AD-A256 835

(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
ELECTE
NOV 06 1992
S E D

THESIS

DESIGN AND IMPLEMENTATION OF A PATIENT TRACKING AND RECALL SYSTEM FOR BRANCH DENTAL CLINIC MONTEREY

by

Timothy P. Steele

March 1992

Thesis Advisor: Hemant K. Bhargava

Approved for public release; distribution is unlimited.

92-29052

257456 121 pp

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL *(If applicable)* Code 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000		7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS			
	Program Element No.	Project No.	Task No.	Work Unit Accession Number

11 TITLE *(Include Security Classification)*

DESIGN AND IMPLEMENTATION OF A PATIENT TRACKING AND RECALL SYSTEM FOR BRANCH DENTAL CLINIC MONTEREY

12 PERSONAL AUTHOR(S) Steele, Timothy P.

13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED From To	14 DATE OF REPORT *(year, month, day)* 1992, March	15 PAGE COUNT 127

16 SUPPLEMENTARY NOTATION

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

17 COSATI CODES			18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUBGROUP	Information System Analysis; Software Development; Patient Tracking and Recall; Relational Database

19. ABSTRACT *(continue on reverse if necessary and identify by block number)*

This thesis analyses the information system requirements of Branch Dental Clinic, Monterey, and develops a computer application to automate the clinic's patient tracking and recall process. The application replaces an existing mainframe-based, single-file system with a PC-based, relational database management system that provides greater functionality, enables increased productivity, improves data integrity and accuracy, and includes currently lacking security features and administrative functions.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Hemant K. Bhargava	22b. TELEPHONE *(Include Area code)* (408) 646-2264	22c OFFICE SYMBOL Code AS/BH

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

Approved for public release: distribution is unlimited

Design and Implementation of a Patient Tracking and Recall System for Branch Dental Clinic Monterey

by

Timothy P. Steele
Lieutenant Commander, United States Navy
B.A., University of Washington, 1979

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1992

Author: Timothy P. Steele
Timothy P. Steele

Approved by: HKBhargava
Hemant K. Bhargava, Thesis Advisor

Dani Zweig, Second Reader

David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

This thesis analyzes the information system requirements of Branch Dental Clinic, Monterey, and develops a computer application to automate the clinic's patient tracking and recall process. The application replaces an existing mainframe-based, single-file system with a PC-based, relational database management system that provides greater functionality, enables increased productivity, improves data integrity and accuracy, and includes currently lacking security features and administrative functions.

Accesion For
NTIS CRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

DTIC QUALITY INSPECTED 4

TABLE OF CONTENTS

I. INTRODUCTION AND PRELIMINARY INVESTIGATION 1

A. BACKGROUND 1

B. PROBLEM DEFINITION 3

C. SCOPE 5

D. EVALUATION OF ALTERNATIVE SOLUTIONS 6

1. Cost Feasibility 6

2. Technical Feasibility 7

3. Schedule Feasibility 8

II. REQUIREMENTS ANALYSIS 9

A. DATA REQUIREMENTS 9

1. Object Development 9

2. Domain Definition 11

B. APPLICATION REQUIREMENTS 11

1. Processes 11

2. Operational and Administrative Requirements 13

3. Environmental Requirements 14

III. SYSTEM DESIGN 15

A. LOGICAL DATABASE DESIGN 15

1. Object to Relation Transformations 15

2. Relation Descriptions 16

B. APPLICATION DESIGN 17

IV. SYSTEM IMPLEMENTATION 21

V. SUMMARY AND RECOMMENDATIONS 22

A. SUMMARY 22

B. RECOMMENDATIONS 22

LIST OF REFERENCES 24

APPENDIX A: OBJECT SPECIFICATIONS 25

APPENDIX B: UPDATE, DISPLAY, AND CONTROL MECHANISMS 28

APPENDIX C: USER'S MANUAL 34

APPENDIX D: RELATION DEFINITIONS 35

APPENDIX E: PROGRAM CODE 36

INITIAL DISTRIBUTION LIST 37

ACKNOWLEDGEMENT

Special acknowledgement must be given to my wife, Anne, for her support and forbearance as I spent long hours at the computer, seemingly oblivious to the meaning of life, working on this thesis. I am truly in her debt.

I. INTRODUCTION AND PRELIMINARY INVESTIGATION

This thesis designs, documents, and implements a computer application to perform dental patient tracking and recall functions for the Branch Dental Clinic, Monterey (BDCM). Information that was collected during a preliminary investigation of the information system requirements of BDCM is presented in this chapter. Specifically, the relevant background of BDCM and the information system problems that led to the conduct of the thesis are presented, the scope of the project is defined, and three alternative solutions are evaluated.

A. BACKGROUND

BDCM provides regular dental care and emergency dental treatment to all active duty military staff and students stationed both at the Naval Postgraduate School (NPS) and the various NPS tenant commands. Dental appointments are regularly scheduled based on a four-class rating system (1 to 4, in order of increasing priority) indicating the member's need for treatment. Emergency care is provided whenever required.

Interviews with the BDCM Director and staff identified four major information-oriented activities within the clinic: (1) appointment scheduling, (2) inventory management, (3) maintenance of a Dental Information and Retrieval System (DIRS) as prescribed by higher authority, and (4) patient tracking and recalls. With regard to appointment scheduling and inventory management activities, BDCM satisfaction with current manual methods was found to be high. Moreover, the clinic Director felt strongly that attempts to computerize these two functions, given the relatively low volume of activity, would not increase efficiency or effectiveness. Hence, these two business functions were dropped from further investigation.

The DIRS system operates on a personal computer (PC) and consists of proprietary software provided by the Navy Regional Dental Center (NRDC) for use at all subordinate branch clinics. Since NRDC mandates that branch clinics use DIRS to collect and report detailed data on all dental care provided, further analysis of this activity was unnecessary.

Patient tracking and recall functions at BDCM are partially automated by a mainframe-based, single-user, single-file database management system. It is this system and the requirements of the patient tracking and recall process that the remainder of this thesis addresses.

The mainframe-based database application allows data entry and updating, tracks members' dental health status (class), generates recall notices, prints sorted member rosters, and provides operational readiness summary statistics. When members check their records into the clinic a dentist's review of their dental records results in a class rating being assigned. A class rating of "1" indicates no need for dental treatment beyond a mandatory annual examination (a T2-exam). A class rating of "2" or "3" indicates a need for additional treatment. A class rating of "4" indicates the member is past due for an annual exam (it is assigned regardless of dental health). Just prior to a member's T2-exam anniversary, he/she is notified by memorandum to make an appointment for an annual exam using an automated patient recall system. Computer generated recall letters are routed to student mail center (SMC) mailboxes or staff offices as appropriate.

B. PROBLEM DEFINITION

The existing application for patient recalls was written several years ago for use on the NPS mainframe computer. When the system was installed it provided significant benefits over the previous labor and time intensive manual recall process. However, the system was crude in its interface, limited in functionality, and difficult to use. Moreover, due to turnover of personnel since its installation, none of the current staff are familiar with the history of the system; no documentation can be found; and no system maintenance is available.

Interviews with end-users revealed five general problem areas with the mainframe-based system: poor access and responsiveness, unfriendly user-interface, inadequate data validation checks, absence of documentation, and incomplete functionality. Examples of specific problems highlighted by end-users in each of these general areas are presented below.

Limited mainframe access and poor responsiveness have been longstanding limitations. BDCM access to the mainframe is via communications software and 1200 baud modem from the clinic PC. By today's standards, this data transfer rate is slow. The system frequently responds slowly during working hours due to both the high number of users and resource-intensive computing tasks. Heavy use of the mainframe by modem users combined with the limited number of modem receiving lines (16 at the time of this investigation) results in the frequent inability to access the system as needed. This necessitates periodic off-hour work by BDCM staff and delays response to telephone queries from NRDC regarding operational readiness.

Unfriendliness of the user-interface is a significant problem, particularly for new users. In most instances the user is presented with only a blank screen and a prompt, which specifies which application module is active (e.g., main, add, edit, delete, print). A rudimentary help function, when invoked, provides a list of options for the active module. Hence, unless all commands are memorized, the user must continuously invoke the help function to navigate and use the system. Data entry itself is facilitated somewhat by a field list from which the user selects a field to enter or edit, but it remains a cumbersome process. The user must select a field from the list, enter the data, and select another field from the list rather than simply automatically moving from one field to the next. Additionally, during record appending the field listing scrolls up and off the screen, leaving no hint of the remaining fields that require additional data entry.

The inadequacy of field validation checks in the mainframe application has allowed a cumulative deterioration in the accuracy and completeness of records in the database. For example, numbers are improperly allowed in various name fields. Moreover, since member records are indexed by name rather than Social Security Number (SSN), two people with the same name are prohibited from being entered properly into the database. In such instances, the user must deliberately attempt to circumvent or "trick" the system by, for example, putting in a middle initial for one member but not the other. Related to this, the system saves a new record whenever data is entered into the name field, regardless of content and regardless of whether the record has any other fields completed. Over time the database has accumulated much erroneous data and many incomplete records. Cleaning the database has been problematic since records cannot be located and edited or deleted unless an exact name match is entered.

The lack of system documentation has forced end-users to learn the system by experimentation. The total functionality of the system is not immediately obvious and can remain undiscovered and unutilized. Moreover, the logic underlying critical processes, such as the triggering of recall notices or updating dental class status remains unspecified. The lack of documentation has also precluded improving the functionality of the system and implementing fixes. For example, necessary follow-up form letters that are not included in the present system must be externally word-processed for each individual. Additionally, hard-coding of the signature name on recall letters has resulted in a long since-transferred Director's name appearing on the recalls sent to members.

C. SCOPE

The scope of this thesis is limited to the patient tracking and recall process. As noted previously, there are other business functions within the clinic, yet the patient tracking and recall process is the only information-intensive business function left up to local implementation that remains problematic.

D. EVALUATION OF ALTERNATIVE SOLUTIONS

Given that the problems with the existing patient tracking and recall system were deemed significant enough to warrant remediation, three alternative solutions were evaluated. The first alternative involved improving both the hardware and software associated with the mainframe-based system: replacing the modem connection with an on-line terminal, rewriting the software for increased functionality and ease of use, and documenting the system. The second and third alternatives involved designing and implementing a PC-based database management system to replace the existing mainframe application, the difference being whether a multi-user versus a single-user configuration should be developed. Multi-user capability was considered a "nice-to-have" feature that might be useful sometime in the future, yet it was clearly not a requirement for satisfactory performance of patient tracking and recall functions. Should a PC-based solution be selected, NRDC stipulated that it must be a compiled application that would not be subject to potential modification by inexperienced clinic staff.

1. Cost Feasibility

At the outset, NRDC made it clear that no funds were available to support improving the existing patient tracking and recall system. This limitation alone ruled-out upgrading the mainframe-based system—the cost of terminal acquisition and connection was prohibitive. Moreover, additional funds would be required to pay a technical expert to rewrite and document the mainframe software. Similarly, to exploit multi-user capability in a PC-based system would require additional funding to purchase required hardware. Hence, these two alternatives were eliminated from further consideration.

