Private Sub btnReclassification_Click()
1776 25 On Error GoTo Err_btnReclassification_Click
1777 26
1778 27 Dim stDocName As String
1779 28 Dim stLinkCriteria As String
1780 29
1781 30 'Close the current form
1782 31 DoCmd.Close
1783 32 'Open specified form
1784 33 stDocName = "frmReclassification"
1785 34 DoCmd.OpenForm stDocName, , stLinkCriteria
1786 35
1787 36 Exit btnReclassification_Click
1788 37 Exit Sub
1789 38
1790 39 Err_btnReclassification_Click
1791 40 MsgBox Err.Description
1792 41 Resume Ext_btnReclassification_Click
1793 42
1794 43
1795 44 End Sub
1796 45
1797 46 Private Sub btnScrubMarineData_Click()
1798 47 On Error GoTo Err_btnScrubMarineData_Click
1799 48
1800 49 Dim stDocName As String
1801 50 Dim stLinkCriteria As String
1802 51
1803 52 'Close the current form
1804 53 DoCmd.Close
1805 54 'Open specified form
1806 55 stDocName = "frmScrubMarine"
1807 56 DoCmd.OpenForm stDocName, , stLinkCriteria
1808 57
1809 58 Exit btnScrubMarineData_Click
1810 59 Exit Sub
1811 60
1812 61 Err_btnScrubMarineData_Click
1813 62 MsgBox Err.Description
1814 63 Resume Ext_btnScrubMarineData_Click
1815 64
1816 65 End Sub
1817 66
1818 67 Private Sub btnSpecialAssignment_Click()
1819 68 On Error GoTo Err_btnSpecialAssignment_Click
1820 69

```

```

1821 70 Dim stDocName As String
1822 71 Dim stLinkCriteria As String
1823 72
1824 73 ' Close the current form
1825 74 DoCmd.Close
1826 75 ' Open specified form
1827 76 stDocName = "frmSpecialAssignment"
1828 77 DoCmd.OpenForm stDocName, , stLinkCriteria
1829 78
1830 79 Exit_btnSpecialAssignment_Click
1831 80 Exit Sub
1832 81
1833 82 Err_btnSpecialAssignment_Click
1834 83 MsgBox Err.Description
1835 84 Resume Exit_btnSpecialAssignment_Click
1836 85
1837 86
1838 87
1839 88 End Sub
1840 89
1841 90 Private Sub Command2_Click()
1842 91 On Error GoTo Err_Command2_Click
1843 92
1844 93 Dim stDocName As String
1845 94 Dim stLinkCriteria As String
1846 95
1847 96 ' Close the current form
1848 97 DoCmd.Close
1849 98 ' Open specified form
1850 99 stDocName = "frmClassQuotaPenaltyAndFit"
1851 100 DoCmd.OpenForm stDocName, , stLinkCriteria
1852 101
1853 102 Exit_Command2_Click
1854 103 Exit Sub
1855 104
1856 105 Err_Command2_Click
1857 106 MsgBox Err.Description
1858 107 Resume Exit_Command2_Click
1859 108
1860 109 End Sub
1861 110 Private Sub Command4_Click()
1862 111
1863 112 Application.SetOption "Confirm Action Queries", False
1864 113 DoCmd.SetWarnings (False)
1865 114 DoCmd.RunSQL "DELETE FROM AMPL_RESULT;"
1866 115 Ampl_Result
1867 116 Application.SetOption "Confirm Action Queries", True
1868 117 DoCmd.SetWarnings (True)
1869 118
1870 119
1871 120 End Sub
1872 121
1873 122 Private Sub Command5_Click()
1874 123 On Error GoTo Err_Command5_Click
1875 124
1876 125 Dim stDocName As String
1877 126 Dim stLinkCriteria As String
1878 127
1879 128 ' Close the current form
1880 129 DoCmd.Close
1881 130 ' Open specified form
1882 131 stDocName = "frmPrepareAndExecuteSolver"
1883 132 DoCmd.OpenForm stDocName, , stLinkCriteria
1884 133
1885 134 Exit_Command5_Click
1886 135 Exit Sub
1887 136
1888 137 Err_Command5_Click
1889 138 MsgBox Err.Description
1890 139 Resume Exit_Command5_Click
1891 140
1892 141 End Sub
1893 142
1894 143 Private Sub Command6_Click()
1895 144 On Error GoTo Err_Command6_Click
1896 145
1897 146 Dim stDocName As String
1898 147 Dim stLinkCriteria As String
1899 148
1900 149 ' Close the current form
1901 150 DoCmd.Close
1902 151 ' Open specified form
1903 152 stDocName = "frmRDM_Main_Switchboard"
1904 153 DoCmd.OpenForm stDocName, , stLinkCriteria
1905 154
1906 155 Exit_Command6_Click
1907 156 Exit Sub
1908 157
1909 158 Err_Command6_Click
1910 159 MsgBox Err.Description
1911 160 Resume Exit_Command6_Click
1912 161
1913 162 End Sub
1914
1915 Form: frmRDM_Main_Switchboard
1916 Code
1917 1 Attribute VB_Name = "Form_frmRDM_Main_Switchboard"
1918 2 Attribute VB_Creatable = True
1919 3 Attribute VB_PredeclaredId = True
1920 4 Attribute VB_Exposed = False
1921 5 Option Compare Database
1922 6 Option Explicit
1923 7 Private Sub Command0_Click()
1924 8
1925 9 On Error GoTo Err_Command0_Click
1926 10
1927 11 Dim stDocName As String
1928 12 Dim stLinkCriteria As String
1929 13
1930 14 ' Close the current form
1931 15 DoCmd.Close
1932 16 ' Open specified form
1933 17 stDocName = "frmImportExportSwitchboard"
1934 18 DoCmd.OpenForm stDocName, , stLinkCriteria
1935 19
1936 20 Exit_Command0_Click
1937 21 Exit Sub
1938 22
1939 23 Err_Command0_Click
1940 24 MsgBox Err.Description
1941 25 Resume Exit_Command0_Click
1942 26
1943 27 End Sub
1944 28 Private Sub Command2_Click()
1945 29 On Error GoTo Err_Command2_Click
1946 30
1947 31 Dim stDocName As String
1948 32 Dim stLinkCriteria As String
1949 33
1950 34 ' Close the current form

```

```

1951 35 DoCmd.Close
1952 36 ' Open specified form
1953 37 stDocName = "frmMaintenanceSwitchboard"
1954 38 DoCmd.OpenForm stDocName, , stLinkCriteria
1955 39
1956 40 Exit_Command2_Click
1957 41 Exit Sub
1958 42
1959 43 Err_Command2_Click
1960 44 MsgBox Err.Description
1961 45 Resume Exit_Command2_Click
1962 46
1963 47 End Sub
1964 48 Private Sub Command4_Click()
1965 49 On Error GoTo Err_Command4_Click
1966 50
1967 51 Dim stDocName As String
1968 52 Dim stLinkCriteria As String
1969 53
1970 54 ' Close the current form
1971 55 DoCmd.Close
1972 56 ' Open specified form
1973 57 stDocName = "frmPreprocessingExecutionSwitchboard"
1974 58 DoCmd.OpenForm stDocName, , stLinkCriteria
1975 59
1976 60 Exit_Command4_Click
1977 61 Exit Sub
1978 62
1979 63 Err_Command4_Click
1980 64 MsgBox Err.Description
1981 65 Resume Exit_Command4_Click
1982 66
1983 67 End Sub
1984 68 Private Sub Command5_Click()
1985 69 On Error GoTo Err_Command5_Click
1986 70
1987 71 Dim stDocName As String
1988 72 Dim stLinkCriteria As String
1989 73
1990 74 ' Close the current form
1991 75 DoCmd.Close
1992 76 ' Open specified form
1993 77 stDocName = "frmAnalyzeResult"
1994 78 DoCmd.OpenForm stDocName, , stLinkCriteria
1995 79
1996 80 Exit_Command5_Click
1997 81 Exit Sub
1998 82
1999 83 Err_Command5_Click
2000 84 MsgBox Err.Description
2001 85 Resume Exit_Command5_Click
2002 86
2003 87 End Sub
2004 88 Private Sub Command9_Click()
2005 89 On Error GoTo Err_Command9_Click
2006 90
2007 91 Dim stDocName As String
2008 92 Dim stLinkCriteria As String
2009 93
2010 94 ' Close the current form
2011 95 DoCmd.Close
2012 96 ' Open specified form
2013 97 stDocName = "frmGenerateReport"
2014 98 DoCmd.OpenForm stDocName, , stLinkCriteria
2015 99
2016 100 Exit_Command9_Click
2017 101 Exit Sub
2018 102
2019 103 Err_Command9_Click
2020 104 MsgBox Err.Description
2021 105 Resume Exit_Command9_Click
2022 106
2023 107 End Sub
2024 108 Private Sub btnExit_Click()
2025 109 On Error GoTo Err_btnExit_Click
2026 110
2027 111
2028 112 DoCmd.Close
2029 113
2030 114 Exit_btnExit_Click
2031 115 Exit Sub
2032 116
2033 117 Err_btnExit_Click
2034 118 MsgBox Err.Description
2035 119 Resume Exit_btnExit_Click
2036 120
2037 121 End Sub
2038
2039 Form: frmRecClassification
2040 Code
2041 1 Attribute VB_Name = "Form_frmRecClassification"
2042 2 Attribute VB_Creatable = True
2043 3 Attribute VB_PredeclaredId = True
2044 4 Attribute VB_Exposed = False
2045 5 Option Compare Database
2046 6 Option Explicit
2047 7
2048 8 Private Sub cmbClassNumber_AfterUpdate()
2049 9
2050 10 Dim strConvert As String
2051 11
2052 12 Me.txtReportDate = Me.cmbClassNumber.Column(2)
2053 13 Me.txtMCC = Me.cmbClassNumber.Column(4)
2054 14
2055 15 ' Convert the fiscal year to a two digit number for RDS File
2056 16 strConvert = CStr(Me.cmbClassNumber.Column(3))
2057 17 strConvert = Right(strConvert, 2)
2058 18 Me.txtFY = Chr(strConvert)
2059 19
2060 20 Me.txtAssignmentType = "R"
2061 21
2062 22 End Sub
2063 23
2064 24 Private Sub cmbClassNumber_Enter()
2065 25
2066 26 Dim strSQL As String
2067 27
2068 28 ' This query finds the class numbers, report dates and class
2069 29 converting dates associated with the chosen school
2070 30 strSQL = "SELECT BNA_EXTRACT.ClassNumber_PK,
2071 31 BNA_EXTRACT.ClassOnDate, BNA_EXTRACT.ReportDate,
2072 32 BNA_EXTRACT.FiscalYear_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT INNER JOIN
2073 33 SCH_TGT_MOS ON (BNA_EXTRACT.CourseNumber_PK =
2074 34 SCH_TGT_MOS.CourseNumber_PK) AND (BNA_EXTRACT.TargetMOS_PK =
2075 35 SCH_TGT_MOS.TargetMOS_PK) WHERE BNA_EXTRACT.CourseNumber_PK =
2076 36 Form!frmRecClassification!cmbClassNumber AND SCH_TGT_MOS.AMOS_PK =
2077 37 Form!frmRecClassification!txtAMOS;"
2078 38 Me.cmbClassNumber.RowSource = strSQL
2079 39
2080 40 End Sub

```

```

2081 33 Private Sub cmbCourseNumber_AfterUpdate()
2082 34
2083 35
2084 36 Me.tblAMOS = Me!cmbCourseNumber.Column(1)
2085 37
2086 38 End Sub
2087 39
2088 40 Private Sub cmbSSN_AfterUpdate()
2089 41
2090 42 Me.tblSQL = Me!cmbSSN.Column(1)
2091 43 Me.tblGradDate = Me!cmbSSN.Column(2)
2092 44
2093 45 End Sub
2094 46
2095 47 Private Sub cmbSSN_Enter()
2096 48
2097 49 Dim strSQL As String
2098 50
2099 51 strSQL = "SELECT DISTINCTROW MARINE.SSN_PK, MARINE.SOI,
2100 MARINE.GradDate FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK =
2101 ASSIGNMENT.SSN_FK WHERE (((ASSIGNMENT.SSN_FK) Is Null));"
2102 52 Me.cmbSSN.RowSource = strSQL
2103 53
2104 54 End Sub
2105 55
2106 56 Private Sub cmbSSNFind_AfterUpdate()
2107 57
2108 58 Dim R As Recordset
2109 59 Set R = Me.RecordsetClone
2110 60 R.FindFirst "[SSN_FK] = " & Chr(34) & Me!cmbSSNFind & Chr(34)
2111 61 Me.Bookmark = R.Bookmark
2112 62 Me!cmbSSNFind = Null
2113 63 Me.cmbSSN.SetFocus
2114 64
2115 65 End Sub
2116 66
2117 67 Private Sub Delete_Click()
2118 68 On Error GoTo Err_Delete_Click
2119 69
2120 70
2121 71 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2122 72 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2123 73
2124 74 Exit_Delete_Click
2125 75 Exit Sub
2126 76
2127 77 Err_Delete_Click:
2128 78 MsgBox Err.Description
2129 79 Resume Exit_Delete_Click
2130 80
2131 81 End Sub
2132 82
2133 83 Private Sub Close_Click()
2134 84 On Error GoTo Err_Close_Click
2135 85
2136 86 Dim stDocName As String
2137 87 Dim stLinkCriteria As String
2138 88
2139 89 ' Close current form
2140 90 DoCmd.Close
2141 91
2142 92 ' Open specified form
2143 93 stDocName = "frmProcessing&ExecutionSwitchboard"
2144 94 DoCmd.OpenForm stDocName, , stLinkCriteria
2145 95
2146 96 Exit_Close_Click:
2147 97 Exit Sub
2148 98
2149 99 Err_Close_Click:
2150 100 MsgBox Err.Description
2151 101 Resume Exit_Close_Click
2152 102
2153 103 End Sub
2154 104
2155 105
2156 106 Form: frmSchoolAssignments
2157 107 Code
2158 108 1 Attribute VB_Name = "Form_frmSchoolAssignments"
2159 109 2 Attribute VB_Creatable = True
2160 110 3 Attribute VB_PredeclaredId = True
2161 111 4 Attribute VB_Exposed = False
2162 112 5 Option Compare Database
2163 113 6 Option Explicit
2164 114 7 Private Sub Close_Click()
2165 115 8 On Error GoTo Err_Close_Click
2166 116 9
2167 117 10 Dim stDocName As String
2168 118 11 Dim stLinkCriteria As String
2169 119 12 Dim db1 As Database
2170 120 13
2171 121 14 Set db1 = CurrentDb()
2172 122 15
2173 123 16 ' Close the current form
2174 124 17 DoCmd.Close
2175 125 18 ' Open specified form
2176 126 19 stDocName = "frmAnalyzeResult"
2177 127 20 DoCmd.OpenForm stDocName, , stLinkCriteria
2178 128 21 Format!frmAnalyzeResult.SetFocus
2179 129 22
2180 130 23
2181 131 24 Exit_Close_Click:
2182 132 25 Exit Sub
2183 133 26
2184 134 27 Err_Close_Click:
2185 135 28 MsgBox Err.Description
2186 136 29 Resume Exit_Close_Click
2187 137 30
2188 138 31 End Sub
2189 139 32
2190 140 33 Private Sub Form_Current()
2191 141 34
2192 142 35 Dim strSQL As String
2193 143 36 Dim rec As Recordset
2194 144 37 Dim db1 As Database
2195 145 38
2196 146 39 Set db1 = CurrentDb()
2197 147 40
2198 148 41 ' Calculates the total quota, and number of schools for the run.
2199 149 42 strSQL = "SELECT Sum(SumOfQuota) AS TotalQuota, Count(SumOfQuota) AS
2200 NumberOfSchools FROM qryTotalQuotaForRun;"
2201 150 43 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2202 151 44 Me.tblTotalQuota = rec!TotalQuota
2203 152 45 Me.tblNumberOfSchools = rec!NumberOfSchools
2204 153 46
2205 154 47 ' Calculates the total number of assignments for the run.
2206 155 48 strSQL = "SELECT Sum(CountOfFiscalYear_PK) AS TotalAssigned FROM
2207 qryTotalQuotaFilesForRun;"
2208 156 49 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2209 157 50 Me.tblTotalAssigned = rec!TotalAssigned
2210 158 51

```