Designing and implementing a single-user, PC-based database management system was attractive from a cost standpoint. The development cost of such a system would be limited to the personal time and effort of the author. Further, appropriate development hardware (an IBM-compatible 80386 computer) and software (Foxpro 2.0 and Foxpro 2.0 Distribution Kit, a dBase-compatible development system with compiler) was already owned by the author. In addition, BDCM would not be required to purchase any additional hardware; their existing computer equipment could be used to evaluate prototypes and to install the final working system. BDCM staff were enthusiastic and committed to assisting with the development process.

2. Technical Feasibility

BDCM owned a Zenith 286 PC and peripherals that were compatible with the foreseeable processor, memory, storage, and video requirements of a new PC-based application. Moreover, Foxpro 2.0 can create applications able to run on any IBM-compatible PC with a minimum of 512K of random access memory (RAM) [Ref. 1]. Preliminary tests of routine database operations (browse, index, sort) with a test database approximately the same size as that of the existing mainframe data file (2000 records with 15 fields per record) using Foxpro 2.0 were successful on the BDCM PC and demonstrated acceptable speed of operations with only 512K of RAM.

Future maintenance of the application would not be provided by the author. Discussion of this issue with both NRDC and BDCM indicated that this was acceptable to them. It was agreed that the application should run on any minimally configured IBM-compatible computer to enable portability and that support for a standard dot-matrix printer should be provided. Program code and documentation would be included with the delivered application to support future maintenance. (NRDC and BDCM acknowledged that any future maintenance would require purchase of Foxpro 2.0 and the Foxpro 2.0 Distribution Kit. Intermediate-level dBase or Foxpro programming skills would also be required.)

3. Schedule Feasibility

Based on the findings of the preliminary investigation, with detailed system analysis to begin 15 August, 1991, implementation of a fully operational PC-based system was scheduled for completion by 1 February, 1992. This left two months for correction of unforseen problems before departure of the author.

II. REQUIREMENTS ANALYSIS

This chapter discusses the requirements phase of project development. The purpose of this phase of development was twofold: (1) during this phase the specific data requirements (objects) that must be represented in the database were defined and (2) the application or functional requirements which support the database were outlined.

A. DATA REQUIREMENTS

Initially, interviews were conducted with the BDCM Director and the dental staff responsible for hands-on use of the existing database. These interviews provided a general idea of the scope and objectives for an upgraded patient tracking and recall system. Working backwards from the existing application's outputs, preliminary object specifications and views were then developed and presented to the dental staff for feedback. Further discussions led to adjustments of the object specifications that satisfactorily met the clinic's needs.

1. Object Development

Important entities identified in the patient tracking and recall process are represented as the objects MEMBER, ACTIVITY, and CURRICULUM shown in Figure 1 below. Each of the objects possesses a collection of named properties. The properties listed within each diagram that are capitalized and within small boxes are themselves objects. The subscript "MV" denotes that the property is multi-valued. The MEMBER object represents patients who have "checked-in" with the clinic upon arrival to NPS or an NPS tenant command. As can be seen in Figure 1, the ACTIVITY and CURRICULUM objects are properties of the MEMBER object. They associate each member with the properties of a specific activity and/or curriculum.

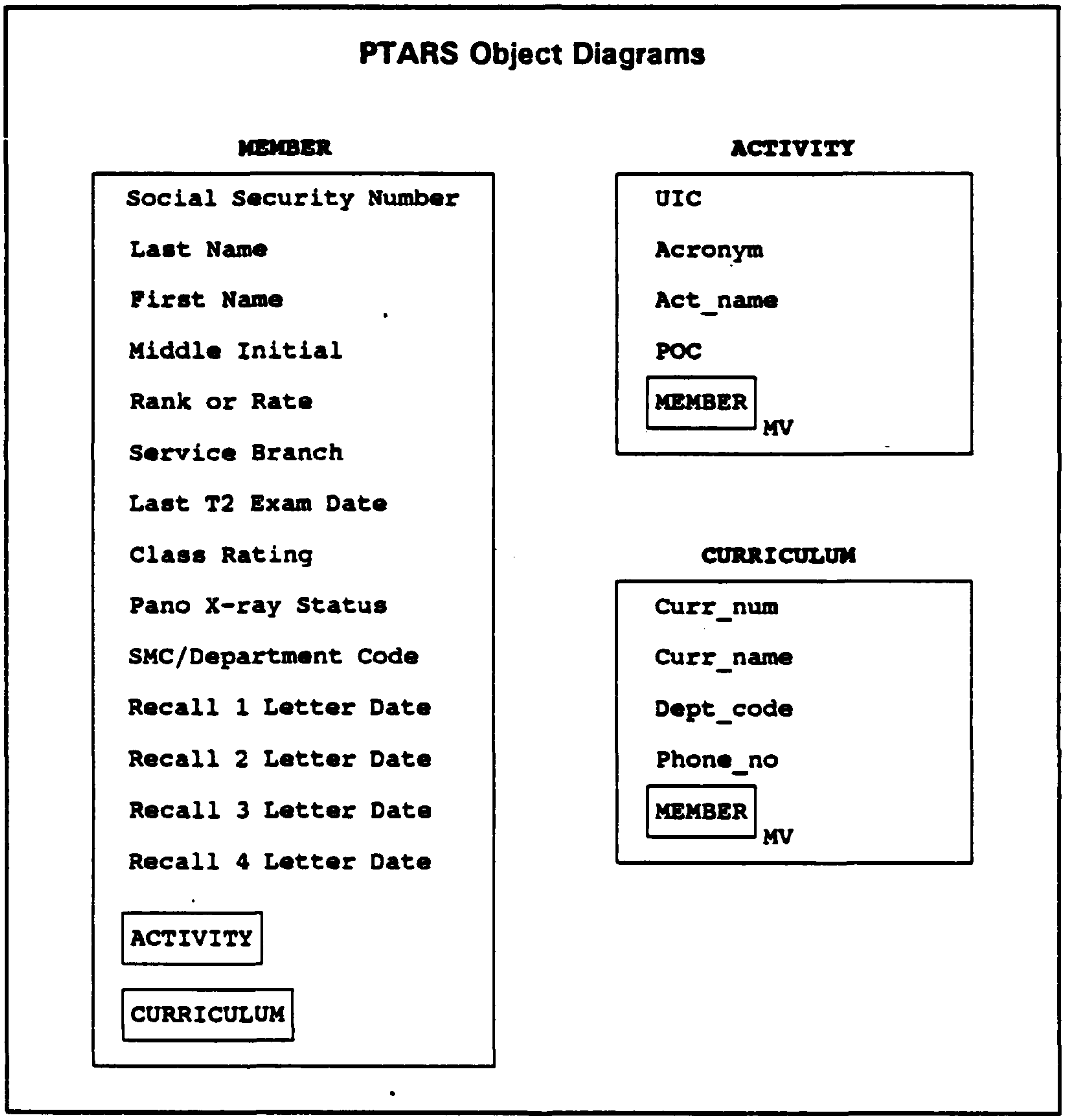

PTARS Object Diagrams
MEMBER
Social Security Number
Last Name
First Name
Middle Initial
Rank or Rate
Service Branch
Last T2 Exam Date
Class Rating
Pano X-ray Status
SMC/Department Code
Recall 1 Letter Date
Recall 2 Letter Date
Recall 3 Letter Date
Recall 4 Letter Date
ACTIVITY
CURRICULUM
ACTIVITY
UIC
Acronym
Act_name
POC
MEMBER
MV
CURRICULUM
Curr_num
Curr_name
Dept_code
Phone_no
MEMBER
MV

Figure 1. Object Diagrams

The ACTIVITY object represents each of the various commands served by BDCM. Note that the multi-valued MEMBER object is also a property of the ACTIVITY object. That is, a specific ACTIVITY can have multiple members.

The CURRICULUM object represents each of the many different curriculums offered at NPS. The MEMBER object is a multi-valued property of the CURRICULUM object; many students can belong to any given curriculum.

2. Domain Definition

The object diagrams were used to summarize knowledge of the objects and to present it to the users in an unambiguous fashion. Following user validation of the object representations, domain definitions were established. The domain of a property is the set of all possible values a property can have. Each domain definition contains a physical description of the type of data (e.g., numeric versus character) and any value constraints. Each definition also describes the function or purpose of the property. Refer to Appendix A for detailed object specifications, including object and domain definitions.

B. APPLICATION REQUIREMENTS

1. Processes

Building upon the data requirements discussed in the previous section, major processes within the patient tracking and recall process were identified through discussions with BDCM end-users. A level-1 data flow diagram (DFD), shown in Figure

2 below, was developed as a basis for validating analyst understanding of the processes with end-users.

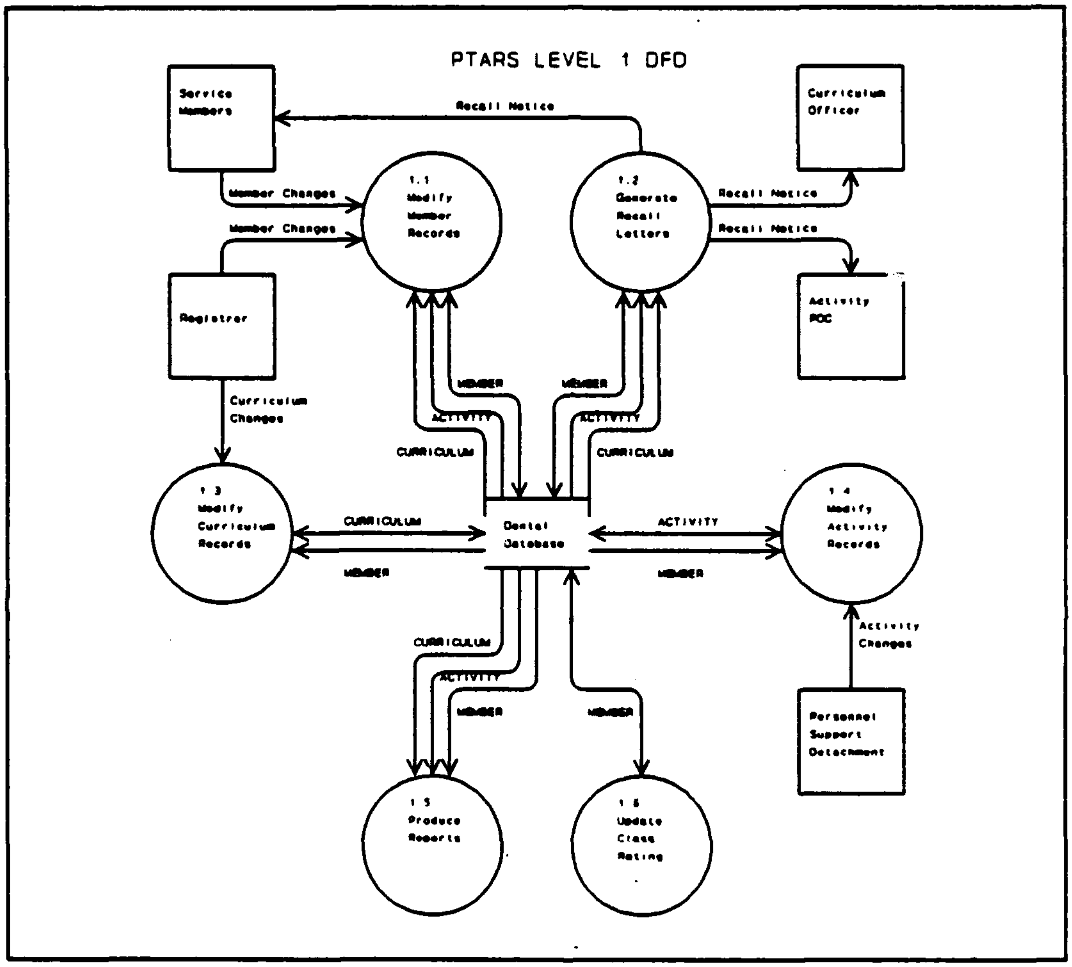

PTARS LEVEL 1 DFD
Service Members
Recall Notice
Curriculum Officer
Member Changes
Member Changes
1.1 Modify Member Records
1.2 Generate Recall Letters
Recall Notice
Recall Notice
Registrar
Activity POC
MEMBER
MEMBER
ACTIVITY
ACTIVITY
CURRICULUM
CURRICULUM
Curriculum Changes
1.3 Modify Curriculum Records
CURRICULUM
MEMBER
Dental Database
ACTIVITY
MEMBER
1.4 Modify Activity Records
CURRICULUM
ACTIVITY
MEMBER
MEMBER
Activity Changes
Personnel Support Detachment
1.5 Produce Reports
1.6 Update Class Rating

Figure 2. Level 1 Data Flow Diagram

Entities external to the system are shown in Figure 2 as square boxes and include Service Member, Registrar, Personnel Support Detachment (PSD), Curriculum Officer, and Activity Point of Contact. These entities are sources of data and/or recipients of system outputs (as indicated by the direction of the data flow arrows). The numbered processes (denoted within the circles) summarize the operations involved in the overall patient tracking and recall process. Processes 1.1, 1.3, and 1.4 comprise the append, edit, and delete operations for the objects, MEMBER, CURRICULUM, and ACTIVITY. Process 1.2 involves the operations associated with generating and printing recall letters. An Operational Readiness report and various sorted rosters are produced in process 1.5. Member dental class is automatically updated to class 4 in process 1.6 for those individuals who have not had an annual examination within 12 months.

Following validation of the information presented by the level-1 DFD, a summary of system update, display, and control mechanisms was developed based on structured interviews with end-users. (See Appendix B.) During this process, information pertaining to each object was obtained that included inputs, outputs, processing notes, volume, and frequency. This information clarified what must be done within each object view.

Prototypes of forms, reports, recall letters, and menus were developed using Foxpro "power tools" (i.e., the Screen Builder and the Report Writer). These early prototypes clarified the expectations of end-users regarding the format of the user-interface and the display of information.

2. Operational and Administrative Requirements

System operational and administrative requirements were identified through discussions with BDCM staff. Operational requirements for the system are listed below:

- Single-user, PC-based application, operable on an "as needed" basis by the BDCM Administrative Petty Officer and/or the BDCM Receptionist

- Portable/re-installable to different, compatible PC
- Extensive "Help" available on-line
- Database backup/restore utilities
- System date and time change utilities
- System-access password protection; password change capability
- Database packing capability

Although it was agreed that program maintenance would not be possible with the compiled application, Foxpro 2.0 program code would be given to BDCM. Hence, should maintenance become critical at some point, modification of the application would be possible with the purchase of Foxpro 2.0 and the Foxpro 2.0 Distribution Kit. A User's Manual (see Appendix C) would be supplied to provide structured guidance for system use, data security and integrity, database backups and restorations, and system optimization.

3. Environmental Requirements

In an efficient system much of the member, activity, and curriculum data should be provided from a master database, shared with the Registrar and PSD. However, this is currently not possible since the data structure and hardware are not compatible. Until such time as the various NPS support entities/ADP-systems can communicate directly, it is incumbent upon the BDCM staff to take the initiative to obtain updated, hard-copy rosters from these two data sources as they become available.

III. SYSTEM DESIGN

In this chapter the two components of system design, logical database design and application design, are discussed. The objective of the design phase was to produce both the logical and physical details of the database and its application. Designing the logical database involved developing a "blueprint" of the database structure. From this blueprint a physical database was designed and the application was developed.

A. LOGICAL DATABASE DESIGN

1. Object to Relation Transformations

The design of the logical database was based on the relational database model [Ref. 2]. The objects MEMBER, ACTIVITY, and CURRICULUM, were transformed into a relational diagram. Figure 3, the relational diagram, shows the three relations that resulted: (1) the compound MEMBER object was transformed into the three relations MEMBER, ACTIVITY, and CURRICULUM; (2) the compound ACTIVITY object was transformed into the two relations MEMBER and ACTIVITY; and (3) the compound CURRICULUM object was transformed into the two relations MEMBER and CURRICULUM.

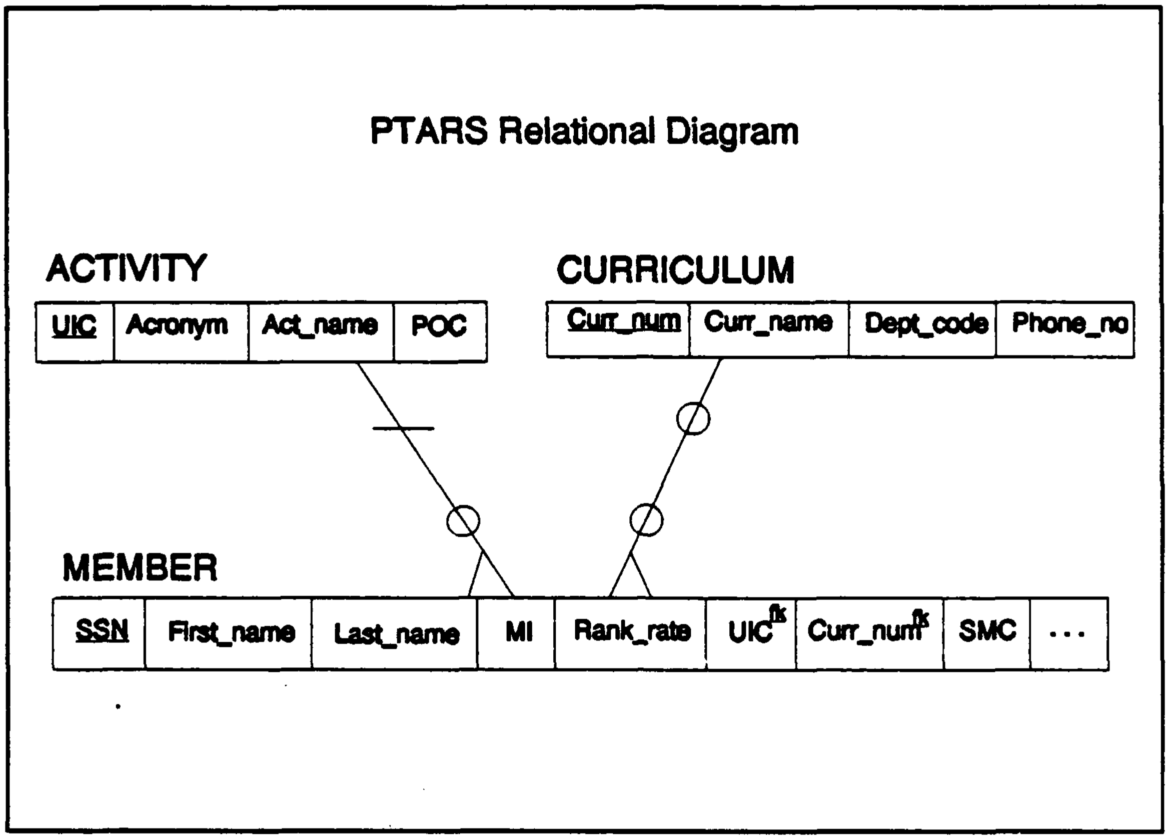

PTARS Relational Diagram
ACTIVITY
UIC
Acronym
Act_name
POC
CURRICULUM
Curr_num
Curr_name
Dept_code
Phone_no
MEMBER
SSN
First_name
Last_name
MI
Rank_rate
UIC fk
Curr_num fk
SMC
...

Figure 3. Relational Diagram

2. Relation Descriptions

Each of the three relations are reflections of the original objects with appropriate foreign keys included. Key data are denoted in Figure 3 by underlining. Foreign keys are denoted with the underlined superscript, $^{\underline{fk}}$. Summary descriptions of each of the relations are presented below. (Refer to Appendix D for detailed relation definitions.)

MEMBER

Number of attributes: 15
Key attributes: Social-Security-Number (SSN)
Foreign keys: Unit-Identification-Code (UIC)
Curriculum-Number

Relationships: ACTIVITY to MEMBER; 1:N; Mandatory:Optional
CURRICULUM to MEMBER; 1:N; Optional:Optional

ACTIVITY
Number of attributes: 4
Key attributes: UIC
Foreign keys: None
Relationships: ACTIVITY to MEMBER; 1:N; Mandatory:Optional

CURRICULUM
Number of attributes: 4
Key attributes: Curriculum-Number
Foreign keys: None
Relationships: CURRICULUM to MEMBER; 1:N; Mandatory:Optional

B. APPLICATION DESIGN

The application is the interface between the user and the database. It contains various control mechanisms to prevent direct access to the database and to maintain the integrity of the database. A menu hierarchy was used to aid and control user interaction with the system. The menu-driven approach was employed because it enables inexperienced end-users to access and use the full functionality of a system faster than with a command-driven system. The menu hierarchy depicted in Figure 4 was derived from user requirements. The Append, Edit/View, and Delete/View sub-menus apply to a selected object database. All user-selectable operations flowed from Main Menu selections. Figure 5 shows the final look of the Main Menu and depicts the generic structure of all menus. Figure 6 provides a view of the form for editing/viewing an existing member record. Although specific fields differ across the various forms in the application, the same form "template" is used throughout the application. Appendix C, the User's Manual, presents comprehensive graphics of application menus, reports, forms, recall letters, and screens.