```

2211 52 Me.tblTotalPercentFill = Int((Me.tblTotalAssigned / Me.tblTotalQuota)
2212 53
2213 54
2214 55 End Sub
2215 56
2216 57 Form: frmSchools
2217 58 Code
2218 59 1 Attribute VB_Name = "Form_frmSchools"
2219 60 2 Attribute VB_Creatable = True
2220 61 3 Attribute VB_PredeclaredId = True
2221 62 4 Attribute VB_Exposed = False
2222 63 5 Option Compare Database
2223 64 6 Option Explicit
2224 65 7
2225 66 8 Private Sub AMOS_PK_AfterUpdate()
2226 67 9 Dim rec As Recordset
2227 68 10 Dim db1 As Database
2228 69 11 Dim strSQL As String
2229 70 12
2230 71 13 ' Update the drop down list for AMOS
2231 72 14 strSQL = "SELECT TargetMOS_PK FROM
2232 qryListUsedTargetMOSFromTARGET_MOS WHERE TCourseNumber_PK = " &
2233 73 15 Me.AMOS_PK.RowSource = strSQL
2234 74 16 Me.lstTargetMOS_PK.RowSource = strSQL
2235 75 17
2236 76 18 ' Update the AMOS_FK in the child relation SCH_TGT_MOS
2237 77 19 Format!frmSchools!tblFormSchools!AMOS_FK.Value = Me.AMOS_PK.Value
2238 78 20
2239 79 21 ' Update the TargetMOS_FK in the child relation SCH_TGT_MOS
2240 80 22 Set db1 = CurrentDb()
2241 81 23 ' Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2242 82 24 Format!frmSchools!tblFormSchools!TargetMOS_FK.Value = rec!TargetMOS_PK
2243 83 25
2244 84 26 ' Input the correct MCC value for the selected course
2245 85 27 strSQL = "SELECT MCC FROM TARGET_MOS WHERE TCourseNumber_PK = " &
2246 86 28 Me.SCourseNumber_PK.Value & ""
2247 87 29 Set db1 = CurrentDb()
2248 88 30 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
2249 89 31 Me.MCC.Value = rec!MCC
2250 90 32
2251 91 33 ' Update the TCourseNumber_FK in the child relation SCH_TGT_MOS
2252 92 34 Format!frmSchools!tblFormSchools!TCourseNumber_FK.Value =
2253 93 35 Me.SCourseNumber_PK.Value
2254 94 36
2255 95 37 End Sub
2256 96 38 Private Sub btnClear_Click()
2257 97 39
2258 98 40 Dim db1 As Database
2259 99 41 Dim strSQL As String
2260 100 42 Dim i As Integer
2261 101 43 Dim rec As Recordset
2262 102 44
2263 103 45 Set db1 = CurrentDb()
2264 104 46
2265 105 47 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2266 106 48 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2267 107 49 SCH_TGT_MOS.TargetMOS_FK AND TARGET_MOS.TCourseNumber_PK =
2268 108 50 SCH_TGT_MOS.TCourseNumber_FK WHERE (((SCH_TGT_MOS.TargetMOS_FK) Is
2269 109 51 Null) AND ((SCH_TGT_MOS.TCourseNumber_FK) Is Null)) ORDER BY
2270 110 52 TCourseNumber_PK;"
2271 111 53 Set rec = db1.OpenRecordset(strSQL, dbOpenDynaset)
2272 112 54
2273 113 55 If rec.EOF = False Then
2274 114 56 rec.MoveNext
2275 115 57
2276 116 58 ' Clear unused course numbers from the target_mos table
2277 117 59
2278 118 60 If rec.RecordCount > 0 Then
2279 119 61 For i = 1 To rec.RecordCount
2280 120 62 ' Deletes the action query confirmation message
2281 121 63 Application.SetOption "Confirm Action Queries", False
2282 122 64 DoCmd.SetWarnings (False)
2283 123 65
2284 124 66 strSQL = "DELETE * FROM TARGET_MOS WHERE
2285 125 67 TARGET_MOS.TCourseNumber_PK = " & rec!TCourseNumber_PK & ""
2286 126 68 db1.Execute strSQL
2287 127 69
2288 128 70 ' Enables the action query confirmation message
2289 129 71 Application.SetOption "Confirm Action Queries", True
2290 130 72 DoCmd.SetWarnings (True)
2291 131 73 rec.MoveNext
2292 132 74 Next i
2293 133 75 End If
2294 134 76
2295 135 77 ' Update the Course Number List
2296 136 78 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2297 137 79 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2298 138 80 SCH_TGT_MOS.TargetMOS_FK AND TARGET_MOS.TCourseNumber_PK =
2299 139 81 SCH_TGT_MOS.TCourseNumber_FK WHERE (((SCH_TGT_MOS.TargetMOS_FK) Is
2300 140 82 Null) AND ((SCH_TGT_MOS.TCourseNumber_FK) Is Null)) ORDER BY
2301 141 83 TCourseNumber_PK;"
2302 142 84 Me.SCourseNumber_PK.RowSource = strSQL
2303 143 85
2304 144 86 End Sub
2305 145 87 Private Sub btnResetAllPenalties_Click()
2306 146 88
2307 147 89 ' Disables the action query confirmation message
2308 148 90 Application.SetOption "Confirm Action Queries", False
2309 149 91 DoCmd.SetWarnings (False)
2310 150 92
2311 151 93 DoCmd.RunSQL "UPDATE SCHOOL SET PenaltyFactor = 24;"
2312 152 94
2313 153 95 ' Enables the action query confirmation message
2314 154 96 Application.SetOption "Confirm Action Queries", True
2315 155 97 DoCmd.SetWarnings (True)
2316 156 98
2317 157 99 cmbPenaltyView = "t"
2318 158 100
2319 159 101 End Sub
2320 160 102 Private Sub cmbCourseNumberFind_AfterUpdate()
2321 161 103
2322 162 104 Dim R As Recordset
2323 163 105 Set R = Me.RecordsetClone
2324 164 106 R.FindFirst "[CourseNumber_PK] = " & Chr(34) &
2325 165 107 Me!cmbCourseNumberFind & Chr(34)
2326 166 108 Me.Bookmark = R.Bookmark
2327 167 109 Me!cmbCourseNumberFind = Null
2328 168 110 Me.SCourseNumber_PK.SetFocus
2329 169 111
2330 170 112 End Sub
2331 171 113 Private Sub cmbPenaltyView_AfterUpdate()
2332 172 114
2333 173 115
2334 174 116
2335 175 117

```

```

2340 108 Select Case cmbPenaltyView
2341 109 Case "4x"
2342 110   bPenaltyFactor = 4 * 24
2343 111 Case "3x"
2344 112   bPenaltyFactor = 3 * 24
2345 113 Case "2x"
2346 114   bPenaltyFactor = 2 * 24
2347 115 Case "1x"
2348 116   bPenaltyFactor = 1 * 24
2349 117 Case "1/2x"
2350 118   bPenaltyFactor = 24 / 2
2351 119 Case "1/3x"
2352 120   bPenaltyFactor = 24 / 3
2353 121 Case "1/4x"
2354 122   bPenaltyFactor = 24 / 4
2355 123 End Select
2356 124
2357 125 ' Save the update
2358 126 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord,
2359 127
2360 128 End Sub
2361 129
2362 130 Private Sub Command14_Click()
2363 131
2364 132 Dim strSQL As String
2365 133
2366 134 ' Disables the action query confirmation message
2367 135 Application.SetOption "Confirm Action Queries", False
2368 136 DoCmd.SetWarnings (False)
2369 137
2370 138 DoCmd.RunSQL "INSERT INTO TARGET_MOS (TCourseNumber_PK,
2371 139 TargetMOS_PK, MCC) SELECT DISTINCT BNA_EXTRACT.CourseNumber_PK,
2372 140 BNA_EXTRACT.TargetMOS_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT;"
2373 141
2374 142 ' Enables the action query confirmation message
2375 143 Application.SetOption "Confirm Action Queries", True
2376 144 DoCmd.SetWarnings (True)
2377 145
2378 146 ' Update the Course Number List
2379 147 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2380 148 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2381 149 SCH_TGT_MOS.TargetMOS_PK AND TARGET_MOS.TCourseNumber_PK =
2382 150 SCH_TGT_MOS.TCourseNumber_PK WHERE ((SCH_TGT_MOS.TargetMOS_PK) Is
2383 151 Null)AND((SCH_TGT_MOS.TCourseNumber_PK) Is Null)ORDER BY
2384 152 TCourseNumber_PK;"
2385 153 Me.SCourseNumber_PK.RowSource = strSQL
2386 154
2387 155 End Sub
2388 156
2389 157
2390 158 Private Sub Command39_Click()
2391 159 Dim frm As Form, cfl As Control
2392 160 Dim varItem As Variant, int1 As Integer
2393 161 Dim strSQL As String
2394 162
2395 163 ' Ensures the current record is save to the SCHOOL table
2396 164 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord,
2397 165
2398 166 ' This code enters the selected Target MOS's into the SCH_TGT_MOS
2399 167 Set frm = Forms!frmSchools
2400 168 Set cfl = frm!tblTargetMOS_FK
2401 169 For Each varItem In cfl.ItemsSelected
2402 170   For int1 = 0 To cfl.ColumnCount - 1
2403 171     ' This puts the value found in the list box into a text box,
2404 172     making it readable by the query
2405 173     Forms!frmSchools.TargetValue.Value = cfl.Column(int1, varItem)
2406 174   Next int1
2407 175
2408 176 ' Disables the action query confirmation message
2409 177 Application.SetOption "Confirm Action Queries", False
2410 178
2411 179 ' Enters the value into the SCH_TGT_MOS table
2412 180 DoCmd.OpenQuery "qryUpdateSCH_TGT_MOS"
2413 181
2414 182 ' Enables the action query confirmation message
2415 183 Application.SetOption "Confirm Action Queries", True
2416 184
2417 185 Next int1
2418 186 Next varItem
2419 187
2420 188 ' Update the Target MOS list
2421 189 strSQL = "SELECT [TargetMOS_FK] FROM [SCH_TGT_MOS] WHERE
2422 190 [SCourseNumber_PK] = " & Forms!frmSchools.TCourseNumber_PK.Value & " " &
2423 191 " AND AMOS_PK = " & Forms!frmSchools.AMOS_PK.Value & " " &
2424 192 " Form!frmSchools.ListTargetMOS_FK_History.RowSource = strSQL"
2425 193
2426 194 ' Update the Potential Target MOS list
2427 195 strSQL = "SELECT TargetMOS_PK FROM
2428 196 qryListUsedTargetMOSFromTARGET_MOS WHERE TCourseNumber_PK = " &
2429 197 Me.ListTargetMOS_FK.RowSource & strSQL
2430 198
2431 199 ' Update the Course Number List
2432 200 strSQL = "SELECT DISTINCTROW TARGET_MOS.TCourseNumber_PK FROM
2433 201 TARGET_MOS LEFT JOIN SCH_TGT_MOS ON TARGET_MOS.TargetMOS_PK =
2434 202 SCH_TGT_MOS.TargetMOS_PK WHERE ((SCH_TGT_MOS.TargetMOS_PK) Is
2435 203 Null)AND((SCH_TGT_MOS.TCourseNumber_PK) Is Null)ORDER BY
2436 204 Me.SCourseNumber_PK.RowSource = strSQL"
2437 205
2438 206 ' Update time stamp
2439 207 [Sch_TimeStamp] = Now
2440 208
2441 209 End Sub
2442 210
2443 211 Private Sub Command40_Click()
2444 212 Dim frm As Form, cfl As Control
2445 213 Dim varItem As Variant, int1 As Integer
2446 214 Dim strSQL As String
2447 215
2448 216 ' This code Deletes the selected Target MOS's and its associated
2449 217 entries from the SCH_TGT_MOS table
2450 218 Set frm = Forms!frmSchools
2451 219 Set cfl = frm!tblTargetMOS_FK_History
2452 220 For Each varItem In cfl.ItemsSelected
2453 221   For int1 = 0 To cfl.ColumnCount - 1
2454 222     ' This puts the value found in the list box into a text box,
2455 223     making it readable by the query
2456 224     Forms!frmSchools.TargetValue.Value = cfl.Column(int1, varItem)
2457 225   Next int1
2458 226
2459 227 ' Disables the action query confirmation message
2460 228 Application.SetOption "Confirm Action Queries", False
2461 229
2462 230 ' Deletes the record associated with the specified value from
2463 231 the SCH_TGT_MOS table
2464 232 DoCmd.OpenQuery "qryDeleteSCH_TGT_MOS"
2465 233
2466 234 ' Enables the action query confirmation message
2467 235 Application.SetOption "Confirm Action Queries", True
2468 236
2469 237 Next int1

```

```

2600 336 Application.SetOption "Confirm Action Queries", True
2601 337
2602 338 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2603 339 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2604 340
2605 341 Exit_btnDelete_Click
2606 342 Exit Sub
2607 343
2608 344 Err_btnDelete_Click
2609 345 MsgBox Err.Description
2610 346 Resume Exit_btnDelete_Click
2611 347
2612 348 End Sub
2613 349 Private Sub btnClose_Click()
2614 350 On Error GoTo Err_btnClose_Click
2615 351
2616 352
2617 353 DoCmd.Close
2618 354 DoCmd.OpenForm "frmMaintenanceSwitchboard"
2619 355
2620 356 Exit_btnClose_Click:
2621 357 Exit Sub
2622 358
2623 359 Err_btnClose_Click:
2624 360 MsgBox Err.Description
2625 361 Resume Exit_btnClose_Click
2626 362
2627 363 End Sub
2628
2629 Form: frmSchoolToPEF_Maint
2630 Code
2631 1 Attribute VB_Name = "Form_frmSchoolToPEF_Maint"
2632 2 Attribute VB_Creatable = True
2633 3 Attribute VB_PredeclaredId = True
2634 4 Attribute VB_Exposed = False
2635 5 Option Compare Database
2636 6 Option Explicit
2637 7
2638 8 Private Sub btnClose_Click()
2639 9 On Error GoTo Err_btnClose_Click
2640 10
2641 11
2642 12 DoCmd.Close
2643 13 DoCmd.OpenForm "frmMaintenanceSwitchboard"
2644 14
2645 15 Exit_btnClose_Click:
2646 16 Exit Sub
2647 17
2648 18 Err_btnClose_Click:
2649 19 MsgBox Err.Description
2650 20 Resume Exit_btnClose_Click
2651 21
2652 22
2653 23 End Sub
2654 24
2655 25 Private Sub btnDelete_Click()
2656 26 On Error GoTo Err_btnDelete_Click
2657 27
2658 28
2659 29 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2660 30 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2661 31
2662 32 Exit_btnDelete_Click:
2663 33 Exit Sub
2664 34
2665 35 Err_btnDelete_Click:
2666 36 MsgBox Err.Description
2667 37 Resume Exit_btnDelete_Click
2668 38
2669 39
2670 40
2671 41 Private Sub cmbCourseNumberFind_AfterUpdate()
2672 42
2673 43 Dim R As Recordset
2674 44 Set R = Me.RecordsetClone
2675 45 R.FindFirst "[SCourseNumber_FK] = " & Chr(34) &
2676 46 Me[cmbCourseNumberFind] & Chr(34)
2677 47 Me.Bookmark = R.Bookmark
2678 48 Me[cmbCourseNumber_Find] = Null
2679 49 Me.SCourseNumber_FK.SetFocus
2680 50
2681 51 End Sub
2682 52 Private Sub Form_Current()
2683 53 Dim strSQL As String
2684 54
2685 55 * Update the Potential PEF list
2686 56 strSQL = "SELECT DISTINCTROW [PEF].[PEF_FK] FROM PEF LEFT JOIN
2687 57 qrySch_Pel ON [PEF].[PEF_FK] = [qrySch_Pel].[PEF_FK] WHERE
2688 58 ([qrySch_Pel].[PEF_FK] Is Null) ORDER BY [PEF].[PEF_FK]:"
2689 59 Me.ListUnselectedPEF.RowSource = strSQL
2690 60
2691 61 * Update the Selected PEF List
2692 62 strSQL = "SELECT [PEF_FK] FROM qrySch_Pel:"
2693 63 Me.ListSelectedPEF.RowSource = strSQL
2694 64
2695 65 End Sub
2696 66 Private Sub btnLevel0_Click()
2697 67 Dim frm As Form, ctl As Control
2698 68 Dim varIn As Variant, int As Integer
2699 69 Dim strSQL As String
2700 70
2701 71 * Ensures the current record is save to the SCH_TGT_MOS
2702 72 table
2703 73 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, ,
2704 74
2705 75 * This code enters the selected Target MOS's into the SCH_TGT_MOS
2706 76 table
2707 77 Set ctl = frm.ListUnselectedPEF
2708 78 For Each varIn In ctl.ItemsSelected
2709 79 For int = 0 To ctl.ColumnCount - 1
2710 80
2711 81 * This puts the value found in the list box into a text box,
2712 82 making it readable by the query
2713 83 Form!frmSchoolToPEF_Maint!PEFValue.Value = ctl.Column(int),
2714 84
2715 85
2716 86 * Disables the action query confirmation message
2717 87 Application.SetOption "Confirm Action Queries", False
2718 88
2719 89 * Enters the value into the SCH_TGT_MOS table
2720 90 DoCmd.OpenQuery "qryUpdateSCH_PEF"
2721 91
2722 92 * Enables the action query confirmation message
2723 93 Application.SetOption "Confirm Action Queries", True
2724 94
2725 95 Next int
2726 96 Next varIn
2727 97
2728 98 * Update the Potential PEF list
2729 99 strSQL = "SELECT DISTINCTROW [PEF].[PEF_FK] FROM PEF LEFT JOIN

```

```

2730 qrySch_Pel ON [PEF].[PEF_FK] = [qrySch_Pel].[PEF_FK] WHERE
2731 ([qrySch_Pel].[PEF_FK] Is Null) ORDER BY [PEF].[PEF_FK]:"
2732 100 Me.ListUnselectedPEF.RowSource = strSQL
2733 101
2734 102 * Update the Selected PEF List
2735 103 strSQL = "SELECT [PEF_FK] FROM qrySch_Pel:"
2736 104 Me.ListSelectedPEF.RowSource = strSQL
2737 105
2738 106 End Sub
2739 107
2740 108 Private Sub RightArrow_Click()
2741 109 Dim frm As Form, ctl As Control
2742 110 Dim varIn As Variant, int As Integer
2743 111 Dim strSQL As String
2744 112
2745 113 * This code Deletes the selected Target MOS's and its associated
2746 114 entries from the SCH_TGT_MOS table
2747 115 Set frm = Form!frmSchoolToPEF_Maint
2748 116 Set ctl = frm.ListSelectedPEF
2749 117 For Each varIn In ctl.ItemsSelected
2750 118 For int = 0 To ctl.ColumnCount - 1
2751 119
2752 120 * This puts the value found in the list box into a text box,
2753 121 making it readable by the query
2754 122 Form!frmSchoolToPEF_Maint!PEFValue.Value = ctl.Column(int),
2755 123
2756 124
2757 125 * Disables the action query confirmation message
2758 126 Application.SetOption "Confirm Action Queries", False
2759 127
2760 128 * Deletes the record associated with the specified value from
2761 129 the SCH_TGT_MOS table
2762 130 DoCmd.OpenQuery "qryDeleteSCH_PEF"
2763 131
2764 132 * Enables the action query confirmation message
2765 133 Application.SetOption "Confirm Action Queries", True
2766 134
2767 135 Next int
2768 136 Next varIn
2769 137
2770 138 * Update the Potential PEF list
2771 139 strSQL = "SELECT DISTINCTROW [PEF].[PEF_FK] FROM PEF LEFT JOIN
2772 140 ([qrySch_Pel ON [PEF].[PEF_FK] = [qrySch_Pel].[PEF_FK] WHERE
2773 141 ([qrySch_Pel].[PEF_FK] Is Null) ORDER BY [PEF].[PEF_FK]:"
2774 142 Me.ListUnselectedPEF.RowSource = strSQL
2775 143
2776 144 * Update the Selected PEF List
2777 145 strSQL = "SELECT [PEF_FK] FROM qrySch_Pel:"
2778 146 Me.ListSelectedPEF.RowSource = strSQL
2779 147
2780 148 End Sub
2781
2782 Form: frmSchoolToPropertyMaint
2783 Code
2784 1 Attribute VB_Name = "Form_frmSchoolToPropertyMaint"
2785 2 Attribute VB_Creatable = True
2786 3 Attribute VB_PredeclaredId = True
2787 4 Attribute VB_Exposed = False
2788 5 Option Compare Database
2789 6 Option Explicit
2790 7
2791 8 Private Sub btnClose_Click()
2792 9 On Error GoTo Err_btnClose_Click
2793 10
2794 11
2795 12 DoCmd.Close
2796 13 DoCmd.OpenForm "frmMaintenanceSwitchboard"
2797 14
2798 15 Exit_btnClose_Click:
2799 16 Exit Sub
2800 17
2801 18 Err_btnClose_Click:
2802 19 MsgBox Err.Description
2803 20 Resume Exit_btnClose_Click
2804 21
2805 22 End Sub
2806 23 Private Sub btnDelete_Click()
2807 24 On Error GoTo Err_btnDelete_Click
2808 25
2809 26
2810 27 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
2811 28 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
2812 29
2813 30 Exit_btnDelete_Click:
2814 31 Exit Sub
2815 32
2816 33 Err_btnDelete_Click:
2817 34 MsgBox Err.Description
2818 35 Resume Exit_btnDelete_Click
2819 36
2820 37
2821 38 End Sub
2822 39 Private Sub btnLevel0_Click()
2823 40 On Error GoTo Err_btnLevel0_Click
2824 41
2825 42 Dim lngRecordNum As Long
2826 43 Dim strSQL As String
2827 44
2828 45 lngRecordNum = Me.CurrentRecord
2829 46
2830 47 DoCmd.Close
2831 48
2832 49 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2833 50 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2834 51 lngRecordNum
2835 52 Form!frmSchoolToPropLevelMaint!ChosenLevel = 0
2836 53
2837 54 * Update the selected property list
2838 55 strSQL = "SELECT [Property Name_FK] FROM FUND_SCH_PROP WHERE
2839 56 SCourseNumber_FK = " &
2840 57 Form!frmSchoolToPropLevelMaint!SCourseNumber_FK.Value & "" & " And
2841 58 AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_FK.Value & "" & "
2842 59 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2843 60 UNION SELECT [Property Name_FK] FROM LOG_SCH_PROP WHERE SCourseNumber_FK
2844 61 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_FK.Value & "" & "
2845 62 And AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_FK.Value & "" & "
2846 63 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2847 64 Form!frmSchoolToPropLevelMaint!Level0.RowSource = strSQL
2848 65
2849 66 End Sub
2850 67 Exit_btnLevel0_Click:
2851 68 Exit Sub
2852 69
2853 70 Err_btnLevel0_Click:
2854 71 MsgBox Err.Description
2855 72 Resume Exit_btnLevel0_Click
2856 73
2857 74
2858 75 End Sub
2859 76

```

```

2860 69 Private Sub btnLevel1_Click()
2861 70 On Error GoTo Err_btnLevel1_Click
2862 71
2863 72 Dim lngRecordNum As Long
2864 73 Dim strSQL As String
2865 74
2866 75 lngRecordNum = Me.CurrentRecord
2867 76
2868 77 DoCmd.Close
2869 78
2870 79 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2871 80 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2872 lngRecordNum
2873 81 Form!frmSchoolToPropLevelMaint!ChosenLevel = 1
2874 82
2875 83 ' Update the selected property list
2876 84 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
2877 S!CourseNumber_PK = " &
2878 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & " And
2879 AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2880 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2881 UNION SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK
2882 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & "
2883 And AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2884 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2885 Form!frmSchoolToPropLevelMaint!Level0.RowSource = strSQL
2886 85
2887 86 Exit_btnLevel1_Click
2888 87 Exit Sub
2889 88
2890 89 Err_btnLevel1_Click
2891 90 MsgBox Err.Description
2892 91 Resume Err_btnLevel1_Click
2893 92
2894 93 End Sub
2895 94
2896 95 Private Sub btnLevel2_Click()
2897 96 On Error GoTo Err_btnLevel2_Click
2898 97
2899 98 Dim lngRecordNum As Long
2900 99 Dim strSQL As String
2901 100
2902 101 lngRecordNum = Me.CurrentRecord
2903 102
2904 103 DoCmd.Close
2905 104
2906 105 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2907 106 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2908 lngRecordNum
2909 107 Form!frmSchoolToPropLevelMaint!ChosenLevel = 2
2910 108
2911 109 ' Update the selected property list
2912 110 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
2913 S!CourseNumber_PK = " &
2914 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & " And
2915 AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2916 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2917 UNION SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK
2918 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & "
2919 And AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2920 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2921 Form!frmSchoolToPropLevelMaint!Level0.RowSource = strSQL
2922 111
2923 112 Exit_btnLevel2_Click
2924 113 Exit Sub
2925 114
2926 115 Err_btnLevel2_Click
2927 116 MsgBox Err.Description
2928 117 Resume Err_btnLevel2_Click
2929 118
2930 119 End Sub
2931 120
2932 121 Private Sub btnLevel3_Click()
2933 122 On Error GoTo Err_btnLevel3_Click
2934 123
2935 124 Dim lngRecordNum As Long
2936 125 Dim strSQL As String
2937 126
2938 127 lngRecordNum = Me.CurrentRecord
2939 128
2940 129 DoCmd.Close
2941 130
2942 131 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2943 132 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2944 lngRecordNum
2945 133 Form!frmSchoolToPropLevelMaint!ChosenLevel = 3
2946 134
2947 135 ' Update the selected property list
2948 136 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
2949 S!CourseNumber_PK = " &
2950 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & " And
2951 AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2952 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2953 UNION SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK
2954 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & "
2955 And AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2956 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2957 Form!frmSchoolToPropLevelMaint!Level0.RowSource = strSQL
2958 137
2959 138 Exit_btnLevel3_Click
2960 139 Exit Sub
2961 140
2962 141 Err_btnLevel3_Click
2963 142 MsgBox Err.Description
2964 143 Resume Err_btnLevel3_Click
2965 144
2966 145 End Sub
2967 146
2968 147 Private Sub btnLevel4_Click()
2969 148 On Error GoTo Err_btnLevel4_Click
2970 149
2971 150 Dim lngRecordNum As Long
2972 151 Dim strSQL As String
2973 152
2974 153 lngRecordNum = Me.CurrentRecord
2975 154
2976 155 DoCmd.Close
2977 156
2978 157 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
2979 158 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
2980 lngRecordNum
2981 159 Form!frmSchoolToPropLevelMaint!ChosenLevel = 4
2982 160
2983 161 ' Update the selected property list
2984 162 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
2985 S!CourseNumber_PK = " &
2986 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & " And
2987 AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2988 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2989 UNION SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK
2990 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & "
2991 And AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
2992 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
2993 Form!frmSchoolToPropLevelMaint!Level0.RowSource = strSQL
2994 163
2995 164 Exit_btnLevel4_Click
2996 165 Exit Sub
2997 166
2998 167 Err_btnLevel4_Click
2999 168 MsgBox Err.Description
3000 169 Resume Err_btnLevel4_Click
3001 170
3002 171 End Sub
3003 172
3004 173 Private Sub btnLevel5_Click()
3005 174 On Error GoTo Err_btnLevel5_Click
3006 175
3007 176 Dim lngRecordNum As Long
3008 177 Dim strSQL As String
3009 178
3010 179 lngRecordNum = Me.CurrentRecord
3011 180
3012 181 DoCmd.Close
3013 182
3014 183 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
3015 184 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
3016 lngRecordNum
3017 185 Form!frmSchoolToPropLevelMaint!ChosenLevel = 5
3018 186
3019 187 ' Update the selected property list
3020 188 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
3021 S!CourseNumber_PK = " &
3022 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & " And
3023 AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
3024 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
3025 UNION SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK
3026 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & "
3027 And AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
3028 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
3029 Form!frmSchoolToPropLevelMaint!Level0.RowSource = strSQL
3030 189
3031 190 Exit_btnLevel5_Click
3032 191 Exit Sub
3033 192
3034 193 Err_btnLevel5_Click
3035 194 MsgBox Err.Description
3036 195 Resume Err_btnLevel5_Click
3037 196
3038 197 End Sub
3039 198
3040 199 Private Sub btnLevel6_Click()
3041 200 On Error GoTo Err_btnLevel6_Click
3042 201
3043 202 Dim lngRecordNum As Long
3044 203 Dim strSQL As String
3045 204
3046 205 lngRecordNum = Me.CurrentRecord
3047 206
3048 207 DoCmd.Close
3049 208
3050 209 DoCmd.OpenForm "frmSchoolToPropLevelMaint"
3051 210 DoCmd.GoToRecord acDataForm, "frmSchoolToPropLevelMaint", acGoTo,
3052 lngRecordNum
3053 211 Form!frmSchoolToPropLevelMaint!ChosenLevel = 6
3054 212
3055 213 ' Update the selected property list
3056 214 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
3057 S!CourseNumber_PK = " &
3058 Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & " And
3059 AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
3060 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
3061 UNION SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK
3062 = " & Form!frmSchoolToPropLevelMaint!SCourseNumber_PK.Value & "" & "
3063 And AMOS_FK = " & Form!frmSchoolToPropLevelMaint!AMOS_PK.Value & "" & "
3064 And Level = " & Form!frmSchoolToPropLevelMaint!ChosenLevel & "" & "
3065 Form!frmSchoolToPropLevelMaint!Level0.RowSource = strSQL
3066 215
3067 216 Exit_btnLevel6_Click
3068 217 Exit Sub
3069 218
3070 219 Err_btnLevel6_Click
3071 220 MsgBox Err.Description
3072 221 Resume Err_btnLevel6_Click
3073 222
3074 223 End Sub
3075 224
3076 225 Private Sub cmbCourseNumberFind_AfterUpdate()
3077 226 Dim R As Recordset
3078 227 Set R = Me.RecordsetClone
3079 228 R.FindFirst "S!CourseNumber_PK = " & Chr(34) &
3080 Me!cmbCourseNumberFind & Chr(34)
3081 229 Me!cmbCourseNumberFind.Null
3082 230 Me!S!CourseNumber_PK.SetFocus
3083 231
3084 232 End Sub
3085 233
3086 234 Private Sub Form_Current()
3087 235 Dim strSQL, strInput1, strInput2 As String
3088 236 Dim lngLevel As Integer
3089 237
3090 238 ' Update the Level 0 list
3091 239 lngLevel = 0
3092 240 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
3093 S!CourseNumber_PK = " & Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK =
3094 " & Me!AMOS_PK.Value & "" & " And Level = " & lngLevel & "" & " UNION
3095 SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK = " &
3096 Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me!AMOS_PK.Value
3097 & "" & " And Level = " & lngLevel & "" & "
3098 241 Me!Level0.RowSource = strSQL
3099 242
3100 243 ' Update the Level 1 list
3101 244 lngLevel = 1
3102 245 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
3103 S!CourseNumber_PK = " & Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK =
3104 " & Me!AMOS_PK.Value & "" & " And Level = " & lngLevel & "" & " UNION
3105 SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK = " &
3106 Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me!AMOS_PK.Value
3107 & "" & " And Level = " & lngLevel & "" & "
3108 246 Me!Level1.RowSource = strSQL
3109 247
3110 248 ' Update the Level 2 list
3111 249 lngLevel = 2
3112 250 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
3113 S!CourseNumber_PK = " & Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK =
3114 " & Me!AMOS_PK.Value & "" & " And Level = " & lngLevel & "" & " UNION
3115 SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK = " &
3116 Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me!AMOS_PK.Value
3117 & "" & " And Level = " & lngLevel & "" & "
3118 251 Me!Level2.RowSource = strSQL
3119 252
3120 253 ' Update the Level 3 list
3121 254 lngLevel = 3
3122 255 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
3123 S!CourseNumber_PK = " & Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK =
3124 " & Me!AMOS_PK.Value & "" & " And Level = " & lngLevel & "" & " UNION
3125 SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK = " &
3126 Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me!AMOS_PK.Value
3127 & "" & " And Level = " & lngLevel & "" & "
3128 256 Me!Level3.RowSource = strSQL
3129 257
3130 258 ' Update the Level 4 list
3131 259 lngLevel = 4
3132 260 strSQL = "SELECT (F!PropertyName_RK) FROM FUND_SCH_PROP WHERE
3133 S!CourseNumber_PK = " & Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK =
3134 " & Me!AMOS_PK.Value & "" & " And Level = " & lngLevel & "" & " UNION
3135 SELECT (L!PropertyName_RK) FROM LOG_SCH_PROP WHERE S!CourseNumber_FK = " &
3136 Me!S!CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me!AMOS_PK.Value
3137 & "" & " And Level = " & lngLevel & "" & "
3138 261 Me!Level4.RowSource = strSQL
3139 262

```

```

3120 & "" & " And Level = " & intLevel & ""
3121 Me.lstLevel2.RowSource = strSQL
3122 261
3123 intLevel = 3
3124 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3125 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3126 " & Me.AMOS_PK.Value & "" & " And Level = " & intLevel & "" & " UNION
3127 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE S.CourseNumber_FK = " &
3128 Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3129 & "" & " And Level = " & intLevel & ""
3130 Me.lstLevel3.RowSource = strSQL
3131 265
3132 intLevel = 4
3133 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3134 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3135 " & Me.AMOS_PK.Value & "" & " And Level = " & intLevel & "" & " UNION
3136 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE S.CourseNumber_FK = " &
3137 Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3138 & "" & " And Level = " & intLevel & ""
3139 Me.lstLevel4.RowSource = strSQL
3140 269
3141 intLevel = 5
3142 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3143 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3144 " & Me.AMOS_PK.Value & "" & " And Level = " & intLevel & "" & " UNION
3145 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE S.CourseNumber_FK = " &
3146 Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3147 & "" & " And Level = " & intLevel & ""
3148 Me.lstLevel5.RowSource = strSQL
3149 273
3150 intLevel = 6
3151 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3152 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3153 " & Me.AMOS_PK.Value & "" & " And Level = " & intLevel & "" & " UNION
3154 SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE S.CourseNumber_FK = " &
3155 Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK = " & Me.AMOS_PK.Value
3156 & "" & " And Level = " & intLevel & ""
3157 Me.lstLevel6.RowSource = strSQL
3158 277
3159 278
3160 End Sub
3161
3162 Form: frmSchoolToPropLevelMaint
3163 Code
3164 1 Attribute VB_Name = "Form_frmSchoolToPropLevelMaint"
3165 2 Attribute VB_Creatable = True
3166 3 Attribute VB_PredeclaredId = True
3167 4 Attribute VB_Exposed = False
3168 5 Option Compare Database
3169 6 Option Explicit
3170 7
3171 8 Private Sub btnClose_Click()
3172 9 On Error GoTo Err_btnClose_Click
3173 10
3174 11 Dim lngRecordNum As Long
3175 12
3176 13 lngRecordNum = Me.CurrentRecord
3177 14
3178 15 DoCmd.Close
3179 16
3180 17 DoCmd.OpenForm "frmSchoolToPropertyMaint"
3181 18 DoCmd.GoToRecord acDataForm, "frmSchoolToPropertyMaint", acGoTo,
3182 lngRecordNum
3183 19
3184 20 Exit_btnClose_Click
3185 21 Exit Sub
3186 22
3187 23 Err_btnClose_Click
3188 24 MsgBox Err.Description
3189 25 Resume Err_btnClose_Click
3190 26
3191 27 End Sub
3192 28
3193 29 Private Sub btnDelete_Click()
3194 30 On Error GoTo Err_btnDelete_Click
3195 31
3196 32
3197 33 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
3198 34 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
3199 35
3200 36 Exit_btnDelete_Click
3201 37 Exit Sub
3202 38
3203 39 Err_btnDelete_Click
3204 40 MsgBox Err.Description
3205 41 Resume Exit_btnDelete_Click
3206 42
3207 43 End Sub
3208 44
3209 45 Private Sub ChosenLevel_AfterUpdate()
3210 46
3211 47 Dim strSQL As String
3212 48
3213 49 ' Update the potential property list
3214 50 strSQL = "SELECT DISTINCTROW ([qryUnionAllProperty].[PropertyName_PK]
3215 FROM qryUnionAllProperty LEFT JOIN qryUnionPropertyByLevel ON
3216 [qryUnionPropertyByLevel].[PropertyName_PK] =
3217 [qryUnionPropertyByLevel].[PropertyName_FK] WHERE
3218 ([qryUnionPropertyByLevel].[PropertyName_FK] Is Null);"
3219 51 Me.lstLevelUnUsed.RowSource = strSQL
3220 52
3221 53 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3222 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3223 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3224 & " UNION SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE
3225 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3226 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3227
3228 54 Me.lstLevel0.RowSource = strSQL
3229 55
3230 56 End Sub
3231 57
3232 58 Private Sub cmbCourseNumberFind_AfterUpdate()
3233 59
3234 60 Dim R As Recordset
3235 61 Set R = Me.RecordsetClone
3236 62 R.FindFirst "[S.CourseNumber_PK] = " & Chr(34) &
3237 Me[cmbCourseNumberFind] & Chr(34)
3238 63 Me.Bookmark = R.Bookmark
3239 64 Me[cmbCourseNumberFind] = Null
3240 65 Me.S.CourseNumber_PK.SetFocus
3241 66
3242 67 End Sub
3243 68
3244 69 Private Sub Form_Current()
3245 70
3246 71 Dim strSQL As String
3247 72
3248 73 ' Update the potential property list
3249 74 strSQL = "SELECT DISTINCTROW ([qryUnionAllProperty].[PropertyName_PK]

```