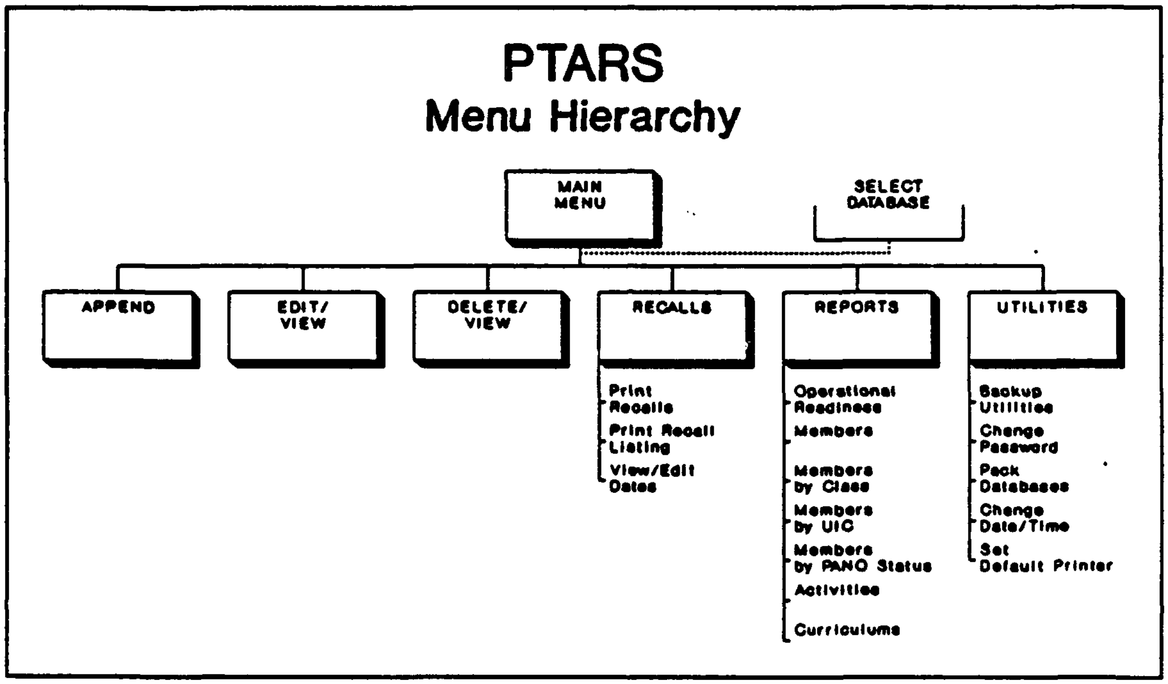
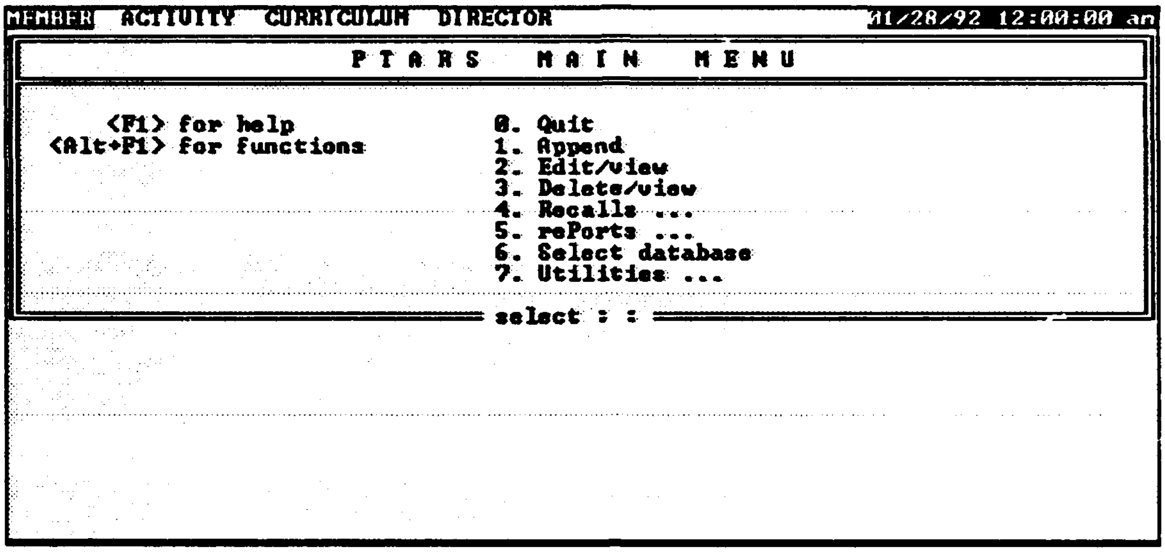
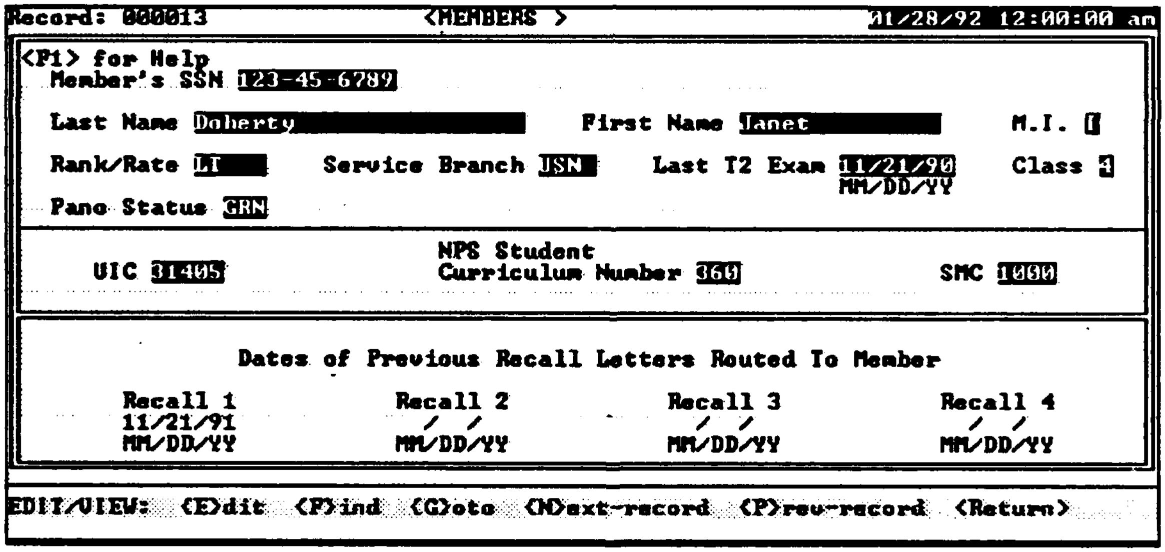

PTARS
Menu Hierarchy
MAIN MENU
SELECT DATABASE
APPEND
EDIT/ VIEW
DELETE/ VIEW
RECALLS
REPORTS
UTILITIES
Print Recalls
Print Recall Listing
View/Edit Dates
Operational Readiness
Members
Members by Class
Members by UIC
Members by PANO Status
Activities
Curriculums
Backup Utilities
Change Password
Pack Databases
Change Date/Time
Set Default Printer

Figure 4. Menu Hierarchy



```
MEMBER  ACTIVITY  CURRICULUM  DIRECTOR                     01/28/92 12:00:00 am
                         P T A R S   M A I N   M E N U

      <F1> for help                  0. Quit
   <Alt+F1> for functions            1. Append
                                     2. Edit/view
                                     3. Delete/view
                                     4. Recalls ...
                                     5. rePorts ...
                                     6. Select database
                                     7. Utilities ...
                                  select : :
```

Figure 5. Main Menu Screen



```
Record: 000013                 <MEMBERS >                  01/28/92 12:00:00 am
<F1> for Help
  Member's SSN 123-45-6789

  Last Name Doherty                First Name Janet              M.I.

  Rank/Rate LT     Service Branch USN    Last T2 Exam 11/21/90   Class 4
                                                      MM/DD/YY
  Pano Status GRN

                               NPS Student
     UIC 31405                 Curriculum Number 360          SMC 1000

               Dates of Previous Recall Letters Routed To Member

     Recall 1            Recall 2            Recall 3            Recall 4
     11/21/91             /  /                /  /                /  /
     MM/DD/YY            MM/DD/YY            MM/DD/YY            MM/DD/YY

EDIT/VIEW: <E>dit  <F>ind  <G>oto  <N>ext-record  <P>rev-record  <Return>
```

Figure 6. MEMBER Edit/View Form

After establishing the menu hierarchy and obtaining user approval of report, form, recall letter, and screen prototypes, an integrated prototype of the application was developed. That is, a working model of the system was created but with incomplete functionality [Ref. 3, 4]. End-user evaluations of the prototype's characteristics and operation were used to iteratively revise the model. This prototype was then expanded in functionality to become the final system. This approach was facilitated by Foxpro's project management capability for unifying and coordinating the separate elements of the application. Added advantage was obtained from the use of this approach in that end-users became intimately involved in the development process and actively influenced the look and functioning of the final system. Thus, by the time of implementation their expectations were satisfied and they were well-versed in the application's functioning.

Care was taken to establish consistency of function across modules with regard to form and menu design, messages, escape procedures, navigation keys, function-key use, and availability of on-line help. Moreover, as indicated in the object specifications (Appendix [illegible]), the range and format of data for most of the fields was carefully controlled.

IV. SYSTEM IMPLEMENTATION

System implementation was the final step of the development process. The primary objective was to build the fully functional physical application that satisfied the end-user. The physical database was constructed using a DBMS-specific methodology, Foxpro 2.0. It is compatible with the widely-used dBase DBMS language and has numerous language extensions. Moreover, as noted previously, the product provides a very efficient, windowed development environment that facilitates coding, compiling, running, and debugging from within an integrated interface.

During implementation, the prototype was expanded to include all modules fully integrated into an application with complete functionality. Appendix C, the User's Manual, provides documentation which details the final application's features and operations. Documented program code, procedure and token listings, and a token cross-reference listing are included in Appendix E.

Installation required converting the mainframe database and adding several data elements. Hence, the installation and transition to the new system took several days to complete. Primary user training was accomplished during the development process.

V. SUMMARY AND RECOMMENDATIONS

A. SUMMARY

The mainframe-based patient tracking and recall system was due for replacement. It was out-dated in its user interface, was unreliable to access, lacked adequate field validation checks, and required additional capabilities. The PTARS system designed and implemented during the course of this thesis addressed all of these deficiencies and included users actively in the development process. The system is user-friendly and includes all necessary functions internally to provide security, data integrity, and an intuitive operation.

B. RECOMMENDATIONS

During the development process much thought was given to anticipating the needs of end-users. On-line, context-sensitive help was provided for all operations and fields; and confirmations, messages, and prompts were built into all operations that affected the content of the database. Nevertheless, it is still incumbent upon the user to make choices and take actions to protect the data and maintain the quality of unrestricted character fields.

Data security will be only as good as the user's attention to it. The password must be protected, the system must not be left running unattended, and regular backups to floppy disk must be made and stored to safety. All of these activities are ultimately left up to the discretion of the user. Proper training and careful reading of the User's Manual should enhance end-user adherence to recommended practice.