```

3250 FROM qryUnionAllProperty LEFT JOIN qryUnionPropertyByLevel ON
3251 [qryUnionAllProperty].[PropertyName_PK] =
3252 [qryUnionPropertyByLevel].[PropertyName_FK] WHERE
3253 ([qryUnionPropertyByLevel].[PropertyName_FK] Is Null);"
3254 75 Me.lstLevelUnUsed.RowSource = strSQL
3255 76
3256 77 ' Update the selected property list
3257 78 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3258 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3259 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3260 & " UNION SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE
3261 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3262 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3263
3264 79
3265 80 End Sub
3266 81
3267 82
3268 83 Private Sub LetArrow_Click()
3269 84 Dim frm As Form, c1 As Control
3270 85 Dim varItem As Variant, int1 As Integer
3271 86 Dim strSQL As String
3272 87
3273 88 ' Ensures the current record is save to the SCHOOL table
3274 89 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, ,
3275 90
3276 91 ' This code enters the selected Target MOS's into the SCH_TGT_MOS
3277 92 Set frm = Form!frmSchoolToPropLevelMaint
3278 93 Set c1 = frm.lstLevelUnUsed
3279 94 For Each varItem In c1.ItemsSelected
3280 95 For int1 = 0 To c1.ColumnCount - 1
3281 96
3282 97 ' This puts the value found in the list box into a text box
3283 98 making it readable by the query
3284 99 Form!frmSchoolToPropLevelMaint.TargetValue.Value =
3285 c1.Column(int1, varItem)
3286 99
3287 100 ' Disables the action query confirmation message
3288 101 Application.SetOption "Confirm Action Queries", False
3289 102 DoCmd.SetWarnings (False)
3290 103
3291 104 ' Enters the value into the SCH_TGT_MOS table
3292 105 DoCmd.OpenQuery "qryUpdateFUND_SCH_PROP"
3293 106 DoCmd.OpenQuery "qryUpdateLOG_SCH_PROP"
3294 107
3295 108 ' Enables the action query confirmation message
3296 109 Application.SetOption "Confirm Action Queries", True
3297 110 DoCmd.SetWarnings (True)
3298 111
3299 112 Next int1
3300 113 Next varItem
3301 114
3302 115 ' Update the potential property list
3303 116 strSQL = "SELECT DISTINCTROW ([qryUnionAllProperty].[PropertyName_PK]
3304 FROM qryUnionAllProperty LEFT JOIN qryUnionPropertyByLevel ON
3305 [qryUnionAllProperty].[PropertyName_PK] =
3306 [qryUnionPropertyByLevel].[PropertyName_FK] WHERE
3307 ([qryUnionPropertyByLevel].[PropertyName_FK] Is Null);"
3308 117 Me.lstLevelUnUsed.RowSource = strSQL
3309 118
3310 119 ' Update the selected property list
3311 120 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3312 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3313 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3314 & " UNION SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE
3315 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3316 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3317
3318 121
3319 122 End Sub
3320 123
3321 124 Private Sub RightArrow_Click()
3322 125 Dim frm As Form, c1 As Control
3323 126 Dim varItem As Variant, int1 As Integer
3324 127 Dim strSQL As String
3325 128
3326 129 ' This code Deletes the selected Target MOS's and its associated
3327 130 entries from the SCH_TGT_MOS table
3328 131 Set frm = Form!frmSchoolToPropLevelMaint
3329 132 Set c1 = frm.lstLevelUnUsed
3330 133 For Each varItem In c1.ItemsSelected
3331 134 For int1 = 0 To c1.ColumnCount - 1
3332 135
3333 136 ' This puts the value found in the list box into a text box
3334 137 making it readable by the query
3335 138 Form!frmSchoolToPropLevelMaint.TargetValue.Value =
3336 c1.Column(int1, varItem)
3337 137
3338 138 ' Disables the action query confirmation message
3339 139 Application.SetOption "Confirm Action Queries", False
3340 140 DoCmd.SetWarnings (False)
3341 141
3342 142 ' Deletes the record associated with the specified value from
3343 143 the FUND or LOG_SCH_PROP table
3344 144 DoCmd.OpenQuery "qryDeleteFUND_SCH_PROP"
3345 145 DoCmd.OpenQuery "qryDeleteLOG_SCH_PROP"
3346 146
3347 147 ' Enables the action query confirmation message
3348 148 Application.SetOption "Confirm Action Queries", True
3349 149 DoCmd.SetWarnings (True)
3350 149
3351 150 Next int1
3352 151 Next varItem
3353 152
3354 153 ' Update the potential property list
3355 154 strSQL = "SELECT DISTINCTROW ([qryUnionAllProperty].[PropertyName_PK]
3356 FROM qryUnionAllProperty LEFT JOIN qryUnionPropertyByLevel ON
3357 [qryUnionAllProperty].[PropertyName_PK] =
3358 [qryUnionPropertyByLevel].[PropertyName_FK] WHERE
3359 ([qryUnionPropertyByLevel].[PropertyName_FK] Is Null);"
3360 155 Me.lstLevelUnUsed.RowSource = strSQL
3361 156
3362 157 ' Update the selected property list
3363 158 strSQL = "SELECT [PropertyName_FK] FROM FUND_SCH_PROP WHERE
3364 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3365 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3366 & " UNION SELECT [PropertyName_FK] FROM LOG_SCH_PROP WHERE
3367 S.CourseNumber_FK = " & Me.S.CourseNumber_PK.Value & "" & " And AMOS_FK =
3368 " & Me.AMOS_PK.Value & "" & " And Level = " & Me.ChosenLevel.Value &
3369
3370 159
3371 160 End Sub
3372 161
3373 162
3374 Form: frmScrubMarine
3375 Code
3376 1 Attribute VB_Name = "Form_frmScrubMarine"
3377 2 Attribute VB_Creatable = True
3378 3 Attribute VB_PredeclaredId = True
3379 4 Attribute VB_Exposed = False
3380 5 Option Compare Database

```

```

3380 6 Option Explicit
3381 7
3382 8 Private Sub Close_Click()
3383 9 On Error GoTo Err_Close_Click
3384 10
3385 11 Dim strDocName As String
3386 12 Dim strLinkCriteria As String
3387 13
3388 14 ' Close current form
3389 15 DoCmd.Close
3390 16
3391 17 ' Open specified form
3392 18 strDocName = "frmPreprocessing&ExecutionSwitchboard"
3393 19 DoCmd.OpenForm strDocName, , , strLinkCriteria
3394 20
3395 21 Exit_Close_Click
3396 22 Exit Sub
3397 23
3398 24 Err_Close_Click
3399 25 MsgBox Err.Description
3400 26 Resume Exit_Close_Click
3401 27
3402 28 End Sub
3403 29
3404 30 Private Sub cmbPefFind_AfterUpdate()
3405 31
3406 32 Dim R As Recordset
3407 33 Set R = Me.RecordsetClone
3408 34 R.FindFirst "[PEF] = " & Chr(34) & Me[cmbPefFind] & Chr(34)
3409 35 Me.Bookmark = R.Bookmark
3410 36 Me[cmbPefFind] = Null
3411 37 Me.SSN_PK.SetFocus
3412 38
3413 39 End Sub
3414 40
3415 41 Private Sub cmbSSNFind_AfterUpdate()
3416 42
3417 43 Dim R As Recordset
3418 44 Set R = Me.RecordsetClone
3419 45 R.FindFirst "[SSN_PK] = " & Chr(34) & Me[cmbSSNFind] & Chr(34)
3420 46 Me.Bookmark = R.Bookmark
3421 47 Me[cmbSSNFind] = Null
3422 48 Me.Close.SetFocus
3423 49
3424 50 End Sub
3425 51
3426 52 Private Sub Delete_Click()
3427 53 On Error GoTo Err_Delete_Click
3428 54
3429 55
3430 56 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
3431 57 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
3432 58
3433 59 Exit_Delete_Click
3434 60 Exit Sub
3435 61
3436 62 Err_Delete_Click
3437 63 MsgBox Err.Description
3438 64 Resume Exit_Delete_Click
3439 65
3440 66 End Sub
3441 67
3442 68 Private Sub Form_Current()
3443 69 Dim strSQL As String
3444 70
3445 71 ' Update the PEF List
3446 72 strSQL = "SELECT DISTINCTROW [MARINE] [PEF] FROM MARINE LEFT JOIN PEF
3447 73 ON [MARINE].[PEF] = [PEF].[PEF_PK] WHERE ([PEF].[PEF_PK] Is Null) ORDER BY
3448 74 [Me.cmbPefFind].RowSource = strSQL
3449 75 End Sub
3450 76
3451 77
3452 Form: frmSpecialAssignment
3453 Code
3454 1 Attribute VB_Name = "Form_frmSpecialAssignment"
3455 2 Attribute VB_Creatable = True
3456 3 Attribute VB_PredeclaredId = True
3457 4 Attribute VB_Exposed = False
3458 5 Option Compare Database
3459 6 Option Explicit
3460 7
3461 8 Private Sub cmbClassNumber_AfterUpdate()
3462 9
3463 10 Dim strConvert As String
3464 11
3465 12 Me.txtReportDate = Me.cmbClassNumber.Column(2)
3466 13 Me.txtMCC = Me.cmbClassNumber.Column(4)
3467 14
3468 15 ' Convert the fiscal year to a two digit number for RD3 File
3469 16 strConvert = CSI(Me.cmbClassNumber.Column(3))
3470 17 strConvert = Right(strConvert, 2)
3471 18 Me.txtFY = CInt(strConvert)
3472 19
3473 20 Me.txtAssignmentType = "S"
3474 21
3475 22 End Sub
3476 23
3477 24 Private Sub cmbClassNumber_Enter()
3478 25
3479 26 Dim strSQL As String
3480 27
3481 28 ' This query finds the class numbers, report dates and class
3482 29 converting dates associated with the chosen school
3483 30 strSQL = "SELECT BNA_EXTRACT.ClassNumber_PK,
3484 31 BNA_EXTRACT.ClassConvDate, BNA_EXTRACT.ReportDate,
3485 32 BNA_EXTRACT.FiscalYear_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT INNER JOIN
3486 33 SCH_TGT_MOS ON (BNA_EXTRACT.CourseNumber_PK =
3487 34 SCH_TGT_MOS.CourseNumber_PK) AND (BNA_EXTRACT.TargetMOS_PK =
3488 35 SCH_TGT_MOS.TargetMOS_PK) WHERE BNA_EXTRACT.CourseNumber_PK =
3489 36 Forms!frmSpecialAssignment.txtCourseNumber AND SCH_TGT_MOS.AMOS_PK =
3490 37 Forms!frmSpecialAssignment.txtAMOS;"
3491 38 Me.cmbClassNumber.RowSource = strSQL
3492 39
3493 40 End Sub
3494 41
3495 42 Private Sub cmbCourseNumber_AfterUpdate()
3496 43
3497 44 Me.txtAMOS = Me.cmbCourseNumber.Column(1)
3498 45
3499 46 End Sub
3500 47
3501 48 Private Sub cmbSSN_AfterUpdate()
3502 49
3503 50 Me.txtSQL = Me.cmbSSN.Column(1)
3504 51 Me.txtDatedDate = Me.cmbSSN.Column(2)
3505 52
3506 53 End Sub
3507 54
3508 55 Private Sub cmbSSN_Enter()
3509 56

```

```

3510 57 Dim strSQL As String
3511 58
3512 59 strSQL = "SELECT DISTINCTROW MARINE.SSN_PK, MARINE.SCI,
3513 60 MARINE.Grading FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK =
3514 61 ASSIGNMENT.SSN_PK WHERE ((ASSIGNMENT.SSN_PK) Is Null);"
3515 62 Me.cmbSSN.RowSource = strSQL
3516 63
3517 64 End Sub
3518 65
3519 66 Private Sub cmbSSNFind_AfterUpdate()
3520 67
3521 68 Dim R As Recordset
3522 69 Set R = Me.RecordsetClone
3523 70 R.FindFirst "[SSN_PK] = " & Chr(34) & Me[cmbSSNFind] & Chr(34)
3524 71 Me.Bookmark = R.Bookmark
3525 72 Me[cmbSSNFind] = Null
3526 73 Me.Close.SetFocus
3527 74
3528 75 End Sub
3529 76
3530 77 Private Sub Delete_Click()
3531 78 On Error GoTo Err_Delete_Click
3532 79
3533 80
3534 81 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
3535 82 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
3536 83
3537 84 Exit_Delete_Click
3538 85 Exit Sub
3539 86
3540 87 Err_Delete_Click
3541 88 MsgBox Err.Description
3542 89 Resume Exit_Delete_Click
3543 90
3544 91 End Sub
3545 92
3546 93 Private Sub Close_Click()
3547 94 On Error GoTo Err_Close_Click
3548 95
3549 96
3550 97 Dim strDocName As String
3551 98 Dim strLinkCriteria As String
3552 99
3553 100 ' Close current form
3554 101 DoCmd.Close
3555 102
3556 103 ' Open specified form
3557 104 strDocName = "frmPreprocessing&ExecutionSwitchboard"
3558 105 DoCmd.OpenForm strDocName, , , strLinkCriteria
3559 106
3560 107 Exit_Close_Click
3561 108 Exit Sub
3562 109
3563 110 Err_Close_Click
3564 111 MsgBox Err.Description
3565 112 Resume Exit_Close_Click
3566 113
3567 114 End Sub
3568 115 Private Sub Form_Current()
3569 116
3570 117 ' Prevents user from accidentally making special assignments to
3571 118 marines assigned normally
3572 119 If IsNull(Me.cmbSSN.Value) = False Then
3573 120 Me.cmbSSN.Enabled = False
3574 121 Me.cmbCourseNumber.Enabled = False
3575 122 Me.cmbClassNumber.Enabled = False
3576 123
3577 124 ' Allows the user to enter values for the fundamental equation
3578 125 Else
3579 126 Me.cmbSSN.Enabled = True
3580 127 Me.cmbCourseNumber.Enabled = True
3581 128 Me.cmbClassNumber.Enabled = True
3582 129
3583 130 End If
3584 131
3585 132 End Sub
3586 133
3587 134 Private Sub Form_Open(Cancel As Integer)
3588 135
3589 136 DoCmd.GoToRecord acDataForm, "frmSpecialAssignment", acNewRec
3590 137
3591 138 End Sub
3592 139
3593 140
3594 141 Form: frmSpecialAssignmentOfUnassignedMarine
3595 142 Code
3596 143
3597 144 Attribute VB_Name = "Form_frmSpecialAssignmentOfUnassignedMarine"
3598 145 Attribute VB_Creatable = True
3599 146 Attribute VB_PredeclaredId = True
3600 147 Attribute VB_Exposed = False
3601 148 Option Compare Database
3602 149 Option Explicit
3603 150
3604 151 Private Sub cmbClassNumber_AfterUpdate()
3605 152
3606 153 Dim strConvert As String
3607 154 Dim strSSN As String
3608 155 Dim R As Recordset
3609 156
3610 157 Me.txtReportDate = Me.cmbClassNumber.Column(2)
3611 158 Me.txtMCC = Me.cmbClassNumber.Column(4)
3612 159
3613 160 ' Convert the fiscal year to a two digit number for RD3 File
3614 161 strConvert = CSI(Me.cmbClassNumber.Column(3))
3615 162 strConvert = Right(strConvert, 2)
3616 163 Me.txtFY = CInt(strConvert)
3617 164
3618 165 Me.txtAssignmentType = "S"
3619 166
3620 167 End Sub
3621 168
3622 169 ' Keep track of the current SSN
3623 170 strSSN = Me.cmbSSN
3624 171
3625 172 Me.txtAMOS = Me.cmbCourseNumber.Column(1)
3626 173
3627 174 ' Updates the current form and subform
3628 175 Me.Requery
3629 176 Me.Repaint
3630 177
3631 178 ' Moves the record back to the current SSN
3632 179 Set R = Me.RecordsetClone
3633 180 R.FindFirst "[SSN_PK] = " & Chr(34) & strSSN & Chr(34)
3634 181 Me.Bookmark = R.Bookmark
3635 182
3636 183 End Sub
3637 184
3638 185 Private Sub cmbClassNumber_Enter()
3639 186
3640 187 Dim strSQL As String
3641 188
3642 189

```

```

3640 44 ' This query finds the class numbers, report dates and class
3641 converting dates associated with the chosen school
3642 45 strSQL = "SELECT BNA_EXTRACT.ClassNumber_PK,
3643 BNA_EXTRACT.ClassConDate, BNA_EXTRACT.ReportDate,
3644 BNA_EXTRACT.FiscalYear_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT INNER JOIN
3645 SCH_TGT_MCS ON (BNA_EXTRACT.CourseNumber_PK =
3646 SCH_TGT_MCS.CourseNumber_PK) AND (BNA_EXTRACT.TargetMOS_PK =
3647 SCH_TGT_MCS.TargetMOS_PK) WHERE BNA_EXTRACT.CourseNumber_PK =
3648 Format(mSpecialAssignmentOfUnassignedMarine)mcbCourseNumber AND
3649 SCH_TGT_MCS.AMOS_FK =
3650 Me.mcbClassNumber.RowSource = strSQL
3651 47
3652 48 End Sub
3653 49
3654 50 Private Sub cmbCourseNumber_AfterUpdate()
3655 51
3656 52 Dim strSQL As String
3657 53 Dim R As Recordset
3658 54
3659 55 strSQL = Me.cmbSSN
3660 56
3661 57 ' Keep track of the current SSN
3662 58 strSQL = Me.cmbSSN
3663 59
3664 60 Me.tblAMOS = Me.cmbCourseNumber.Column(1)
3665 61
3666 62 ' Updates the current form and subform
3667 63 Me.Refresh
3668 64 Me.Repaint
3669 65
3670 66 ' Moves the record back to the current SSN
3671 67 Set R = Me.RecordsetClone
3672 68 R.FindFirst "[SSN_FK] = " & Chr(34) & strSQL & Chr(34)
3673 69 Me.Bookmark = R.Bookmark
3674 70
3675 71 End Sub
3676 72
3677 73 Private Sub cmbSSN_AfterUpdate()
3678 74
3679 75 Me.tblSQL = Me.cmbSSN.Column(1)
3680 76 Me.tblGradeDate = Me.cmbSSN.Column(2)
3681 77
3682 78 End Sub
3683 79
3684 80 Private Sub cmbSSN_Enter()
3685 81
3686 82 Dim strSQL As String
3687 83
3688 84 strSQL = "SELECT DISTINCTROW MARINE.SSN_PK, MARINE.SOI,
3689 MARINE.GradeDate FROM MARINE LEFT JOIN ASSIGNMENT ON MARINE.SSN_PK =
3690 ASSIGNMENT.SSN_FK WHERE ((ASSIGNMENT.SSN_FK) Is Null);"
3691 85 Me.cmbSSN.RowSource = strSQL
3692 86
3693 87 End Sub
3694 88
3695 89 Private Sub cmbSSNFind_AfterUpdate()
3696 90
3697 91 Dim R As Recordset
3698 92 Set R = Me.RecordsetClone
3699 93 R.FindFirst "[SSN_FK] = " & Chr(34) & Me[cmbSSNFind] & Chr(34)
3700 94 Me.Bookmark = R.Bookmark
3701 95 Me[cmbSSNFind] = Null
3702 96 Me.Close.SetFocus
3703 97
3704 98 End Sub
3705 99
3706 100 Private Sub Delete_Click()
3707 101 On Error GoTo Err_Delete_Click
3708 102
3709 103
3710 104 DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70
3711 105 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70
3712 106
3713 107 Exit_Delete_Click
3714 108 Exit Sub
3715 109
3716 110 Err_Delete_Click
3717 111 MsgBox Err.Description
3718 112 Resume Exit_Delete_Click
3719 113
3720 114 End Sub
3721 115
3722 116 Private Sub Close_Click()
3723 117 On Error GoTo Err_Close_Click
3724 118
3725 119 ' Close current form
3726 120 DoCmd.Close
3727 121
3728 122 Exit_Close_Click
3729 123 Exit Sub
3730 124
3731 125 Err_Close_Click
3732 126 MsgBox Err.Description
3733 127 Resume Exit_Close_Click
3734 128
3735 129 End Sub
3736 130
3737 131 Private Sub Form_Current()
3738 132
3739 133 Me.cmbSSN.Enabled = True
3740 134 Me.cmbCourseNumber.Enabled = True
3741 135 Me.cmbClassNumber.Enabled = True
3742 136
3743 137
3744 138
3745 139 End Sub
3746 140
3747 141 Private Sub Form_Open(Cancel As Integer)
3748 142
3749 143
3750 144 DoCmd.GoToRecord acDataForm,
3751 frmSpecialAssignmentOfUnassignedMarine, acNewRec
3752 145
3753 146
3754 147 End Sub
3755 148
3756 149 Form: frmUnassignedMarines
3757 150 Code
3758 151 1 Attribute VB_Name = "Form_frmUnassignedMarines"
3759 152 2 Attribute VB_Creatable = True
3760 153 3 Attribute VB_PredeclaredId = True
3761 154 4 Attribute VB_Exposed = False
3762 155 5 Option Compare Database
3763 156 6 Option Explicit
3764 157
3765 158 8 Private Sub Close_Click()
3766 159 9 On Error GoTo Err_Close_Click
3767 160 10
3768 161 11 Dim sDocName As String
3769 162 12 Dim sLinkCriteria As String

```