Finally, NRDC currently provides PC hardware and software support to branch clinics. Upon request, a PC technical expert will troubleshoot problems with BDCM computer resources. The necessity of PCs in the branch clinics is acknowledged and some standard software is provided for an integrated dental information system. Yet, clinics are not provided the resources to protect their systems. For example, no user training is conducted regarding routine machine or data maintenance or security. This could develop into a significant problem in the event of a large data loss. NRDC should consider providing all branch clinics with reasonably efficient backup software, disk maintenance and data recovery software utilities, and the training to use them effectively.

LIST OF REFERENCES

1. *Foxpro Developer's Guide*, Fox Software, Perrysburg, Ohio, 1991.

2. Kroenke, D. M. and Dolan, K.A., *Database Processing* (Third Edition), Science Research Associates, Inc., Chicago, Illinois, 1988.

3. Boar, B., *Application Prototyping: A Requirements Definition Strategy for the 80's*, John Wiley & Sons, New York, New York, 1984.

4. Senn, J. A., *Analysis and Design of Information Systems* (Second Edition), McGraw-Hill, New York, New York, 1989.

APPENDIX A: OBJECT SPECIFICATIONS

Object Definitions

MEMBER OBJECT

Descriptive name	Domain name
Social Security Number	SSN
Last Name	Last_name
First Name	First_name
Middle Initial	MI
Rank or Rate	Rank_rate
Service Branch	Branch
Last T2 Exam	Last_T2
Class Rating	Class
Pano X-ray Status	Pano
SMC or Department Code	SMC
Recall Letter 1 Date	Recall_1
Recall Letter 2 Date	Recall_2
Recall Letter 3 Date	Recall_3
Recall Letter 4 Date	Recall_4
ACTIVITY; ACTIVITY object	
CURRICULUM; CURRICULUM object	

ACTIVITY OBJECT

Descriptive name	Domain name
Unit Identification Code	UIC
Unit Acronym	Acronym
Activity Name	Act_name
Point-of-Contact	POC
MEMBER; MEMBER object; MV	

CURRICULUM OBJECT

Descriptive name	Domain name
Curriculum Number	Curr_num
Curriculum Name	Curr_name
Department Code	Dept_code
Phone Number	Phone_no
MEMBER; MEMBER object; MV	

Domain Definitions

Acronym:
Character (11)
Abbreviated activity name

Act_name:
Character (47)
Official abbreviated name of an NPS tenant command served by BDCM

Branch:
Character (4)
Abbreviation for member's service branch

Class:
Numeric (1), range 1-4
Class rating assigned by dentist to each member

Curr_name:
Character (46)
NPS curriculum name

Curr_num:
Character (3)
Unique NPS curriculum number code

Dept_code:
Character (2)
Curriculum office NPS department code

First_name:
Character (15)
Member's first name

Last_name:
Character (23)
Member's last name

Last_T2:
Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year
Last T2 exam date

MI:
Character (1)
Member's middle initial

Pano:
Character (3)
Member's pano x-ray status

Rank_rate:
Character (5)
Member's rank or rate

Recall_1:
Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year
Recall letter 1 date

Recall_2:
Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year
Recall letter 2 date

Recall_3:
Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year
Recall letter 3 date

Recall_4:
Date (8); Mask MM/DD/YY, where MM is month, DD is day, YY is year
Recall letter 4 date

SMC:
Character (4)
Member's student mail center number or staff department mail code

SSN:
Character (11); Mask NNN-NN-NNNN, where N are any digits
Unique member Social Security Number

UIC:
Character (6)
Unique Unit Identification Code of NPS tenant command

APPENDIX B: UPDATE, DISPLAY, AND CONTROL MECHANISMS

I. Update Mechanisms
   A. Append/Edit MEMBER data
      1. Inputs
         - Initial member data received at physical check-in of member records to BDCM
         - Member change data received on roster from PSD
         - Member change data received on roster from Registrar
         - MEMBER object instance from database
         - ACTIVITY object instance from database
         - CURRICULUM object instance from database
         - System-date and time
      2. Outputs
         - New or modified MEMBER object instance in database
         - Confirmation message on screen
      3. Processing notes
         - This function used for both new and current members
         - All initial member data manually entered after review of member's dental record
         - Student SMC number may not be available initially
      4. Volume
         - 225 Jun; 75 Feb/Jul; 250 Mar/Sep/Dec
         - Seven per week on average after quarter start
         - 275 edits per week on average
      5. Frequency
         - Six times per year for large batch; otherwise daily
   B. Delete MEMBER data
      1. Inputs
         - Member takes physical custody of dental records upon detachment
         - MEMBER objects in database
      2. Outputs
         - Confirmation notice on screen
      3. Processing notes
         - Backups of MEMBER data should be made prior to processing a batch of deletions
      4. Volume
         - 250 at end of each academic quarter
         - Seven per week on average after quarter end
      5. Frequency
         - Four times per year for large batch; otherwise daily
   C. Append/Edit ACTIVITY data
      1. Inputs
         - Activity data change from Personnel Support Detachment (PSD)
         - ACTIVITY object instance from database
      2. Outputs
         - New or modified ACTIVITY object instance in database
         - Confirmation message on screen
      3. Processing notes

    - This function will be seldom used since it will be triggered by the addition or modification of a generally stable client organization
    - This function used for both new and current activities
4. Volume
    - Variable, approximately one instance every two years on the average
5. Frequency
    - Variable, approximately once every two years

D. Delete ACTIVITY data
1. Inputs
    - Activity data change from Personnel Support Detachment (PSD)
    - ACTIVITY object instance from database
2. Outputs
    - Confirmation notice on screen
3. Processing notes
    - This function will be seldom used since it will be triggered by the elimination of a generally stable client organization
    - Backup of ACTIVITY data should be made prior to deletion
4. Volume
    - Variable, approximately one instance every four years on the average
5. Frequency
    - Variable, approximately once every four years

E. Append/Edit CURRICULUM data
1. Inputs
    - Curriculum data change from Registrar
    - CURRICULUM object instance from database
2. Outputs
    - New or modified CURRICULUM object instance
    - Confirmation message on screen
3. Processing notes
    - This function will be seldom used since it will be triggered by the addition or modification of generally stable curriculums
    - This function used for both new and current curriculums
4. Volume
    - Variable, approximately two instances per year on the average
5. Frequency
    - Variable, approximately twice per year prior to new student class

F. Delete CURRICULUM data
1. Inputs
    - Curriculum data change from Registrar
    - CURRICULUM object instance from database
2. Outputs
    - Confirmation message on screen
3. Processing notes
    - This function will be seldom used since it will be triggered by the elimination of a generally stable curriculum
    - Backup of curriculum data should be made prior to deletion
4. Volume
    - Variable, approximately one instance every five years on the average
5. Frequency
    - Variable, approximately once every five years

II. Display Mechanisms
  A. Query on MEMBER
    1. Output description
      - Form showing all data for a member to screen
    2. Source data
      - MEMBER object
      - Member SSN or name keyed by user
    3. Processing notes
      - Used by Administrative Petty Officer or Receptionist
    4. Volume
      - Five per week
    5. Frequency
      - Daily
  B. Recall letter 1
    1. Output description
      - Memorandum mailed to member
      - New or modified MEMBER object instance in database
    2. Source data
      - MEMBER object
      - System-date
    3. Processing notes
      - This process is initiated from a menu by the user. It creates recall letter one for all members whose last T2 exam was more than 10 months prior to the system-date and for whom recall letter one was not previously produced
      - This process inserts system-date as Recall-Ltr1-Date when conditions above exist
    4. Volume
      - 160 monthly
    5. Frequency
      - End of every month
  C. Recall letter 2
    1. Output description
      - Memorandum mailed to member
      - New or modified MEMBER object instance in database
    2. Source data
      - MEMBER object
      - System-date
    3. Processing notes
      - This process is initiated from a menu by the user. It creates recall letter two for all members whose last T2 exam was more than 11 months prior to the system-date, for whom recall letter one was produced, and for whom recall letter two was not previously produced
      - This process inserts system-date as Recall-Ltr2-Date when conditions above exist
    4. Volume
      - 100 monthly
    5. Frequency
      - End of every month
  D. Recall letter 3
    1. Output description
      - Letter mailed to member
      - New or modified MEMBER object instance in database

2. Source data
    - MEMBER object
    - System-date
3. Processing notes
    - This process is initiated from a menu by the user. It produces recall letter three for all members whose last T2 exam was more than 12 months prior to the system-date, for whom recall letter two was produced, and for whom recall letter three was not previously produced
    - This process inserts system-date as Recall-Ltr2-Date when conditions above exist
4. Volume
    - 30 monthly
5. Frequency
    - End of every month

E. Recall letter 4
1. Output description
    - Letter mailed to Curriculum Officer for student members and Activity POC for all other members
    - New or modified MEMBER object instance in database
2. Source data
    - MEMBER object
    - ACTIVITY object
    - CURRICULUM object
    - System-date
3. Processing notes
    - This process is initiated from a menu by the user. It produces recall letter four for all members whose last T2 exam was more than 13 months prior to the system-date, for whom recall letter three was produced, and for whom recall letter four was not previously produced
    - This process inserts system-date as Recall-Ltr4-Date when conditions above exist
    - Student members uniquely belong to UIC 31405
4. Volume
    - 3 monthly
5. Frequency
    - End of every month

F. Operational Readiness Report
1. Output description
    - Screen display of summary count and percent of patient load for all members by class
    - Screen display of summary count and percent of all patients in Pano x-ray status categories
2. Source data
    - MEMBER object
    - System-date
3. Processing notes
    - This process is initiated from a menu by the user. It creates a summary report of the number and percentage of all members in each of the four different dental classes. The report can be optionally printed.
4. Volume
    - 1 monthly
5. Frequency
    - End of every month

G. Member Roster
  1. Output description
     - Printed roster of all members sorted alphabetically or by SSN
  2. Source data
     - MEMBER object
     - System-date
  3. Processing notes
     - This process is initiated from a menu by the user.
  4. Volume
     - 1 monthly
  5. Frequency
     - End of every month

H. Member Roster by Class
  1. Output description
     - Printed roster of members sorted alphabetically or by SSN; available for all or for specified class
  2. Source data
     - MEMBER object
     - System-date
  3. Processing notes
     - This process is initiated from a menu by the user.
  4. Volume
     - 1 monthly
  5. Frequency
     - End of every month

I. Member Roster by UIC
  1. Output description
     - Printed roster of all members sorted alphabetically or by SSN
  2. Source data
     - MEMBER object
     - System-date
  3. Processing notes
     - This process is initiated from a menu by the user.
  4. Volume
     - 1 monthly
  5. Frequency
     - End of every month

J. Member Roster by Pano X-ray status
  1. Output description
     - Printed roster of members sorted alphabetically or by SSN; available for all members or for specified Pano status
  2. Source data
     - MEMBER object
     - System-date
  3. Processing notes
     - This process is initiated from a menu by the user.
  4. Volume
     - 1 monthly
  5. Frequency