```

3770 13
3771 14 ' Close the current form
3772 15 DoCmd.Close
3773 16
3774 17 ' Open specified form
3775 18 sDocName = "frmAnalyzeResult"
3776 19 DoCmd.OpenForm sDocName, , , sLinkCriteria
3777 20
3778 21 Exit_Close_Click
3779 22 Exit Sub
3780 23
3781 24 Err_Close_Click
3782 25 MsgBox Err.Description
3783 26 Resume Exit_Close_Click
3784 27
3785 28 End Sub
3786 29 Private Sub cmbSSNFind_AfterUpdate()
3787 30
3788 31 Dim R As Recordset
3789 32 Set R = Me.RecordsetClone
3790 33 R.FindFirst "[SSN_PK] = " & Chr(34) & Me[cmbSSNFind] & Chr(34)
3791 34 Me.Bookmark = R.Bookmark
3792 35 Me[cmbSSNFind] = Null
3793 36 Me.Close.SetFocus
3794 37
3795 38 End Sub
3796 39 Private Sub Form_Current()
3797 40
3798 41 Dim strSQL As String
3799 42 Dim rec As Recordset
3800 43 Dim strSQL As String
3801 44 Dim db1 As Database
3802 45
3803 46 Set db1 = CurrentDb()
3804 47
3805 48 ' Calculates the number of unassigned marines.
3806 49 strSQL = "SELECT Count(PEF) AS TotalUnassigned FROM
3807 50 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
3808 51 Me.tblTotalUnassigned = rec.TotalUnassigned
3809 52
3810 53 End Sub
3811 54 Private Sub btnAssign_Click()
3812 55 On Error GoTo Err_btnAssign_Click
3813 56
3814 57 Dim sDocName As String
3815 58
3816 59 sDocName = "frmSpecialAssignmentOfUnassignedMarine"
3817 60
3818 61 DoCmd.OpenForm sDocName
3819 62
3820 63 Exit_btnAssign_Click
3821 64 Exit Sub
3822 65
3823 66 Err_btnAssign_Click
3824 67 MsgBox Err.Description
3825 68 Resume Exit_btnAssign_Click
3826 69
3827 70 End Sub
3828 71
3829 72 Form: subfrmFundamentalProperty
3830 73 Code
3831 74 1 Attribute VB_Name = "Form_subfrmFundamentalProperty"
3832 75 2 Attribute VB_Creatable = True
3833 76 3 Attribute VB_PredeclaredId = True
3834 77 4 Attribute VB_Exposed = False
3835 78 5 Option Compare Database
3836 79 6 Option Explicit
3837 80 7 Private Sub cmbValue_FK_AfterUpdate()
3838 81
3839 82 Form: frmFundamentalProperty([FPropertyPK], SelfFocus
3840 83
3841 84 End Sub
3842 85
3843 86 Private Sub cmbValue_FK_BeforeUpdate(Cancel As Integer)
3844 87
3845 88 On Error GoTo Err cmbValue_FK_BeforeUpdate
3846 89
3847 90 Dim db As Database
3848 91 Dim rec As Recordset
3849 92 Dim strSQL As String
3850 93 Dim Convert As Variant
3851 94
3852 95 ' Places value in a form agreeable to Access (e.g. 'H' or '10')
3853 96 Convert = "" &
3854 97 Form: frmFundamentalProperty(subfrmFundamentalProperty.cmbValue_FK & ""
3855 98 24
3856 99 ' Opens the table PROPERTY_VALUE
3857 100 strSQL = "PROPERTY_VALUE"
3858 101 Set db = CurrentDb()
3859 102 Set rec = db.OpenRecordset(strSQL, dbOpenSnapshot)
3860 103
3861 104 ' Searches for the value entered by the user in the
3862 105 rec.FindFirst "Value_PK = " & Convert
3863 106
3864 107 ' If the value is not in the PROPERTY_VALUE table, it is added.
3865 108 If rec.NoMatch = True Then
3866 109
3867 110 ' Disables the action query confirmation message
3868 111 Application.SetOption "Confirm Action Queries", False
3869 112
3870 113 ' Enters the value into the PROPERTY_VALUE table
3871 114 DoCmd.OpenQuery "qryUpdateProperty_Value"
3872 115
3873 116 ' Updates the drop down list for the combobox
3874 117 strSQL = "SELECT DISTINCT " &
3875 118 Form: frmFundamentalProperty(MarineField) & " FROM MARINE;"
3876 119
3877 120 Form: frmFundamentalProperty(subfrmFundamentalProperty.cmbValue_FK.RowSource
3878 121 rse = strSQL
3879 122
3880 123 ' Enables the action query confirmation message
3881 124 Application.SetOption "Confirm Action Queries", True
3882 125 End If
3883 126
3884 127 rec.Close
3885 128
3886 129 Exit cmbValue_FK_BeforeUpdate:
3887 130 Exit Sub
3888 131
3889 132 Err cmbValue_FK_BeforeUpdate:
3890 133 MsgBox Err.Description
3891 134 Application.SetOption "Confirm Action Queries", True
3892 135 Resume Exit cmbValue_FK_BeforeUpdate
3893 136
3894 137 End Sub
3895 138
3896 139 Form: subfrmFundamentalPropertyList
3897 140 Code
3898 141 1 Attribute VB_Name = "Form_subfrmFundamentalPropertyList"

```