        - End of every month
    - K. Activities Listing
        1. Output description
            - Printed roster of Activities sorted by UIC
        2. Source data
            - ACTIVITY object
            - System-date
        3. Processing notes
            - This process is initiated from a menu by the user.
        4. Volume
            - 1 monthly
        5. Frequency
            - End of every month
    - L. Curriculums Listing
        1. Output description
            - Printed roster of Curriculums sorted by curriculum number
        2. Source data
            - CURRICULUM object
            - System-date
        3. Processing notes
            - This process is initiated from a menu by the user.
        4. Volume
            - 1 monthly
        5. Frequency
            - End of every month
- III. Control Mechanisms
    - A. Access to the system is protected by a password known only by the Administrative Petty Officer and the Receptionist
    - B. The system is limited to use by one person at a time.
    - C. Monthly validations of various member data are accomplished using rosters obtained from PSD and the Registrar

APPENDIX C: USER'S MANUAL

NPS DENTAL CLINIC PATIENT TRACKING & RECALL SYSTEM

Timothy P. Steele
LCDR, MSC, USN
Naval Postgraduate School

CONTENTS

Introduction 1

Features overview 1
Typographical conventions 2

Chapter 1 Getting started 3

System requirements 3
Installation 4
  Installation overview 4
  Installing PTARS 4
  Re-installing PTARS 5
Starting PTARS 6
  Updating member CLASS 6
  Security 6
  Creating a start-up batch file 7

Chapter 2 Getting around 9

Navigation/Activation keys 9
Function keys 9
Using on-line Help 10
Menus overview 11
Main Menu 12
  Selecting a database 12
  Exiting PTARS 13

Chapter 3 Database updating 15

Appending records 15
Editing/viewing records 16
Deleting/viewing records 19

Chapter 4 Recalls 21

Printing recalls 22
Printing recalls lists 27
Viewing/editing recall dates 27

Chapter 5 Reports 29

Operational readiness 29
Rosters 30

Chapter 6 Utilities 33

Backup utilities 33
    Backing up database(s) 34
    Listing files 35
    Formatting a floppy disk 35
    Restoring backup(s) 35
Changing the password 36
Packing the database(s) 37
Changing the date or time 37
Selecting the default printer 37

Appendix A Optimizing PTARS 39

Disk defrag/compress 39
Memory 39
    Conventional memory 39
    Expanded memory 40
    Extended memory 40
Config.sys 41
Pack the database(s) 41

Appendix B File definitions 43

Appendix C Database specifications 45

Introduction

Welcome to the Naval Postgraduate School Dental Clinic (NPSDC) Patient Tracking and Recall System (PTARS). This database application was developed to provide an in-house, PC-based capability for NPSDC to maintain the patient data necessary to track and recall patients for annual exams and to produce operational readiness statistics. The system provides fast, dependable access to member records and automates the recall process.

PTARS was designed based on extensive interviews with the NPSDC staff to identify clinic requirements. Prototypes of the system were iteratively developed and demonstrated to ensure that clinic end-users were fully satisfied with the final system specifications. A primary design objective was to develop an application that was very user-friendly. Hence, you will be able to use the system productively with only a minimum amount of familiarization time. Please take a few minutes now to review this User's Manual.

Features overview

PTARS employs four database files that are directly accessible to user modification: MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and DIRECTOR.DBF. MEMBERS.DBF contains the information pertinent to each patient. The files ACTIVITY.DBF and CURRICUL.DBF are used for locating patients and for printing recall letter addresses. ACTIVITY.DBF contains information specific to each UIC served by NPSDC and CURRICUL.DBF contains information specific to each NPS student curriculum. DIRECTOR.DBF contains the name of the current NPSDC Director for placement into the signature line of recall letters.

The application provides a series of simple menus and sub-menus from which to choose its various options. You will be able to view, append, update, and delete Member, Activity, Curriculum, and Director data using screen forms with built-in error-checking routines for each action or data entry. You will also be able to print special reports, sorted database listings, and recall letters. Additional features include but are not limited to:

- Password controlled access to PTARS; changeable password
- Automatic updating of member treatment class status

- Context-sensitive help
- System information display
- Continuous date and time display
- Automatic determination of appropriate recall letters to print
- Backup database(s) to hard disk or floppy disk; restore backup(s)
- Format floppy disk from within application
- List files on hard disk or floppy disk
- Automatic reminders for database backup (if more than one month since last backup) and database pack (if more than 10% of records marked for deletion)

Typographical conventions

The following typographical conventions are used in this manual:

Input	Anything that you type is in the Courier typeface, for example, `a:\setup <Enter>`

Keys	Keys to be pressed are represented like this:
	`<Esc>` `<Enter>` `<F1>` `{C}`
	Press both keys simultaneously when a "+" symbol is present, as in:
	`<Alt+F1>`

Direction	Cursor movement keys are indicated as: `<PgUp>` `<PgDn>` `<Arrows>`

1

Getting started

This chapter contains all the information you need to install and run PTARS. It also discusses the various settings that you can change.

It contains the following sections:

- System requirements
- Installation
- Starting PTARS
- Creating a batch file

System requirements

PTARS requires the following hardware and software:

- An IBM compatible computer with at least 512K of random access memory (RAM) (640K of RAM strongly recommended)
- One floppy disk and one hard disk drive (with at least 3 megabytes of space available)
- Version 2.0 or later of DOS
- A CONFIG.SYS file in your root directory with a Files=25 (or greater) statement
- An EGA or VGA video adapter
- An Epson E/F/J/RX/LQ compatible or IBM Proprinter compatible dot-matrix printer

Additional requirements:

- To take advantage of Expanded memory support, you need an expanded memory card that is hardware and software compatible with the Lotus-Intel-Microsoft standard 4.0 or later (LIM 4.0 EMS). If you have an Intel 80386 or 80486 processor you can also use extended memory and a software expanded memory emulator program. PTARS can use 64K of expanded memory as additional general purpose memory and any remaining expanded memory to speed up file I/O.

- If expanded memory is not available but the computer has extended memory, PTARS can be configured during installation to use 512K of the available extended memory for a disk cache to speed up file I/O.

- Double-copy paper to automatically make copies of recall letters. Since a copy of Recall 3 is identified as an enclosure to Recall 4, a copy of Recall 3 should be available before routing Recall 4. An alternative to double-copy paper would be making a copy of all Recall 3 letters after printing; then filing them in the event a Recall 4 was necessary for the same individual(s).

Installation

Installation overview

You have been provided with four numbered floppy disks. Disks 1 to 3 contain the files necessary to install and run PTARS. Disk 4 contains the initial database files that were current at the time of program delivery (i.e., MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and DIRECTOR.DBF). There are two steps to installing PTARS:

- **Make a backup and install the program.** Before you do anything else, copy the original disks and store them in a safe place. Then, use your copies of the original disks and run the Setup program to install PTARS on your hard disk.
- **Choose the default printer.** Before you print for the first time, you should select the default printer emulation from the Utilities Menu.

Installing PTARS

Refer to your computer's documentation (or ask your local computer guru) to determine whether your computer has expanded memory, disk caching hardware or software, and/or extended memory. You will be queried during the installation process regarding your computer's configuration. Note that you need at least 3 megabytes of available hard disk space before you begin.

One cautionary note before beginning your installation. PTARS was designed to run using only one computer at a time. Although in the future it may be tempting to install PTARS on a second computer, **avoid installing PTARS on more than one computer.** Because the separate installations can not communicate, there is no built-in, guaranteed way for the separate databases to maintain the same up-to-date data. Although you could theoretically transfer data using floppy disks, almost assuredly over time some data would exist in one machine but not the other, and vice-versa.

The steps for installing PTARS are as follows:

1. Insert the PTARS disk #1 in drive A.

2. At the DOS prompt, type `a:\setup`. The Setup program will start.

3. When prompted by Setup, specify the disk where you want to install PTARS (e.g., `c`). Setup creates the subdirectory "\PTARS" on the hard disk specified and copies the program files and initial database files to it. Setup prompts you to insert each disk when necessary.

4. After copying, assembling, and un-compressing all the files from the installation disks, Setup queries whether your computer has expanded memory and/or a disk cache. Respond `y` or `n`, as appropriate. If you respond negatively, Setup queries whether you have extended memory. Again, respond as appropriate. This process determines how PTARS is configured for start-up.

5. When the installation is complete, Setup presents a screen with installation notes. Read the notes. Setup then queries whether you want to start PTARS. If you respond affirmatively, PTARS loads immediately.

6. Before printing from PTARS for the first time, select the default printer from the Utilities Menu. Refer to your printer's documentation to determine which emulation (Epson E/F/J/RX/LQ or IBM Proprinter) your printer uses. The default printer emulation is initially Epson.

7. Align the paper in your printer. Test the margin adjustments of your paper by printing the Operational Readiness Report from the Reports Menu. The top of your paper should be set in your printer so that one blank line exists at the top of the printed report. Likewise, the paper should be set so that one blank space exists to the left of the header statement "FOR OFFICIAL USE ONLY". If your paper is adjusted in the printer to satisfy these conditions, all printing from PTARS will be formatted properly.

Re-installing PTARS

There are two instances when you may want to re-install PTARS: 1) when there is some problem with any of the program files or 2) the computer has been modified with regard to expanded memory, a disk cache, or extended memory.

The re-install process is exactly the same as the initial installation with two exceptions. Setup attempts to determine if PTARS has been installed previously. If Setup detects that this is a re-installation you will be presented with a listing of existing database files in the "\PTARS" subdirectory and a re-installation note on screen. You can elect to continue or quit the re-installation at this time. If you elect to continue, you will be queried regarding which, if any, of the initial database files you may want to re-install. Note that if you have been using PTARS for any period of time you will probably elect not to re-install any of the initial database files. This is because they will be out of date. Use the "Restore backup(s)" option in the Backup Utilities Menu to restore your most recent data from floppy disk, if necessary.

Starting PTARS

If necessary, change to the "\PTARS" subdirectory on the drive where you installed PTARS (e.g., at the DOS prompt, type `cd\ptars`). Then type `ptars` and press `<Enter>`. A logo screen will appear and pause briefly. (You can eliminate the pause by pressing any key during the logo display.) Following the pause, the PTARS Access Screen appears and you are requested to enter the password. The initial password to use is "`zyxabc`". You will be given up to three attempts to enter the correct password. After a third failure, PTARS shuts down.

After correctly entering the password, you will be queried whether the system date and time are correct. If you respond negatively, you are prompted to enter the correct date and/or time according to the format displayed.

Updating member CLASS

When the system date and time are correct, PTARS updates each member's dental CLASS rating. CLASS ratings of "1", "2", or "3" are assigned to members by an examining dentist. A CLASS rating of "4" indicates simply that the member is due for his/her mandatory annual dental examination. PTARS scans each record in the MEMBERS.DBF database file and checks to see if the LAST_T2 date is more than 12 months prior to the current system date. If so, it replaces the existing CLASS rating with "4". After updating member CLASS, PTARS displays the Main Menu.

Security

It is *strongly* recommended that the default password be changed after installing the PTARS program. Your data is extremely important. Inadvertent or deliberate tampering with your data by an unauthorized person can only be prevented by taking security precautions (*and* taking them seriously). In addition to keeping a secure password, it is very important that you do not leave PTARS running unattended. The temptation to do so, however, will be great. Making regular backups of your data to floppy disk and putting them in a safe place is probably the best way to ensure against loss of data due to any cause.

Creating a start-up batch file

A DOS batch file can be created that will enable you to start PTARS at any time regardless of what directory you may currently be in, without having to type additional DOS commands. Use a text editor (or a word processor mode that does not insert hidden formatting codes) to create a batch file like the example below. The example batch file assumes that you have installed PTARS to the C drive.

```
C:
CD\PTARS
PTARS
```

When the batch file is complete, name it "PTARS.BAT" and place it in your root directory or any directory that is in your DOS path. Henceforth, simply type `PTARS` to load the PTARS program from any location. See any DOS reference for terminology assistance.

2

Getting around

This chapter contains the information you need to navigate the menus, forms, and fields of PTARS. It covers:

- Navigation/Action keys
- Function keys
- Using on-line Help
- Menu overview
- Main menu
- Exiting PTARS

Navigation/Action keys

Each PTARS screen shows the available commands or options. The following keys let you move around a screen, between or within fields, or perform various generic actions:

Press:	To:
`<Arrows>`	move up or down one line; move left or right one character or screen
`<PgUp>/<PgDn>`	display previous or next screen of a multiple records screen
`<Home>`	move to the start of a multiple records screen or input field
`<End>`	move to the end of a multiple records screen or input field
`<Backspace>`	delete character to left; move back one input field
`<Return>`	accept an entry; move to next field
`<Insert>`	toggle insert/typeover mode
`<Del>`	delete a character or record
`<Esc>`	cancel the current task

Function keys

Function keys `<F1>` through `<F4>` are assigned specific actions as described below. Pressing `<Alt+F1>` (pressing both keys simultaneously) at any time presents a popup reminder list of the functions available. Functions are activated by pressing the assigned function key or selecting the function from the popup list. Functions are available at all times, regardless of the current activity. The functions available are:

Help <F1>	Context-sensitive help window. See the next section, "Using on-line Help".
Calendar <F2>	Pops-up a monthly Calendar display. It shows the current month in row and column form with the current day highlighted. You can move forward or backward in months by pressing <PgUp> or <PgDn>, and in years by pressing <Ctrl-PgUp> or <Ctrl-PgDn>, respectively. To get back to the current date, press {T}. As with almost all operations in PTARS, press <Esc> to exit.
Poptris <F3>	A Tetris-like diversion. The object is to fill the rectangular field with the falling objects from the bottom up without leaving any open spaces. Use the numeric keypad arrows to position falling objects within the field. Pressing the number 5 key causes the shape of the falling object to change. It can be pressed repeatedly to cycle the shape of the falling object. Pressing the ↓ arrow key causes the falling object to land immediately, hence, speeding up the activity. Additional commands/functions are displayed on-screen. Poptris code has been included by permission of Gerald F. Garcia.
About PTARS <F4>	A window containing system environment information. It includes information on the operating system, computer hardware, RAM, and disk space.

Using on-line Help

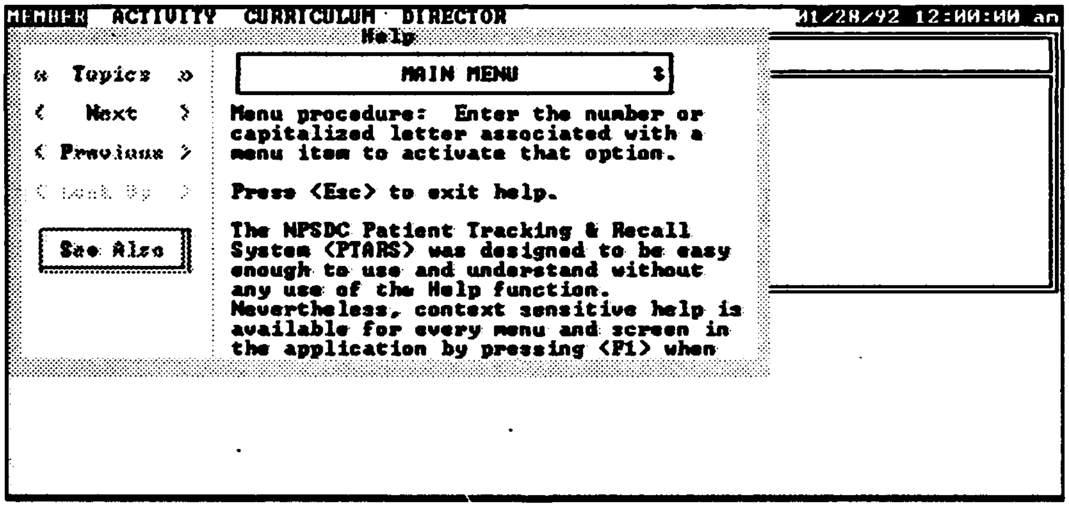
On-line Help is available at all times by pressing <F1>. Help is "context-sensitive" since the Help Topic details initially displayed apply to the current PTARS screen. When the ↕ symbol is present in the topic box, you can scroll down or up through the Help window to view additional text using the ↓ or ↑ arrow keys.

As shown in Figure 1, the Help window consists of two panels—one lists Help Topics and the other displays details about each Topic. At the bottom of the Topics list all fields in the various databases are identified with a " ~ " prefix and are defined. Commands available in Help are described below:

« Topics »	This provides a list of Topics available in the Help system. To select a Topic you can: 1) use the arrow keys to scroll through the Topics

to find the one you want or 2) type a letter or series of letters to select the first Topic beginning with those letter(s). To see details about a Topic, select the Topic and press `<Return>`.

`<Next>`	This selects Help details for the next Topic in the help file list.
`<Previous>`	This selects Help details for the prior Topic in the help file list.
`<Look Up>`	Enables you to find the closest Topic match to a word that you highlight within Help details. When you highlight a word in the Help text, the `<Look Up>` function becomes available. You highlight a word by placing the cursor at the first letter in a word using the ← and → arrow keys. Then press `<Shift+→>` to highlight the word.
`See Also`	This lists Help Topics that may be of interest related to the current Topic.
`<Esc>`	Exits Help.



Menu procedure: Enter the number or capitalized letter associated with a menu item to activate that option.

Press <Esc> to exit help.

The NPSDC Patient Tracking & Recall System (PTARS) was designed to be easy enough to use and understand without any use of the Help function. Nevertheless, context sensitive help is available for every menu and screen in the application by pressing <F1> when

Figure 1. Help window appearing over Main Menu.

Menus overview

PTARS is a "menu-driven" system. All operations are activated by selecting options from full-screen menus, from sub-menus located at the bottom of the screen, or from pop-up menus. An option can be selected on all menus by pressing the highlighted (and capitalized) letter associated with the option. On full-screen menus the number of the menu option will also activate the option. On popup menus you can also scroll to the desired option and press `<Enter>` to activate the option. Figure 2 below provides a graphical view of the major menu operations within PTARS.

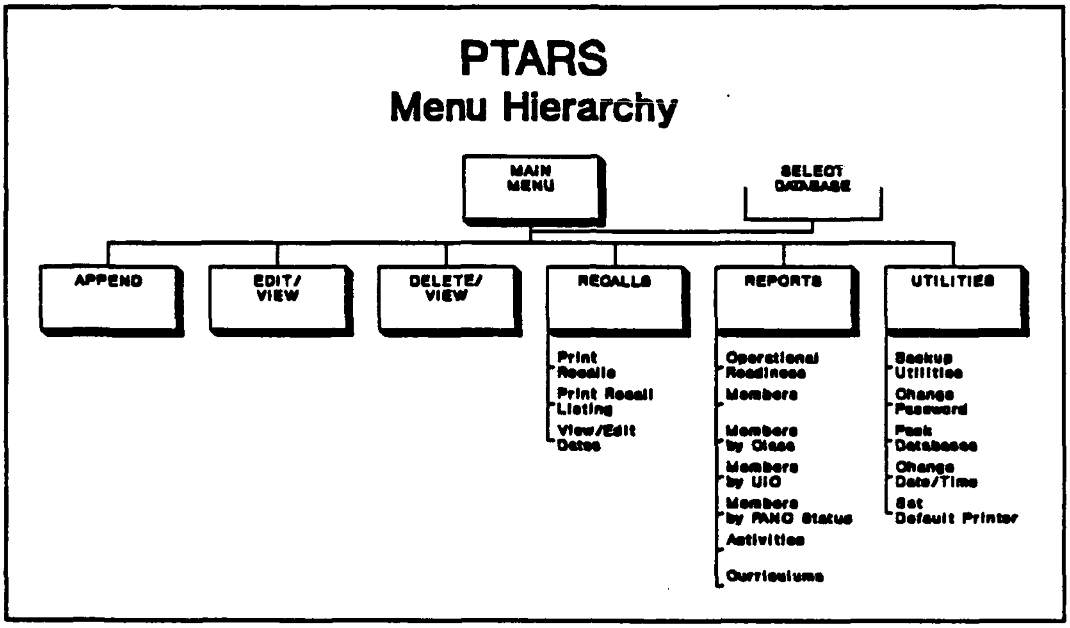

PTARS
Menu Hierarchy
MAIN MENU
SELECT DATABASE
APPEND
EDIT/ VIEW
DELETE/ VIEW
RECALLS
REPORTS
UTILITIES
Print Recalls
Print Recall Listing
View/Edit Dates
Operational Readiness
Members
Members by Class
Members by UIC
Members by PANO Status
Activities
Curriculums
Backup Utilities
Change Password
Pack Databases
Change Date/Time
Set Default Printer

Figure 2. PTARS menu hierarchy.