```

3900 2 Attribute VB_Creatable = True
3901 3 Attribute VB_PredeclaredId = True
3902 4 Attribute VB_Exposed = False
3903 5 Option Compare Database
3904 6 Option Explicit
3905 7
3906 8 Private Sub cmbValue_FK_BeforeUpdate(Cancel As Integer)
3907 9 On Error GoTo Err_cmbValue_FK_BeforeUpdate
3908 10
3909 11 Dim db As Database
3910 12 Dim rec As Recordset
3911 13 Dim strSQL As String
3912 14 Dim Convert As Variant
3913 15
3914 16 'Opens the subfrmFundamentalsProperty so the following code will
3915 17 DoCmd.OpenForm "subfrmFundamentalsProperty", , , , acHidden
3916 18
3917 19
3918 20 'Places value in a form agreeable to Access (e.g. 'HT' or '10')
3919 21 Convert = "" & Form!subfrmFundamentalsProperty!cmbValue_FK &
3920 22 'Opens the table PROPERTY_VALUE
3921 23 strSQL = "PROPERTY_VALUE"
3922 24
3923 25 Set db = CurrentDb()
3924 26 Set rec = db.OpenRecordset(strSQL, dbOpenSnapshot)
3925 27
3926 28 rec.FindFirst "Value_PK = " & Convert
3927 29 If rec.NoMatch = True Then
3928 30 ' Disables the action query confirmation message
3929 31 Application.SetOption "Confirm Action Queries", False
3930 32
3931 33 DoCmd.OpenQuery "qryUpdateProperty_Value"
3932 34 strSQL = "SELECT DISTINCT ' &
3933 35 Form!subfrmFundamentalsProperty!MarineField & ' FROM MARINE;"
3934 36 Me.cmbValue_FK.RowSource = strSQL
3935 37
3936 37 ' Enables the action query confirmation message
3937 38 Application.SetOption "Confirm Action Queries", True
3938 39 End If
3939 40
3940 41 rec.Close
3941 42
3942 43 'Closes the subfrmFundamentalsProperty without saving changes to
3943 44 DoCmd.Close acForm, "subfrmFundamentalsProperty"
3944 45
3945 46
3946 47 Exit_cmbValue_FK_BeforeUpdate:
3947 48 Exit Sub
3948 49
3949 50 Err_cmbValue_FK_BeforeUpdate:
3950 51 MsgBox Err.Description
3951 52 Application.SetOption "Confirm Action Queries", True
3952 53 Resume Exit_cmbValue_FK_BeforeUpdate
3953 54
3954 55 End Sub
3955 56
3956 57 Private Sub Value_FK_BeforeUpdate(Cancel As Integer)
3957 58 On Error GoTo Err_Value_FK_BeforeUpdate
3958 59
3959 60 Dim db As Database
3960 61 Dim rec As Recordset
3961 62 Dim strSQL As String
3962 63 Dim Convert As Variant
3963 64
3964 65 'Places value in a form agreeable to Access (e.g. 'HT' or '10')
3965 66 Convert = "" &
3966 67 Form!subfrmFundamentalsProperty!subfrmFundamentalsPropertyListValue_FK & ""
3967 68 'Opens the table PROPERTY_VALUE
3968 69 strSQL = "PROPERTY_VALUE"
3969 70
3970 71 Set db = CurrentDb()
3971 72 Set rec = db.OpenRecordset(strSQL, dbOpenSnapshot)
3972 73
3973 74 rec.FindFirst "Value_PK = " & Convert
3974 75 ' This If Then statement checks to see if the entered value is in
3975 76 ' PROPERTY_VALUE table. If not, it is added to the table.
3976 77 If rec.NoMatch = True Then
3977 78 ' Disables the action query confirmation message
3978 79 Application.SetOption "Confirm Action Queries", False
3979 80
3980 81 DoCmd.OpenQuery "qryUpdateProperty_ValueList"
3981 82
3982 82 ' Enables the action query confirmation message
3983 83 Application.SetOption "Confirm Action Queries", True
3984 84 End If
3985 85
3986 86 rec.Close
3987 87
3988 88 Exit_Value_FK_BeforeUpdate:
3989 89 Exit Sub
3990 90
3991 91 Err_Value_FK_BeforeUpdate:
3992 92 MsgBox Err.Description
3993 93 Application.SetOption "Confirm Action Queries", True
3994 94 Resume Exit_Value_FK_BeforeUpdate
3995 95
3996 96 End Sub
3997
3998
3999 Module: modFitnessDetermination
4000 Code
4001 1 Attribute VB_Name = "modFitnessDetermination"
4002 2 Option Compare Database
4003 3 Option Explicit
4004 4
4005 5 Dim lngStart, lngEnd As Long
4006 6
4007 7
4008 8 Public Function FundPropTest(FundPropName As String, SSN As String) As
4009 9
4010 10 Dim rec As Recordset, rec2 As Recordset, rec3 As Recordset
4011 11 Dim strMarField As String, strFundOperator As String, strSQL As
4012 12 Dim strMarValueSQL As String, strFundValueSQL As String, strInput As
4013 13 Dim verFundValue, verMarValue
4014 14 Dim db1 As Database
4015 15
4016 16 FundPropName = "" & FundPropName & "" 'This expression
4017 17 properly formats the argument FundPropName, by placing a ' in front of
4018 18 and behind the value found in FundPropName.
4019 19
4020 20 Set db1 = CurrentDb()
4021 21
4022 22 strSQL = "SELECT MarineField, Operator FROM FUNDAMENTAL_PROPERTY
4023 23 WHERE PropertyName_PK = " & FundPropName
4024 24 Set rec = db1.OpenRecordset(strSQL, dbOpenSnapshot)
4025 25 strMarField = rec!MarineField
4026 26
4027 27 strFundOperator = rec!Operator
4028 28
4029 29
4030 30 strInput = "" & SSN & ""
4031 31 strMarValueSQL = "SELECT " & strMarField & " AS [Value] FROM MARINE
4032 32 WHERE SSN_PK = " & strInput
4033 33 Set rec2 = db1.OpenRecordset(strMarValueSQL, dbOpenSnapshot)
4034 34 verMarValue = rec2!Value
4035 35
4036 36 strFundValueSQL = "SELECT Value_FK FROM FUND_PROP_VAL WHERE
4037 37 PropertyName_PK = " & FundPropName
4038 38 Set rec3 = db1.OpenRecordset(strFundValueSQL, dbOpenSnapshot)
4039 39 verFundValue = rec3!Value_FK
4040 40
4041 41 ' Convert verFundValue and verMarValue to a numeric, if possible
4042 42 If IsNumeric(verFundValue) And IsNumeric(verMarValue) Then
4043 43 verFundValue = CInt(verFundValue)
4044 44 verMarValue = CInt(verMarValue)
4045 45 Else
4046 46 verFundValue = CStr(verFundValue)
4047 47 verMarValue = CStr(verMarValue)
4048 48 End If
4049 49
4050 50
4051 51 Select Case strFundOperator
4052 52 Case "="
4053 53 If verMarValue = verFundValue Then
4054 54 FundPropTest = True
4055 55 End If
4056 56 Case "<"
4057 57 If verMarValue < verFundValue Then
4058 58 FundPropTest = True
4059 59 End If
4060 60 Case "<="
4061 61 If verMarValue <= verFundValue Then
4062 62 FundPropTest = True
4063 63 End If
4064 64 Case ">"
4065 65 If verMarValue > verFundValue Then
4066 66 FundPropTest = True
4067 67 End If
4068 68 Case ">="
4069 69 If verMarValue >= verFundValue Then
4070 70 FundPropTest = True
4071 71 End If
4072 72 Case "<>"
4073 73 If verMarValue <> verFundValue Then
4074 74 FundPropTest = True
4075 75 End If
4076 76 Case ">="
4077 77 If verMarValue >= verFundValue Then
4078 78 FundPropTest = True
4079 79 End If
4080 80 Case "<="
4081 81 If verMarValue <= verFundValue Then
4082 82 FundPropTest = True
4083 83 End If
4084 84 Case "<"
4085 85 If verMarValue < verFundValue Then
4086 86 FundPropTest = True
4087 87 End If
4088 88 Case ">"
4089 89 If verMarValue > verFundValue Then
4090 90 FundPropTest = True
4091 91 End If
4092 92 Case "="
4093 93 If verMarValue = verFundValue Then
4094 94 FundPropTest = True
4095 95 End If
4096 96 Case "<>"
4097 97 If verMarValue <> verFundValue Then
4098 98 FundPropTest = True
4099 99 End If
4100 100 Case "<"
4101 101 If verMarValue < verFundValue Then
4102 102 FundPropTest = True
4103 103 End If
4104 104 Case ">"
4105 105 If verMarValue > verFundValue Then
4106 106 FundPropTest = True
4107 107 End If
4108 108 Case "="
4109 109 If verMarValue = verFundValue Then
4110 110 FundPropTest = True
4111 111 End If
4112 112 Case "<>"
4113 113 If verMarValue <> verFundValue Then
4114 114 FundPropTest = True
4115 115 End If
4116 116 Case "<"
4117 117 If verMarValue < verFundValue Then
4118 118 FundPropTest = True
4119 119 End If
4120 120 Case ">"
4121 121 If verMarValue > verFundValue Then
4122 122 FundPropTest = True
4123 123 End If
4124 124 Case "="
4125 125 If verMarValue = verFundValue Then
4126 126 FundPropTest = True
4127 127 End If
4128 128 Case "<>"
4129 129 If verMarValue <> verFundValue Then
4130 130 FundPropTest = True
4131 131 End If
4132 132 Case "<"
4133 133 If verMarValue < verFundValue Then
4134 134 FundPropTest = True
4135 135 End If
4136 136 Case ">"
4137 137 If verMarValue > verFundValue Then
4138 138 FundPropTest = True
4139 139 End If
4140 140 Case "="
4141 141 If verMarValue = verFundValue Then
4142 142 FundPropTest = True
4143 143 End If
4144 144 Case "<>"
4145 145 If verMarValue <> verFundValue Then
4146 146 FundPropTest = True
4147 147 End If
4148 148 Case "<"
4149 149 If verMarValue < verFundValue Then
4150 150 FundPropTest = True
4151 151 End If
4152 152 Case ">"
4153 153 If verMarValue > verFundValue Then
4154 154 FundPropTest = True
4155 155 End If
4156 156 Case "="
4157 157 If verMarValue = verFundValue Then
4158 158 FundPropTest = True
4159 159 End If
4160 160 Case "<>"
4161 161 If verMarValue <> verFundValue Then
4162 162 FundPropTest = True
4163 163 End If
4164 164 Case "<"
4165 165 If verMarValue < verFundValue Then
4166 166 FundPropTest = True
4167 167 End If
4168 168 Case ">"
4169 169 If verMarValue > verFundValue Then
4170 170 FundPropTest = True
4171 171 End If
4172 172 Case "="
4173 173 If verMarValue = verFundValue Then
4174 174 FundPropTest = True
4175 175 End If
4176 176 Case "<>"
4177 177 If verMarValue <> verFundValue Then
4178 178 FundPropTest = True
4179 179 End If
4180 180 Case "<"
4181 181 If verMarValue < verFundValue Then
4182 182 FundPropTest = True
4183 183 End If
4184 184 Case ">"
4185 185 If verMarValue > verFundValue Then
4186 186 FundPropTest = True
4187 187 End If
4188 188 Case "="
4189 189 If verMarValue = verFundValue Then
4190 190 FundPropTest = True
4191 191 End If
4192 192 Case "<>"
4193 193 If verMarValue <> verFundValue Then
4194 194 FundPropTest = True
4195 195 End If
4196 196 Case "<"
4197 197 If verMarValue < verFundValue Then
4198 198 FundPropTest = True
4199 199 End If
4200 200 Case ">"
4201 201 If verMarValue > verFundValue Then
4202 202 FundPropTest = True
4203 203 End If
4204 204 Case "="
4205 205 If verMarValue = verFundValue Then
4206 206 FundPropTest = True
4207 207 End If
4208 208 Case "<>"
4209 209 If verMarValue <> verFundValue Then
4210 210 FundPropTest = True
4211 211 End If
4212 212 Case "<"
4213 213 If verMarValue < verFundValue Then
4214 214 FundPropTest = True
4215 215 End If
4216 216 Case ">"
4217 217 If verMarValue > verFundValue Then
4218 218 FundPropTest = True
4219 219 End If
4220 220 Case "="
4221 221 If verMarValue = verFundValue Then
4222 222 FundPropTest = True
4223 223 End If
4224 224 Case "<>"
4225 225 If verMarValue <> verFundValue Then
4226 226 FundPropTest = True
4227 227 End If
4228 228 Case "<"
4229 229 If verMarValue < verFundValue Then
4230 230 FundPropTest = True
4231 231 End If
4232 232 Case ">"
4233 233 If verMarValue > verFundValue Then
4234 234 FundPropTest = True
4235 235 End If
4236 236 Case "="
4237 237 If verMarValue = verFundValue Then
4238 238 FundPropTest = True
4239 239 End If
4240 240 Case "<>"
4241 241 If verMarValue <> verFundValue Then
4242 242 FundPropTest = True
4243 243 End If
4244 244 Case "<"
4245 245 If verMarValue < verFundValue Then
4246 246 FundPropTest = True
4247 247 End If
4248 248 Case ">"
4249 249 If verMarValue > verFundValue Then
4250 250 FundPropTest = True
4251 251 End If
4252 252 Case "="
4253 253 If verMarValue = verFundValue Then
4254 254 FundPropTest = True
4255 255 End If
4256 256 Case "<>"
4257 257 If verMarValue <> verFundValue Then
4258 258 FundPropTest = True
4259 259 End If
4260 260 Case "<"
4261 261 If verMarValue < verFundValue Then
4262 262 FundPropTest = True
4263 263 End If
4264 264 Case ">"
4265 265 If verMarValue > verFundValue Then
4266 266 FundPropTest = True
4267 267 End If
4268 268 Case "="
4269 269 If verMarValue = verFundValue Then
4270 270 FundPropTest = True
4271 271 End If
4272 272 Case "<>"
4273 273 If verMarValue <> verFundValue Then
4274 274 FundPropTest = True
4275 275 End If
4276 276 Case "<"
4277 277 If verMarValue < verFundValue Then
4278 278 FundPropTest = True
4279 279 End If
4280 280 Case ">"
4281 281 If verMarValue > verFundValue Then
4282 282 FundPropTest = True
4283 283 End If
4284 284 Case "="
4285 285 If verMarValue = verFundValue Then
4286 286 FundPropTest = True
4287 287 End If
4288 288 Case "<>"
4289 289 If verMarValue <> verFundValue Then
4290 290 FundPropTest = True
4291 291 End If
4292 292 Case "<"
4293 293 If verMarValue < verFundValue Then
4294 294 FundPropTest = True
4295 295 End If
4296 296 Case ">"
4297 297 If verMarValue > verFundValue Then
4298 298 FundPropTest = True
4299 299 End If
4300 300 Case "="
4301 301 If verMarValue = verFundValue Then
4302 302 FundPropTest = True
4303 303 End If
4304 304 Case "<>"
4305 305 If verMarValue <> verFundValue Then
4306 306 FundPropTest = True
4307 307 End If
4308 308 Case "<"
4309 309 If verMarValue < verFundValue Then
4310 310 FundPropTest = True
4311 311 End If
4312 312 Case ">"
4313 313 If verMarValue > verFundValue Then
4314 314 FundPropTest = True
4315 315 End If
4316 316 Case "="
4317 317 If verMarValue = verFundValue Then
4318 318 FundPropTest = True
4319 319 End If
4320 320 Case "<>"
4321 321 If verMarValue <> verFundValue Then
4322 322 FundPropTest = True
4323 323 End If
4324 324 Case "<"
4325 325 If verMarValue < verFundValue Then
4326 326 FundPropTest = True
4327 327 End If
4328 328 Case ">"
4329 329 If verMarValue > verFundValue Then
4330 330 FundPropTest = True
4331 331 End If
4332 332 Case "="
4333 333 If verMarValue = verFundValue Then
4334 334 FundPropTest = True
4335 335 End If
4336 336 Case "<>"
4337 337 If verMarValue <> verFundValue Then
4338 338 FundPropTest = True
4339 339 End If
4340 340 Case "<"
4341 341 If verMarValue < verFundValue Then
4342 342 FundPropTest = True
4343 343 End If
4344 344 Case ">"
4345 345 If verMarValue > verFundValue Then
4346 346 FundPropTest = True
4347 347 End If
4348 348 Case "="
4349 349 If verMarValue = verFundValue Then
4350 350 FundPropTest = True
4351 351 End If
4352 352 Case "<>"
4353 353 If verMarValue <> verFundValue Then
4354 354 FundPropTest = True
4355 355 End If
4356 356 Case "<"
4357 357 If verMarValue < verFundValue Then
4358 358 FundPropTest = True
4359 359 End If
4360 360 Case ">"
4361 361 If verMarValue > verFundValue Then
4362 362 FundPropTest = True
4363 363 End If
4364 364 Case "="
4365 365 If verMarValue = verFundValue Then
4366 366 FundPropTest = True
4367 367 End If
4368 368 Case "<>"
4369 369 If verMarValue <> verFundValue Then
4370 370 FundPropTest = True
4371 371 End If
4372 372 Case "<"
4373 373 If verMarValue < verFundValue Then
4374 374 FundPropTest = True
4375 375 End If
4376 376 Case ">"
4377 377 If verMarValue > verFundValue Then
4378 378 FundPropTest = True
4379 379 End If
4380 380 Case "="
4381 381 If verMarValue = verFundValue Then
4382 382 FundPropTest = True
4383 383 End If
4384 384 Case "<>"
4385 385 If verMarValue <> verFundValue Then
4386 386 FundPropTest = True
4387 387 End If
4388 388 Case "<"
4389 389 If verMarValue < verFundValue Then
4390 390 FundPropTest = True
4391 391 End If
4392 392 Case ">"
4393 393 If verMarValue > verFundValue Then
4394 394 FundPropTest = True
4395 395 End If
4396 396 Case "="
4397 397 If verMarValue = verFundValue Then
4398 398 FundPropTest = True
4399 399 End If
4400 400 Case "<>"
4401 401 If verMarValue <> verFundValue Then
4402 402 FundPropTest = True
4403 403 End If
4404 404 Case "<"
4405 405 If verMarValue < verFundValue Then
4406 406 FundPropTest = True
4407 407 End If
4408 408 Case ">"
4409 409 If verMarValue > verFundValue Then
4410 410 FundPropTest = True
4411 411 End If
4412 412 Case "="
4413 413 If verMarValue = verFundValue Then
4414 414 FundPropTest = True
4415 415 End If
4416 416 Case "<>"
4417 417 If verMarValue <> verFundValue Then
4418 418 FundPropTest = True
4419 419 End If
4420 420 Case "<"
4421 421 If verMarValue < verFundValue Then
4422 422 FundPropTest = True
4423 423 End If
4424 424 Case ">"
4425 425 If verMarValue > verFundValue Then
4426 426 FundPropTest = True
4427 427 End If
4428 428 Case "="
4429 429 If verMarValue = verFundValue Then
4430 430 FundPropTest = True
4431 431 End If
4432 432 Case "<>"
4433 433 If verMarValue <> verFundValue Then
4434 434 FundPropTest = True
4435 435 End If
4436 436 Case "<"
4437 437 If verMarValue < verFundValue Then
4438 438 FundPropTest = True
4439 439 End If
4440 440 Case ">"
4441 441 If verMarValue > verFundValue Then
4442 442 FundPropTest = True
4443 443 End If
4444 444 Case "="
4445 445 If verMarValue = verFundValue Then
4446 446 FundPropTest = True
4447 447 End If
4448 448 Case "<>"
4449 449 If verMarValue <> verFundValue Then
4450 450 FundPropTest = True
4451 451 End If
4452 452 Case "<"
4453 453 If verMarValue < verFundValue Then
4454 454 FundPropTest = True
4455 455 End If
4456 456 Case ">"
4457 457 If verMarValue > verFundValue Then
4458 458 FundPropTest = True
4459 459 End If
4460 460 Case "="
4461 461 If verMarValue = verFundValue Then
4462 462 FundPropTest = True
4463 463 End If
4464 464 Case "<>"
4465 465 If verMarValue <> verFundValue Then
4466 466 FundPropTest = True
4467 467 End If
4468 468 Case "<"
4469 469 If verMarValue < verFundValue Then
4470 470 FundPropTest = True
4471 471 End If
4472 472 Case ">"
4473 473 If verMarValue > verFundValue Then
4474 474 FundPropTest = True
4475 475 End If
4476 476 Case "="
4477 477 If verMarValue = verFundValue Then
4478 478 FundPropTest = True
4479 479 End If
4480 480 Case "<>"
4481 481 If verMarValue <> verFundValue Then
4482 482 FundPropTest = True
4483 483 End If
4484 484 Case "<"
4485 485 If verMarValue < verFundValue Then
4486 486 FundPropTest = True
4487 487 End If
4488 488 Case ">"
4489 489 If verMarValue > verFundValue Then
4490 490 FundPropTest = True
4491 491 End If
4492 492 Case "="
4493 493 If verMarValue = verFundValue Then
4494 494 FundPropTest = True
4495 495 End If
4496 496 Case "<>"
4497 497 If verMarValue <> verFundValue Then
4498 498 FundPropTest = True
4499 499 End If
4500 500 Case "<"
4501 501 If verMarValue < verFundValue Then
4502 502 FundPropTest = True
4503 503 End If
4504 504 Case ">"
4505 505 If verMarValue > verFundValue Then
4506 506 FundPropTest = True
4507 507 End If
4508 508 Case "="
4509 509 If verMarValue = verFundValue Then
4510 510 FundPropTest = True
4511 511 End If
4512 512 Case "<>"
4513 513 If verMarValue <> verFundValue Then
4514 514 FundPropTest = True
4515 515 End If
4516 516 Case "<"
4517 517 If verMarValue < verFundValue Then
4518 518 FundPropTest = True
4519 519 End If
4520 520 Case ">"
4521 521 If verMarValue > verFundValue Then
4522 522 FundPropTest = True
4523 523 End If
4524 524 Case "="
4525 525 If verMarValue = verFundValue Then
4526 526 FundPropTest = True
4527 527 End If
4528 528 Case "<>"
4529 529 If verMarValue <> verFundValue Then
4530 530 FundPropTest = True
4531 531 End If
4532 532 Case "<"
4533 533 If verMarValue < verFundValue Then
4534 534 FundPropTest = True
4535 535 End If
4536 536 Case ">"
4537 537 If verMarValue > verFundValue Then
4538 538 FundPropTest = True
4539 539 End If
4540 540 Case "="
4541 541 If verMarValue = verFundValue Then
4542 542 FundPropTest = True
4543 543 End If
4544 544 Case "<>"
4545 545 If verMarValue <> verFundValue Then
4546 546 FundPropTest = True
4547 547 End If
4548 548 Case "<"
4549 549 If verMarValue < verFundValue Then
4550 550 FundPropTest = True
4551 551 End If
4552 552 Case ">"
4553 553 If verMarValue > verFundValue Then
4554 554 FundPropTest = True
4555 555 End If
4556 556 Case "="
4557 557 If verMarValue = verFundValue Then
4558 558 FundPropTest = True
4559 559 End If
4560 560 Case "<>"
4561 561 If verMarValue <> verFundValue Then
4562 562 FundPropTest = True
4563 563 End If
4564 564 Case "<"
4565 565 If verMarValue < verFundValue Then
4566 566 FundPropTest = True
4567 567 End If
4568 568 Case ">"
4569 569 If verMarValue > verFundValue Then
4570 570 FundPropTest = True
4571 571 End If
4572 572 Case "="
4573 573 If verMarValue = verFundValue Then
4574 574 FundPropTest = True
4575 575 End If
4576 576 Case "<>"
4577 577 If verMarValue <> verFundValue Then
4578 578 FundPropTest = True
4579 579 End If
4580 580 Case "<"
4581 581 If verMarValue < verFundValue Then
4582 582 FundPropTest = True
4583 583 End If
4584 584 Case ">"
4585 585 If verMarValue > verFundValue Then
4586 586 FundPropTest = True
4587 587 End If
4588 588 Case "="
4589 589 If verMarValue = verFundValue Then
4590 590 FundPropTest = True
4591 591 End If
4592 592 Case "<>"
4593 593 If verMarValue <> verFundValue Then
4594 594 FundPropTest = True
4595 595 End If
4596 596 Case "<"
4597 597 If verMarValue < verFundValue Then
4598 598 FundPropTest = True
4599 599 End If
4600 600 Case ">"
4601 601 If verMarValue > verFundValue Then
4602 602 FundPropTest = True
4603 603 End If
4604 604 Case "="
4605 605 If verMarValue = verFundValue Then
4606 606 FundPropTest = True
4607 607 End If
4608 608 Case "<>"
4609 609 If verMarValue <> verFundValue Then
4610 610 FundPropTest = True
4611 611 End If
4612 612 Case "<"
4613 613 If verMarValue < verFundValue Then
4614 614 FundPropTest = True
4615 615 End If
4616 616 Case ">"
4617 617 If verMarValue > verFundValue Then
4618 618 FundPropTest = True
4619 619 End If
4620 620 Case "="
4621 621 If verMarValue = verFundValue Then
4622 622 FundPropTest = True
4623 623 End If
4624 624 Case "<>"
4625 625 If verMarValue <> verFundValue Then
4626 626 FundPropTest = True
4627 627 End If
4628 628 Case "<"
4629 629 If verMarValue < verFundValue Then
4630 630 FundPropTest = True
4631 631 End If
4632 632 Case ">"
4633 633 If verMarValue > verFundValue Then
4634 634 FundPropTest = True
4635 635 End If
4636 636 Case "="
4637 637 If verMarValue = verFundValue Then
4638 638 FundPropTest = True
4639 639 End If
4640 640 Case "<>"
4641 641 If verMarValue <> verFundValue Then
4642 642 FundPropTest = True
4643 643 End If
4644 644 Case "<"
4645 645 If verMarValue < verFundValue Then
4646 646 FundPropTest = True
4647 647 End If
4648 648 Case ">"
4649 649 If verMarValue > verFundValue Then
4650 650 FundPropTest = True
4651 651 End If
4652 652 Case "="
4653 653 If verMarValue = verFundValue Then
4654 654 FundPropTest = True
4655 655 End If
4656 656 Case "<>"
4657 657 If verMarValue <> verFundValue Then
4658 658 FundPropTest = True
4659 659 End If
4660 660 Case "<"
4661 661 If verMarValue < verFundValue Then
4662 662 FundPropTest = True
4663 663 End If
4664 664 Case ">"
4665 665 If verMarValue > verFundValue Then
4666 666 FundPropTest = True
4667 667 End If
4668 668 Case "="
4669 669 If verMarValue = verFundValue Then
4670 670 FundPropTest = True
4671 671 End If
4672 672 Case "<>"
4673 673 If verMarValue <> verFundValue Then
4674 674 FundPropTest = True
4675 675 End If
4676 676 Case "<"
4677 677 If verMarValue < verFundValue Then
4678 678 FundPropTest = True
4679 679 End If
4680 680 Case ">"
4681 681 If verMarValue > verFundValue Then
4682 682 FundPropTest = True
4683 683 End If
4684 684 Case "="
4685 685 If verMarValue = verFundValue Then
4686 686 FundPropTest = True
4687 687 End If
4688 688 Case "<>"
4689 689 If verMarValue <> verFundValue Then
4690 690 FundPropTest = True
4691 691 End If
4692 692 Case "<"
4693 693 If verMarValue < verFundValue Then
4694 694 FundPropTest = True
4695 695 End If
4696 696 Case ">"
4697 697 If verMarValue > verFundValue Then
4698 698 FundPropTest = True
4699 699 End If
4700 700 Case "="
4701 701 If verMarValue = verFundValue Then
4702 702 FundPropTest = True
4703 703 End If
4704 704 Case "<>"
4705 705 If verMarValue <> verFundValue Then
4706 706 FundPropTest = True
4707 707 End If
4708 708 Case "<"
4709 709 If verMarValue < verFundValue Then
4710 710 FundPropTest = True
4711 711 End If
4712 712 Case ">"
4713 713 If verMarValue > verFundValue Then
4714 714 FundPropTest = True
4715 715 End If
4716 716 Case "="
4717 717 If verMarValue = verFundValue Then
4718 718 FundPropTest = True
4719 719 End If
4720 720 Case "<>"
4721 721 If verMarValue <> verFundValue Then
4722 722 FundPropTest = True
4723 723 End If
4724 724 Case "<"
4725 725 If verMarValue < verFundValue Then
4726 726 FundPropTest = True
4727 727 End If
4728 728 Case ">"
4729 729 If verMarValue > verFundValue Then
4730 730 FundPropTest = True
4731 731 End If
4732 732 Case "="
4733 733 If verMarValue = verFundValue Then
4734 734 FundPropTest = True
4735 735 End If
4736 736 Case "<>"
4737 737 If verMarValue <> verFundValue Then
4738 738 FundPropTest = True
4739 739 End If
4740 740 Case "<"
4741 741 If verMarValue < verFundValue Then
4742 742 FundPropTest = True
4743 743 End If
4744 744 Case ">"
4745 745 If verMarValue > verFundValue Then
4746 746 FundPropTest = True
4747 747 End If
4748 748 Case "="
4749 749 If verMarValue = verFundValue Then
4750 750 FundPropTest = True
4751 751 End If
4752 752 Case "<>"
4753 7
```

```

4160 145 characters, therefore we subtract the position of the space
4161 146 the length of the string. This gives us the number of
4162 characters remaining.
4163 147 strLogicalEquation = LTrim(Right$(strLogicalEquation,
4164 Len(strLogicalEquation) - intSpacePos))
4165 148
4166 149 'find the next space
4167 150 intSpacePos = InStr(strLogicalEquation, " ")
4168 151 Loop
4169 152
4170 153 For j = 1 To (UBound(strWorkingEqn))
4171 154 If strWorkingEqn(j) = "(" Then
4172 155
4173 156 Line1:
4174 157 For k = (j + 1) To (UBound(strWorkingEqn))
4175 158 Select Case strWorkingEqn(k)
4176 159 Case ")"
4177 160 For i = (j + 1) To (k - 1)
4178 161 Select Case strWorkingEqn(i)
4179 162 Case "("
4180 163 boolOperator = True
4181 164 Case "And"
4182 165 boolAndOperator = True
4183 166 Case "Not"
4184 167 boolNotOperator = True
4185 168 Case "True"
4186 169 If strWorkingEqn(i) = "(" Then
4187 170 If boolOperator Then
4188 171 strWorkingEqn(i) = "False"
4189 172 boolNotOperator = False
4190 173 Else
4191 174 strWorkingEqn(i) = "True"
4192 175 End If
4193 176 Else 'Executed after the first
4194 177 bracketed property is tested and stored in the location of "("
4195 178 If boolOperator Then
4196 179 strWorkingEqn(i) = "False"
4197 180 boolNotOperator = False
4198 181 End If
4199 182 If boolOperator Then
4200 183 strWorkingEqn(i) =
4201 LogOrTest(CBool(strWorkingEqn(i)), CBool(strWorkingEqn(i)))
4202 184 ElseIf boolAndOperator Then
4203 185 strWorkingEqn(i) =
4204 LogAndTest(CBool(strWorkingEqn(i)), CBool(strWorkingEqn(i)))
4205 186 ElseIf boolNotOperator = False
4206 187 strWorkingEqn(i) = "False"
4207 188 End If
4208 189 Case "False"
4209 190 If strWorkingEqn(i) = "(" Then
4210 191 If boolOperator Then
4211 192 strWorkingEqn(i) = "True"
4212 193 boolNotOperator = True
4213 194 Else
4214 195 strWorkingEqn(i) = "False"
4215 196 End If
4216 197 Else 'Executed after the first
4217 198 bracketed property is tested and stored in the location of "("
4219 199 If boolOperator Then
4220 200 strWorkingEqn(i) = "True"
4221 201 boolNotOperator = False
4222 202 End If
4223 203 If boolOperator Then
4224 204 strWorkingEqn(i) =
4225 LogOrTest(CBool(strWorkingEqn(i)), CBool(strWorkingEqn(i)))
4226 205 ElseIf boolAndOperator = False
4227 206 strWorkingEqn(i) =
4228 LogAndTest(CBool(strWorkingEqn(i)), CBool(strWorkingEqn(i)))
4229 207 ElseIf boolNotOperator = False
4230 208 strWorkingEqn(i) = "False"
4231 209 End If
4232 210 Case "}" 'Do nothing
4233 211 Case Else
4234 212 If strWorkingEqn(i) = "(" Then
4235 213 If boolOperator Then
4236 214 strWorkingEqn(i) = Not
4237 (FundPropTest(strWorkingEqn(i), SSN))
4238 215 ElseIf boolNotOperator = False
4239 216 strWorkingEqn(i) =
4240 FundPropTest(strWorkingEqn(i), SSN)
4241 217 Else 'Executed after the first
4242 218 bracketed property is tested and stored in the location of "("
4243 219 If boolOperator Then
4244 220 strWorkingEqn(i) = Not
4245 221 (FundPropTest(strWorkingEqn(i), SSN))
4246 222 ElseIf boolNotOperator = False
4247 223 strWorkingEqn(i) =
4248 FundPropTest(strWorkingEqn(i), SSN)
4249 224 Else
4250 225 strWorkingEqn(i) =
4251 FundPropTest(strWorkingEqn(i), SSN)
4252 226 End If
4253 227 If boolOperator Then
4254 228 strWorkingEqn(i) =
4255 LogOrTest(CBool(strWorkingEqn(i)), CBool(strWorkingEqn(i)))
4256 229 ElseIf boolAndOperator = False
4257 230 strWorkingEqn(i) =
4258 LogAndTest(CBool(strWorkingEqn(i)), CBool(strWorkingEqn(i)))
4259 231 ElseIf boolNotOperator = False
4260 232 strWorkingEqn(i) = "False"
4261 233 End If
4262 234 End Select
4263 235 Next i
4264 236 Next k
4265 237 Next j
4266 238 'Must transform the working equation to account
4267 239 for the bracketed expression evaluation.
4268 240
4269 241 intNewUBound = UBound(strWorkingEqn) - (k - j)
4270 242 If intNewUBound <= 0 Then
4271 243 ReDim strTransformedEqn(1 To intNewUBound)
4272 244 For m = 1 To j
4273 245 strTransformedEqn(m) = strWorkingEqn(m)
4274 246 Next m
4275 247 For m = (k + 1) To UBound(strWorkingEqn)
4276 248 strTransformedEqn(m - k + j) =
4277 strWorkingEqn(m)
4278 249 Next m
4279 250 ReDim strWorkingEqn(1 To intNewUBound)
4280 251 For m = 1 To intNewUBound
4281 252 strWorkingEqn(m) = strTransformedEqn(m)
4282 253 Next m
4283 254 j = 1 'Set back to a value of 1.
4284 255 GoTo Line1 'Jump back to For j = 1 to
4285 (UBound(strWorkingEqn)) loop
4286 256 Else
4287 257 LogPropTest = strWorkingEqn(j)
4288 258 Erase strWorkingEqn
4289 259 Erase strTransformedEqn

```

```

4290 258 Exit Function
4291 259 End If
4292 260 Case "C"
4293 261 i = k
4294 262 GoTo Line1 'Jumps back to For j = 1 to
4295 (UBound(strWorkingEqn)) loop
4296 263 Exit For 'Exits the For k = (j + 1) to
4297 (UBound(strWorkingEqn)) loop
4298 264 End Select
4299 265 Next k
4300 266 End If
4301 267 Next j
4302 268
4303 269 End Function
4304 270
4305 271 Public Sub Fitness()
4306 272 Dim strSQL As String, strClassNumber As String, strAMCS As String
4307 273 Dim strInput1 As String, strInput2 As String
4308 274 Dim i As Integer, j As Integer, k As Integer, l As Integer
4309 275 Dim intNormalFactor As Integer, intFit As Integer, intInput As
4310 276 Dim intLevel1 As Integer, intLevel2 As Integer, intLevel3 As Integer
4311 277 Dim intLevel4 As Integer, intLevel5 As Integer, intLevel6 As Integer
4312 278 Dim angDesiredWeight As Single, angEpsilon As Single, angSumOfFit As
4313 279 Dim angCurrent As Single, angAvgFit As Single, angFitness As Single
4314 280 Dim angMandFitValue As Single
4315 281 Dim lngGradDate As Long
4316 282 Dim db1 As Database
4317 283 Dim recClass As Recordset, recMarine As Recordset, recMarChFit As
4318 284 Dim recPEF As Recordset, recDesired As Recordset, recLDesired As
4319 285 Dim recMand As Recordset, recL Mand As Recordset
4320 286 Dim boolMandProp As Boolean, boolPEF As Boolean, boolMandatory As
4321 287 Dim varRetVal As Variant
4322 288 Dim frmCurrent As Form, frmClassQuotaPenaltyAndFit
4323 289
4324 290 Set frmCurrent = Form!frmClassQuotaPenaltyAndFit
4325 291 Set db1 = CurrentDb()
4326 292
4327 293 'Create TEMP_FIT table
4328 294 Application.SetOption "Confirm Action Queries", False
4329 295
4330 296 lngGradDate = frmCurrent.MCTGradDate
4331 297
4332 298 'Create 2 TEMP_FIT tables from MAR_CLS_FIT table
4333 299 db1.Execute "INSERT INTO TEMP_FIT ( SSN,FK ) SELECT MARINE.SSN,FK FROM
4334 300 MAR_CLS_FIT WHERE GradDate = " & lngGradDate & "
4335 301 db1.Execute "INSERT INTO TEMP_FIT1 ( SSN,FK ) SELECT MARINE.SSN,FK
4336 302 FROM MAR_CLS_FIT WHERE GradDate = " & lngGradDate & "
4337 303
4338 303 'Insert into the TEMP_FIT tables the applicable SSNs based on MCT
4339 304 graduation date
4340 305 db1.Execute "INSERT INTO TEMP_FIT1 ( SSN,FK ) SELECT MARINE.SSN,FK FROM
4341 306 MAR_CLS_FIT WHERE GradDate = " & lngGradDate & "
4342 307 db1.Execute "INSERT INTO TEMP_FIT1 ( SSN,FK ) SELECT MARINE.SSN,FK
4343 308 FROM MAR_CLS_FIT WHERE GradDate = " & lngGradDate & "
4344 309 Application.SetOption "Confirm Action Queries", True
4345 310
4346 310 'Create a dynaset of the temp_fit table
4347 311 strSQL = "SELECT * FROM TEMP_FIT"
4348 312 Set recClass = db1.OpenRecordset(strSQL, dbOpenDynaset)
4349 313 If recClass.EOF = False Then
4350 314 recClass.MoveLast
4351 315 recClass.MoveFirst
4352 316
4353 316 'Create the MarineIndex for the Temp_fit table
4354 317 If recClass.RecordCount > 0 Then
4355 318 recMarChFit.MoveFirst
4356 319 For i = 1 To recMarChFit.RecordCount
4357 320 recMarChFit.Edit
4358 321 recMarChFit.MarineIndex = i
4359 322 recMarChFit.Update
4360 323 recMarChFit.MoveNext
4361 324 Next i
4362 325 End If
4363 326
4364 326 'Create a dynaset of the Class table
4365 327 strSQL = "SELECT * FROM MARINE_FK_AMCS_FK FROM CLASS"
4366 328 Set recClass = db1.OpenRecordset(strSQL, dbOpenDynaset)
4367 329 If recClass.EOF = False Then
4368 330 recClass.MoveLast
4369 331 recClass.MoveFirst
4370 332 recClass.MoveFirst
4371 333 End If
4372 333 'Create a dynaset of the Marine table based on TEMP_FIT1 table
4373 334 strSQL = "SELECT * FROM MARINE_FK_TEMP_FIT1 WHERE
4374 335 ((MARINE.SSN,FK)=(TEMP_FIT1(SSN,FK)))"
4375 336 Set recMarine = db1.OpenRecordset(strSQL, dbOpenDynaset)
4376 337
4377 337 If recMarine.EOF = False Then
4378 338 recMarine.MoveLast
4379 339 recMarine.MoveFirst
4380 340 End If
4381 341
4382 342 'Initialize variables
4383 343 boolMandProp = True
4384 344 intFit = 0
4385 345 boolPEF = False
4386 346 boolMandatory = False
4387 347 intNormalFactor = 100
4388 348 intLevel1 = 32
4389 349 intLevel2 = 16
4390 350 intLevel3 = 8
4391 351 intLevel4 = 4
4392 352 intLevel5 = 2
4393 353 intLevel6 = 1
4394 354 angDesiredWeight =
4395 355 angMandFitValue = (1 - angDesiredWeight) * intNormalFactor
4396 356 varRetVal = SysCmd(acSysCmdInMeter, "Status...")
4397 357 angEpsilon = 0.000001
4398 358
4399 359 'Verify recClass has values
4400 360 If recClass.RecordCount > 0 Then
4401 361 strClassNumber = "X"
4402 362 strAMCS = "X"
4403 363
4404 364 'Cycle through all the classes
4405 365 For i = 1 To recClass.RecordCount
4406 366 'Yield control temporarily to Windows
4407 367 DoEvents
4408 368 'Update status bar
4409 369 varRetVal = SysCmd(acSysCmdUpdateMeter, i)
4410 370
4411 371 'Update the classindex_FK for the MarChFit record
4412 372 If recMarChFit.RecordCount > 0 Then
4413 373 recMarChFit.MoveFirst
4414 374 For j = 1 To recMarChFit.RecordCount
4415 375 recMarChFit.Edit
4416 376 recMarChFit.Classindex_FK = i
4417 377 recMarChFit.Update
4418 378 recMarChFit.MoveNext
4419 379 Next j

```

```

4420 380      recMarClsFit.MoveFirst
4421 381      End If
4422 382      * Check to see if next record is the same school as the last
4423 383      * If it is, append it to the MAR_CLS_FIT table
4424 384      If (recClassSCourseNumber_FK = strClassNumber) And
4425 385      Application.SeOption "Confirm Action Queries", False
4426 386      db1.Execute "INSERT INTO MAR_CLS_FIT (SSN_FK,
4427 387      MarineIndex, ClassIndex_FK, Fitness) SELECT TEMP_FIT.SSN_FK,
4428 388      TEMP_FIT.MarineIndex, TEMP_FIT.ClassIndex_FK, TEMP_FIT.Fitness FROM
4429 389      TEMP_FIT WHERE TEMP_FIT.Fitness > 0;"
4430 390      Application.SeOption "Confirm Action Queries", True
4431 391      recClass.MoveNext
4432 392      Else
4433 393      angSumOfFit = 0
4434 394      angCount = 0
4435 395      booPEF = False
4436 396      * Verify recMarine has values
4437 397      If recMarine.RecordCount > 0 Then
4438 398      * Initialize variables
4439 399      strInput1 = "" & recClassSCourseNumber_FK & ""
4440 400      strInput2 = "" & recClassAMOS_FK & ""
4441 401      intInput = 0
4442 402      * Find all PEFs associated with the current school
4443 403      strSQL = "SELECT PEF_FK FROM SCH_PROP WHERE
4444 404      SCourseNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4445 405      Set recPEF = db1.OpenRecordset(strSQL)
4446 406      If recPEF.EOF = False Then
4447 407      recPEF.MoveLast
4448 408      recPEF.MoveFirst
4449 409      End If
4450 410      * Find mandatory fundamental properties associated
4451 411      with the current school
4452 412      strSQL = "SELECT FPropertyName_FK FROM FUND_SCH_PROP
4453 413      WHERE SCourseNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4454 414      AND Level = " & intInput & ""
4455 415      Set recFMand = db1.OpenRecordset(strSQL)
4456 416      If recFMand.EOF = False Then
4457 417      recFMand.MoveLast
4458 418      recFMand.MoveFirst
4459 419      End If
4460 420      * Find mandatory logical properties associated with
4461 421      the current school
4462 422      strSQL = "SELECT LPropertyName_FK FROM LOG_SCH_PROP
4463 423      WHERE SCourseNumber_FK = " & strInput1 & " AND AMOS_FK = " & strInput2 & ""
4464 424      AND Level = " & intInput & ""
4465 425      Set recLMand = db1.OpenRecordset(strSQL)
4466 426      If recLMand.EOF = False Then
4467 427      recLMand.MoveLast
4468 428      recLMand.MoveFirst
4469 429      End If
4470 430      * Find desired fundamental properties associated with
4471 431      the current school
4472 432      strSQL = "SELECT FPropertyName_FK [Level] FROM
4473 433      FUND_SCH_PROP WHERE SCourseNumber_FK = " & strInput1 & " AND AMOS_FK = " &
4474 434      strInput2 & " AND Level <= " & intInput & ""
4475 435      Set recFDesired = db1.OpenRecordset(strSQL)
4476 436      If recFDesired.EOF = False Then
4477 437      recFDesired.MoveLast
4478 438      recFDesired.MoveFirst
4479 439      End If
4480 440      * Find desired logical properties associated with the
4481 441      current school
4482 442      strSQL = "SELECT LPropertyName_FK [Level] FROM
4483 443      LOG_SCH_PROP WHERE SCourseNumber_FK = " & strInput1 & " AND AMOS_FK = " &
4484 444      strInput2 & " AND Level <= " & intInput & ""
4485 445      Set recLDesired = db1.OpenRecordset(strSQL)
4486 446      If recLDesired.EOF = False Then
4487 447      recLDesired.MoveLast
4488 448      recLDesired.MoveFirst
4489 449      End If
4490 450      * Cycle through all the marines
4491 451      For j = 1 To recMarine.RecordCount
4492 452      * Yield control temporarily to Windows
4493 453      DoEvents
4494 454      * Check to see if marine's PEF is an open
4495 455      If recMarinePEF <> "00" Then
4496 456      booPEF = False
4497 457      If recPEF.EOF = False Then
4498 458      * Check to see if marine's PEF is
4499 459      associated with the school
4500 460      For k = 1 To recPEF.RecordCount
4501 461      If recPEFPEF_FK = recMarinePEF Then
4502 462      booPEF = True
4503 463      End If
4504 464      recPEF.MoveNext
4505 465      Next k
4506 466      recPEF.MoveFirst
4507 467      End If
4508 468      Else
4509 469      booPEF = True
4510 470      End If
4511 471      * Compare marine attributes with the mandatory
4512 472      properties
4513 473      If booPEF = True Then
4514 474      * Check for mandatory fundamental properties
4515 475      If recFMand.EOF = False Then
4516 476      For k = 1 To recFMand.RecordCount
4517 477      * All tests must prove true for the
4518 478      mandatory properties
4519 479      If booMandProp = True Then
4520 480      FundPropTest(recFMandFPropertyName_FK, recMarineSSN_PK) = False Then
4521 481      booMandProp = False
4522 482      End If
4523 483      recFMand.MoveNext
4524 484      End If
4525 485      Next k
4526 486      recFMand.MoveFirst
4527 487      End If
4528 488      * Check for mandatory logical properties
4529 489      If booLMandProp = True Then
4530 490      If recLMand.EOF = False Then
4531 491      For k = 1 To recLMand.RecordCount
4532 492      * All tests must prove true for
4533 493      the mandatory properties
4534 494      If booLMandProp = True Then
4535 495      LogPropTest(recLMandLPropertyName_FK, recMarineSSN_PK) = False Then
4536 496      booLMandProp = False
4537 497      End If
4538 498      recLMand.MoveNext
4539 499      End If
4540 500      Next k
4541 501      End If
4542 502      * Compare with desired fundamental
4543 503      properties
4544 504      If recFDesired.EOF = False Then
4545 505      For i = 1 To recFDesired.RecordCount
4546 506      If
4547 507      FundPropTest(recFDesiredFPropertyName_FK, recMarineSSN_PK) = True Then
4548 508      Select Case recFDesiredLevel
4549 509      Case 1
4550 510      intFit = intFit +
4551 511      Case 2
4552 512      intFit = intFit +
4553 513      Case 3
4554 514      intFit = intFit +
4555 515      Case 4
4556 516      intFit = intFit +
4557 517      Case 5
4558 518      intFit = intFit +
4559 519      Case 6
4560 520      intFit = intFit +
4561 521      End Select
4562 522      End If
4563 523      recFDesired.MoveNext
4564 524      Next i
4565 525      recFDesired.MoveFirst
4566 526      End If
4567 527      * Compare with desired logical properties
4568 528      If recLDesired.EOF = False Then
4569 529      For i = 1 To recLDesired.RecordCount
4570 530      If
4571 531      FundPropTest(recLDesiredLPropertyName_FK, recMarineSSN_PK) = True Then
4572 532      Select Case recLDesiredLevel
4573 533      Case 1
4574 534      intFit = intFit +
4575 535      Case 2
4576 536      intFit = intFit +
4577 537      Case 3
4578 538      intFit = intFit +
4579 539      Case 4
4580 540      intFit = intFit +
4581 541      Case 5
4582 542      intFit = intFit +
4583 543      Case 6
4584 544      intFit = intFit +
4585 545      End Select
4586 546      End If
4587 547      recLDesired.MoveNext
4588 548      Next i
4589 549      recLDesired.MoveFirst
4590 550      End If
4591 551      * Compare with desired logical properties
4592 552      If recLDesired.EOF = False Then
4593 553      For i = 1 To recLDesired.RecordCount
4594 554      If
4595 555      FundPropTest(recLDesiredLPropertyName_FK, recMarineSSN_PK) = True Then
4596 556      Select Case recLDesiredLevel
4597 557      Case 1
4598 558      intFit = intFit +
4599 559      Case 2
4600 560      intFit = intFit +
4601 561      Case 3
4602 562      intFit = intFit +
4603 563      Case 4
4604 564      intFit = intFit +
4605 565      Case 5
4606 566      intFit = intFit +
4607 567      Case 6
4608 568      intFit = intFit +
4609 569      End Select
4610 569      End If
4611 570      recLDesired.MoveNext
4612 571      Next i
4613 572      recLDesired.MoveFirst
4614 573      End If
4615 574      * If the marine's PEF is not satisfied by the
4616 575      school or the marine does not meet
4617 576      * all the mandatory properties for the school, a
4618 577      score of zero is assigned.
4619 578      * Otherwise, a value >= 1 is assigned. A score of
4620 579      1 indicates the school had no
4621 580      * desired properties or the marine did not
4622 581      satisfy any of the desired properties.
4623 582      * Later, 1 is subtracted from these scores and
4624 583      replaced with a small value called
4625 584      * epsilon. The epsilon is used to prevent a
4626 585      division by zero problem.
4627 586      recMarClsFit.Fitness = intFit
4628 587      recMarClsFit.Update
4629 588      recMarClsFit.MoveNext
4630 589      * Prevents counting the entries with zeros since
4631 590      they are removed later
4632 591      * Sums up all the fits and gets a count of how
4633 592      many fits there are
4634 593      If intFit > 0 Then
4635 594      * The 1 is subtracted from the intFit score
4636 595      and replaced with a small epsilon
4637 596      angSumOfFit = angSumOfFit + CSng(intFit - 1)
4638 597      #NAME?
4639 598      angCount = angCount + 1
4640 599      End If
4641 600      intFit = 0
4642 601      recMarine.MoveNext
4643 602      Next j
4644 603      * Present dividing by zero
4645 604      If angCount > 0 Then
4646 605      angAvgFit = angSumOfFit / angCount
4647 606      recMarClsFit.MoveFirst
4648 607      * Update the TEMP_FIT table with the normalized
4649 608      fitness values
4650 609      For k = 1 To recMarClsFit.RecordCount
4651 610      If recMarClsFit.Fitness > 0 Then
4652 611      recMarClsFit.Edit
4653 612      * The 1 is subtracted from the stored
4654 613      fitness score and replaced with a small epsilon
4655 614      angFitness = CSng(recMarClsFit.Fitness -
4656 615      1) + angEpsilon
4657 616      recMarClsFit.Fitness =
4658 617      CSng(angMandFit/Value + angFitness * InverseFactor * angDesiredWeight /
4659 618      recMarClsFit.Update
4660 619      End If
4661 620      recMarClsFit.MoveNext
4662 621      Next k
4663 622      * Insert the TEMP_FIT records into the
4664 623      Application.SeOption "Confirm Action Queries",
4665 624      db1.Execute "INSERT INTO MAR_CLS_FIT (SSN_FK,
4666 625      MarineIndex, ClassIndex_FK, Fitness) SELECT TEMP_FIT.SSN_FK,
4667 626      TEMP_FIT.MarineIndex, TEMP_FIT.ClassIndex_FK, TEMP_FIT.Fitness FROM
4668 627      TEMP_FIT WHERE TEMP_FIT.Fitness > 0;"
4669 628      Application.SeOption "Confirm Action Queries",
4670 629      strClassNumber = recClassSCourseNumber_FK
4671 630      strAMOS = recClassAMOS_FK
4672 631      End If
4673 632      recMarine.MoveFirst
4674 633      If recClass.EOF = False Then
4675 634      recClass.MoveNext
4676 635      End If
4677 636      * If recMarine.RecordCount > 0 Then
4678 637      End If
4679 638      Next i
4680 639      End If

```

```

4680 596
4681 597 recMeCtRtClose
4682 598
4683 599 * Prepare the AMPL_rdm2.dat data file
4684 600 Application.SetOption "Confirm Action Queries", False
4685 601 db1.Execute "DELETE * FROM TEMP_FIT;"
4686 602 * Specify the number of marines
4687 603 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK, MarineIndex) VALUES
4688 (param intMarines * 2 & recMarine.RecordCount & ');";
4689 604 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK) VALUES (');";
4690 605 * Specify the number of school classes
4691 606 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK, MarineIndex) VALUES
4692 (param intClasses * 2 & recClass.RecordCount & ');";
4693 607 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK) VALUES (');";
4694 608 * Specify the class demand
4695 609 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK) VALUES (param demand
4696 610 db1.Execute "INSERT INTO TEMP_FIT (ClassIndex, PK_Fitness) SELECT
4697 CLASS.ClassIndex, CLASS.Quota FROM CLASS;"
4698 611 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK) VALUES (');";
4699 612 * Specify the class penalty
4700 613 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK) VALUES (param penalty
4701 614 db1.Execute "INSERT INTO TEMP_FIT (ClassIndex, PK_Fitness) SELECT
4702 PENALTY.ClassIndex, PK_PENALTY.Penalty FROM PENALTY;"
4703 615 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK) VALUES (');";
4704 616 DoCmd.TransferText acExportFixed, "rdm2_Data Export Specification",
4705 TEMP_FIT, "C:\VDM\AmplData\rdm2.dat"
4706 617
4707 618 * Prepare the AMPL_rdm3.dat data file
4708 619 db1.Execute "DELETE * FROM TEMP_FIT;"
4709 620 db1.Execute "INSERT INTO TEMP_FIT SELECT * FROM MAR_CLS_FIT;"
4710 621 db1.Execute "ALTER TABLE TEMP_FIT DROP COLUMN SSN_LPK;"
4711 622 db1.Execute "ALTER TABLE TEMP_FIT ADD COLUMN SSN_LPK text;"
4712 623 db1.Execute "INSERT INTO TEMP_FIT (SSN_LPK) VALUES (param fitness
4713 624 db1.Execute "SELECT * INTO AMPL_TEMP FROM TEMP_FIT ORDER BY SSN_LPK
4714 625 db1.Execute "INSERT INTO AMPL_TEMP (SSN_LPK) VALUES (');";
4715 626 DoCmd.TransferText acExportFixed, "rdm3_Data Export Specification",
4716 627 AMPL_TEMP, "C:\VDM\AmplData\rdm3.dat"
4717 628 Application.SetOption "Confirm Action Queries", True
4718 629
4719 630 recMarine.Close
4720 631 recClass.Close
4721 632
4722 631 * Remove the created tables
4723 632 db1.Execute "DROP TABLE TEMP_FIT;"
4724 633 db1.Execute "DROP TABLE AMPL_TEMP;"
4725 634 db1.Execute "Drop Table TEMP_FIT1;"
4726 635
4727 635 * Remove the meter from the status bar
4728 637 varRetVal = SysCmd(acSysCmdRemoveMeter)
4729 638
4730 639 End Sub
4731 640
4732 641
4733 642 Public Sub Quota_Penalty(varUdateBound As Variant)
4734 643 Dim varDate
4735 644 Dim rec As Recordset, rec1 As Recordset
4736 645 Dim strSQL As String, strConvert As String
4737 646 Dim i As Integer
4738 647 Dim db1 As Database
4739 648
4740 649
4741 650 * Create new class quota file
4742 651 Application.SetOption "Confirm Action Queries", False
4743 652
4744 653 * Delete the old penalty and class files
4745 654 DoCmd.RunSQL "DELETE * FROM PENALTY"
4746 655 DoCmd.RunSQL "DELETE * FROM CLASS"
4747 656
4748 657 * Query generates new class quota
4749 658 DoCmd.SetWarnings (False)
4750 659 DoCmd.OpenQuery "qryNewQuota"
4751 660 DoCmd.SetWarnings (True)
4752 661
4753 662 * Delete any values < 0
4754 663 DoCmd.RunSQL "DELETE * FROM CLASS WHERE Quota <= 0;"
4755 664
4756 665 Application.SetOption "Confirm Action Queries", True
4757 666
4758 667 * Create index for the class table and format the FY for RD3 file
4759 668 Set db1 = CurrentDb()
4760 669 strSQL = "SELECT CLASS, SCHOOL.PenaltyFactor FROM SCHOOL INNER JOIN
4761 670 CLASS ON (SCHOOL.AMOS_PK = CLASS.AMOS_PK AND (SCHOOL.SCourseNumber_PK =
4762 671 CLASS.CourseNumber_PK)"
4763 671 Set rec = db1.OpenRecordset(strSQL, dbOpenDynaset)
4764 672 rec.MoveLast
4765 673
4766 673 If rec.RecordCount > 0 Then
4767 674 rec.MoveFirst
4768 675 For i = 1 To rec.RecordCount
4769 676 rec.Edit
4770 677 rec.ClassIndex = i
4771 678 * Converts the fiscal year to a two digit number for RD3 file
4772 679 strConvert = CSng(rec.FiscalYear_PK)
4773 680 strConvert = Right(strConvert, 2)
4774 681 rec.FiscalYear_PK = CInt(strConvert)
4775 682 rec.Update
4776 683 rec.MoveNext
4777 684 Next i
4778 685 End If
4779 686
4780 687 * Create penalty file
4781 688 * First, update the index from the CLASS table
4782 689 Application.SetOption "Confirm Action Queries", False
4783 690 DoCmd.OpenQuery "qryPenaltyIndex"
4784 691 Application.SetOption "Confirm Action Queries", True
4785 692 * Create a record set of the PENALTY table
4786 693 strSQL = "SELECT * FROM PENALTY"
4787 694 Set rec1 = db1.OpenRecordset(strSQL, dbOpenDynaset)
4788 695 If IsNull(varUdateBound) = False Then
4789 696 * Format this date for use in the DateDiff function
4790 697 varUdateBound = Format(varUdateBound, "###/##/##")
4791 698
4792 699 Module: modFitnessDetermination Page: 16
4793 699
4794 700 If rec.RecordCount > 0 Then
4795 701 For i = 1 To rec.RecordCount
4796 702 rec1.Edit
4797 703 * Get report date from class record
4798 704 varDate = rec.ReportDate
4799 705 * Format the report date for use in the DateDiff function
4800 706 varDate = Format(varDate, "###/##/##")
4801 707 * Make the penalty difference between the two values
4802 708 * This makes the penalty linear, based on date
4803 709 rec1.Penalty = DateDiff("d", varDate, varUdateBound) *
4804 710 rec1.PenaltyFactor / 24
4805 711 rec1.Update
4806 712 rec.MoveNext
4807 713 rec1.MoveNext
4808 714 Next i
4809 714 End If

```

```

4810 715 End If
4811 716
4812 717 End Sub
4813 718
4814 719 Public Sub DeleteTable()
4815 720 On Error GoTo DeleteTable_Err
4816 721
4817 722 Dim db1 As Database
4818 723
4819 724 Set db1 = CurrentDb()
4820 725 * Delete the table if it already exists
4821 726 db1.TableDefs.Delete "PERCENTAGE"
4822 727 DeleteTable_Exit
4823 728 Exit Sub
4824 729
4825 730 DeleteTable_Err:
4826 731 Select Case Err.Number
4827 732 ' This number occurs if the PERCENTAGE table does not exist.
4828 733 Case 3265
4829 734 Resume DeleteTable_Exit
4830 735 Case Else
4831 736 MsgBox Err.Description
4832 737 Resume DeleteTable_Exit
4833 738 Resume
4834 739 End Select
4835 740
4836 741 End Sub
4837 742
4838 743 Module: modImportExportData
4839 744 Code
4840 745 1 Attribute VB_Name = "modImportExportData"
4841 746 2 Option Compare Database
4842 747 3 Option Explicit
4843 748 4
4844 749 5 Public Sub ImportBNA()
4845 750 6
4846 751 DoCmd.TransferText acImportFixed, "BNA_Extract", "qryBNA_Extract",
4847 752 C:\ToBeRDM\rdmData\BNA.bt
4848 753
4849 754 8 End Sub
4850 755 9
4851 756 10 Public Sub ImportRD1()
4852 757 11
4853 758 12
4854 759 13 DoCmd.TransferText acImportFixed, "RD1", "qryMarine",
4855 760 C:\ToBeRDM\rdmData\RD1.bt
4856 761
4857 762 14 End Sub
4858 763 15
4859 764 16
4860 765 17 Public Sub ImportTargetMOS() (ActionQuery As String)
4861 766 18
4862 767 19 CurrentDb().Execute "INSERT INTO TARGET_MOS ( CourseNumber_PK,
4863 768 TargetMOS_PK, MCC) SELECT DISTINCT BNA_EXTRACT.CourseNumber_PK,
4864 769 BNA_EXTRACT.TargetMOS_PK, BNA_EXTRACT.MCC FROM BNA_EXTRACT"
4865 770 20
4866 771 21 End Sub
4867 772 22
4868 773 23 Public Sub AmpL_Result()
4869 774 24
4870 775 25 DoCmd.TransferText acImportDelim, "Ampl_Result", "AMPL_RESULT",
4871 776 C:\ToBeRDM\AmplData\rdm.out, True
4872 777 26
4873 778 27 End Sub
4874 779 28
4875 780 29 Public Sub AmpDate()
4876 781 30 Dim db1 As Database
4877 782 31
4878 783 32 Set db1 = CurrentDb()
4879 784 33
4880 785 34 * Prepare the AMPL_rdm2.dat data file
4881 786 35 Application.SetOption "Confirm Action Queries", False
4882 787 36 db1.Execute "CREATE TABLE CONSTANTS (ConstantName Text, InValue
4883 788 37 * Specify the fit constant
4884 789 38 db1.Execute "INSERT INTO CONSTANTS (ConstantName, InValue) VALUES
4885 790 (param fitConstant * 2 & Format(fitConstant, "###/##/##") &
4886 791 * Specify the fit constant
4887 792 db1.Execute "INSERT INTO CONSTANTS (ConstantName, InValue) VALUES
4888 793 (param fitConstant * 2 & Format(fitConstant, "###/##/##") &
4889 794 DoCmd.TransferText acExportFixed, "rdm2_Data Export Specification",
4890 795 CONSTANTS, "C:\ToBeRDM\AmplData\rdm.dat"
4891 796 44 Application.SetOption "Confirm Action Queries", True
4892 797 45
4893 798 46 * Remove the created table
4894 799 47 db1.Execute "DROP TABLE CONSTANTS;"
4895 800 48 db1.Close
4896 801 49 End Sub
4897 802 50
4898 803 51 Public Sub ExportRD3()
4899 804 52
4900 805 53 DoCmd.TransferText acExportFixed, "ExportRD3File", "qryRD3File",
4901 806 C:\ToBeRDM\rdmData\RD3.bt
4902 807 54 Application.SetOption "Confirm Action Queries", False
4903 808 55 DoCmd.OpenQuery "qryUpdateAssignmentType"
4904 809 56 DoCmd.OpenQuery "qryAppendResultToAssignment"
4905 810 57 Application.SetOption "Confirm Action Queries", True
4906 811 58
4907 812 59
4908 813 60 End Sub
4909 814 61
4910 815 62 Public Sub ArchiveData()
4911 816 63
4912 817 64 Dim varTodaysDate
4913 818 65 Dim strSQL As String
4914 819 66
4915 820 varTodaysDate = Date
4916 821 68
4917 822 69 * Format the date
4918 823 varTodaysDate = Format(varTodaysDate, "yyyymmdd")
4919 824 70 * Convert the date into a number
4920 825 varTodaysDate = CLng(varTodaysDate)
4921 826 71
4922 827 72 * Archive the applicable data
4923 828 73 DoCmd.RunSQL "INSERT INTO MARINE_ASSIGN_ARCHIVE SELECT DISTINCTROW
4924 829 MARINE.SSN_LPK, MARINE.LName, MARINE.Initials, MARINE.Gender,
4925 830 MARINE.DriverLicense, MARINE.SCI, MARINE.TrainingCompany,
4926 831 MARINE.MFO_Code, MARINE.PINNumber, MARINE.Citizenship, MARINE.GradDate,
4927 832 MARINE.SwimQual, MARINE.Obligation, MARINE.Age, MARINE.Education,
4928 833 MARINE.HighestMath, MARINE.MentalAbility, MARINE.ClearanceEst,
4929 834 MARINE.Height, MARINE.Vision, MARINE.ColorPerception, MARINE.EthnicGroup,
4930 835 MARINE.DUI, MARINE.EDFIT, MARINE.DeviantCode, MARINE.Type, MARINE.EL,
4931 836 MARINE.IVA, MARINE.CL, MARINE.GT, MARINE.Component, MARINE.PEF,
4932 837 MARINE.CivConviction, MARINE.RSN, MARINE.AFADBO,
4933 838 ASSIGNMENT.CourseNumber, ASSIGNMENT.AMOS, ASSIGNMENT.MCC,
4934 839 ASSIGNMENT.ClassNumber, ASSIGNMENT.ReportDate, ASSIGNMENT.FiscalYear,
4935 840 ASSIGNMENT.AssignmentType FROM MARINE INNER JOIN ASSIGNMENT ON
4936 841 MARINE.SSN_LPK = ASSIGNMENT.SSN_LPK WHERE ASSIGNMENT.ReportDate <= " &
4937 842 varTodaysDate & "
4938 843 74
4939 844 75 * Delete the applicable data from the Marine and Assignment tables
4940 845 76 DoCmd.RunSQL "DELETE DISTINCTROW MARINE, ASSIGNMENT FROM MARINE"

```

```
4939 78 DoCmd.RunSQL "DELETE DISTINCTROW MARINE.* FROM MARINE
4940 INNER JOIN ASSIGNMENT ON MARINE.SSN_PK = ASSIGNMENT.SSN_FK WHERE
4941 ASSIGNMENT.ReportDate < " & varTodaysDate & "
4942 79
4943 80 Select Case MsgBox("Delete unassigned Marines from the database?",
4944 vbYesNo + vbQuestion, "Delete Unassigned Marines")
4945 81 Case 6 ' If yes is clicked
4946 82 ' Deletes entries in the Marine table without a corresponding
4947 entry in the Assignment table
4948 83 DoCmd.RunSQL "DELETE DISTINCTROW MARINE.* FROM MARINE
4949 INNER JOIN ASSIGNMENT ON MARINE.SSN_PK = ASSIGNMENT.SSN_FK
4950 WHERE (((ASSIGNMENT.SSN_FK) Is Null)):"
4951 84 Case 7 ' If no is clicked
4952 85 ' No action taken
4953 86 End Select
4954 87
4955 88 End Sub
```

LIST OF REFERENCES

- [1] USMC, Manpower Management Information Systems Branch, Manpower and Reserve Affairs Department. "Statement of Work for the USMC Manpower Model Modernization: Reengineering of the Enlisted Assignment Model and Recruit Distribution Model," September, 1997.
- [2] Decision Support Associates Inc. *Users Manual-Recruit Distribution Model*. Developed In-house, Decision Support Associates Inc., 1993.
- [3] C.J. Date. *An Introduction to Database Systems*. Reading, MA: Addison-Wesley, 1992.
- [4] J. Dumas. *Designing User Interfaces for Software*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [5] CPLEX, A Division of ILOG. CPLEX Home Page, 1998. [<http://www.cplex.com>].
- [6] R.H. Sprague. "A Framework for the Development of Decision Support Systems." *MIS Quarterly* 4, Pages 1-26, 1980.
- [7] Logic Works, Inc. "Logic Works BPWin Tutorial Guide," 1994-1995.
- [8] E.F. Codd. "A Relational Model of Data for Large Shared Databanks." *Communications of the ACM* 13, June 1970.
- [9] R. Smith, D. Sussman. *Beginning Access 97 VBA Programming*. Wrox Press, 1997.
- [10] R. Fourer, D. Gay, B. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Boyd & Fraser Publishing Company, 1993.
- [11] B. Render, R. Stair. *Quantitative Analysis for Management*. Prentice-Hall Inc., 1997.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
8725 John J. Kingman Rd., Ste 0944
Fort Belvoir, VA 22060-6218
2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Rd.
Monterey, CA 93943-5101
3. Professor Hemant Bhargava (Code SM/BH) 7
Systems Management Department
Naval Postgraduate School
Monterey, CA 93940-5000
4. Professor Suresh Sridhar (Code SM/SR) 1
Systems Management Department
Naval Postgraduate School
Monterey, CA 93943-5000
5. Professor Daniel Dolk (Code SM/Dk) 1
Systems Management Department
Naval Postgraduate School
Monterey, CA 93943-5000
6. LT Kevin J. Snoap 2
541 Westway Drive N.W.
Grand Rapids, MI 49544