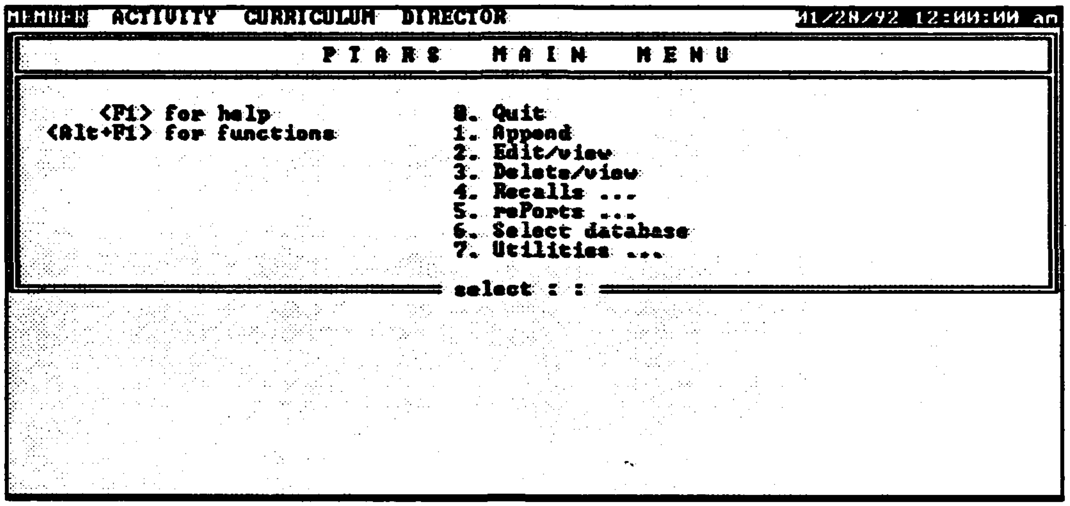
Main Menu

After updating member CLASS, PTARS displays the Main Menu, as shown in Figure 3 on the next page. Each screen in PTARS continuously displays the system date and time in the upper right corner.

Selecting a database

In the upper left corner of the Main Menu the four databases of interest are identified. The active database is highlighted and blinking. By default, Members is the initially active database. The Main Menu options "Append", "Edit/view", and "Delete/view" apply only to the active database. A different database can be made active by choosing the option, "Select database", and then selecting the desired database from the popup selection list.

Exiting PTARS is discussed in the following sub-section. The remaining Main Menu options are covered in detail in subsequent chapters.



```
MEMBER  ACTIVITY  CURRICULUM  DIRECTOR                       01/28/92 12:00:00 am
                          P T A R S   M A I N   M E N U

       <F1> for help                  0. Quit
   <Alt+F1> for functions             1. Append
                                      2. Edit/view
                                      3. Delete/view
                                      4. Recalls ...
                                      5. rePorts ...
                                      6. Select database
                                      7. Utilities ...
                                   select : :
```

Figure 3. PTARS Main Menu.

Exiting PTARS

It is very important that you exit (quit) PTARS using the Main Menu "Quit" option. If you reboot the computer with `<Ctrl+Alt+Del>` or shut the power off without first quitting properly, any databases which are in use at the time are vulnerable to damage. Hence, it is essential that you exit only by using the Main Menu "Quit" option.

When quitting, several things happen before the system shuts down. First, PTARS checks to see if it has been more than one month since MEMBERS.DBF has been backed-up to a floppy disk. If so, a reminder message pops-up on screen and you are given the option to perform a backup. If you choose to perform a backup, PTARS switches to the Backup Utilities Menu where you can perform your backup operations and quit when you are finished.

Next, PTARS checks to see if more than 10% of the records in MEMBERS.DBF have been marked for deletion. If so, a message pops-up and you are queried whether you want to "pack" the database. See Chapter 6 for details on packing the database.

Finally, before shutting down, PTARS queries whether you want to back-up the databases to the hard disk. This allows you to save a second copy of your session's work on the hard disk. See Chapter 6 for further coverage of backing-up.

3

Database updating

This chapter contains the information necessary for updating the databases by appending, editing, or deleting records. Several example screens will be shown to preview the look of PTARS when working with its various modes.

Appending Records

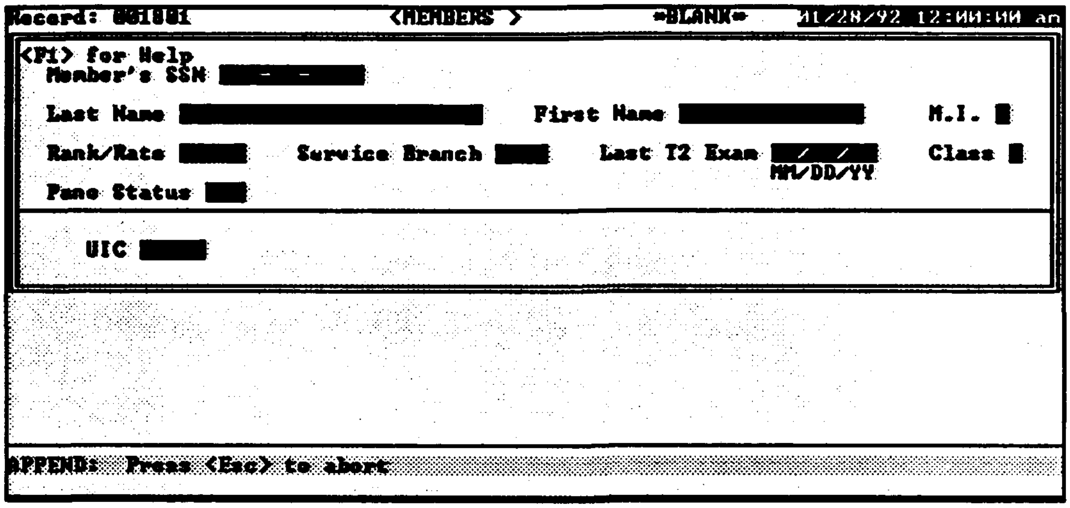
Select the "Append" option from the Main Menu to append records. Appending records involves adding new records to a database. New records can be appended to MEMBERS.DBF, ACTIVITY.DBF, and CURRICUL.DBF. Unlike the foregoing three databases, DIRECTOR.DBF contains only one record. This record contains the name of the current clinic director and must always be present. Hence, it can only be edited.

As discussed in Chapter 2, PTARS starts by default with MEMBERS.DBF as the active database. You can select a different database from the Main Menu option "Select Database". To append records, press `{A}` from the Main Menu. A blank form will appear, ready to receive new data. You can abort from appending by pressing `<Esc>` and the record will not be saved.

When appending a record almost all fields require an entry. If a field is left blank and `<Enter>` is pressed, either a warning will appear stating that an entry is required or a popup list of valid field entries will appear. When a popup list appears, scroll to the desired field entry and press `<Enter>` to insert the entry into the form. Figure 4 shows the Append data entry form for Members.

If the member is an NPS student (i.e., UIC = "31405"), a field for Curriculum Number and SMC (Student Mail Center number) will appear following UIC. Alternatively, if the member is a non-student, a field for Activity Department Code will appear. Enter data into these fields as appropriate.

As a reminder, if you have any doubts regarding the contents of a certain field, be sure to utilize the Help function. Each field in all the databases is described in the Topics section of Help. Field names are prefixed with the " ~ " symbol and are located at the bottom of the scrollable Help Topics list.



```
Record: 001001            <MEMBERS >            *BLANK*     01/28/92 12:00:00 am
<F1> for Help
  Member's SSN    -  -
  Last Name                         First Name                  M.I.
  Rank/Rate        Service Branch        Last T2 Exam   /  /     Class
                                                     MM/DD/YY
  Pane Status

     UIC

APPEND: Press <Esc> to abort
```

Figure 4. Append record form for Members in append mode.

After completing the data entry for a new record or after aborting an append, a sub-menu will appear at the bottom of the screen with several options:

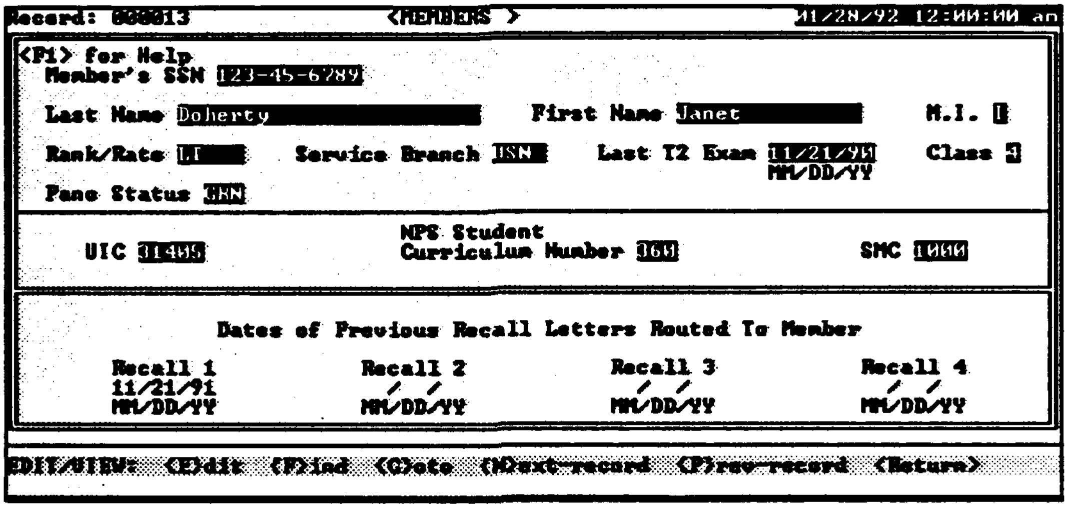
```
<Return>:add-another    {E}dit    {F}inished    <Del>
```

Pressing `<Return>` brings up a blank form for appending another new record. Pressing `{E}dit` allows editing of the currently displayed record. Selecting `{F}inished` appends the record (if completely entered and not marked for deletion) and returns you to the Main Menu. Pressing `<Del>` toggles between deleting and saving the current record. For example, assume you discover an error in a record that you have just entered and you want to delete it so that you can get the correct info later and re-enter it. Press `<Del>` to delete it. This allows you to then press `<Enter>` to keep entering new records without saving the erroneous one. When a record is "Deleted" a status indicator at the top of the screen says " *Deleted* ". In the next section, forms for editing each of the databases will be displayed. The forms look very similar to the forms for appending data.

Editing/viewing records

The "Edit/view" option of the Main Menu allows you to edit records in the active database. Editing is performed with one record displayed at a time. This option also provides a means to view all the data in a record of the active database on a single screen.

As can be seen in Figure 5, the Edit/view form for Members is very similar to the Append form for Members. The difference is that the sub-menu of options available is more extensive and that additional information is shown on the form. In the lower portion of the Edit Members form the dates of recall letters previously printed to the Member are displayed. This information can not be edited from the Edit/view screen but is for viewing only. Editing of recall dates will be discussed in Chapter 4.



```
Record: 000013              <MEMBERS >                    01/28/92 12:00:00 am
<F1> for Help
  Member's SSN 123-45-6789

  Last Name Doherty                 First Name Janet              M.I. I

  Rank/Rate LT    Service Branch USN    Last T2 Exam 11/21/90     Class 3
                                                     MM/DD/YY
  Pane Status SRN

                              NPS Student
     UIC 31485                Curriculum Number 360               SMC 1000


              Dates of Previous Recall Letters Routed To Member

        Recall 1           Recall 2           Recall 3           Recall 4
        11/21/91             /  /               /  /               /  /
        MM/DD/YY           MM/DD/YY           MM/DD/YY           MM/DD/YY

EDIT/VIEW: <E>dit <F>ind <G>oto <N>ext-record <P>rev-record <Return>
```

Figure 5. Edit/view form for Members.

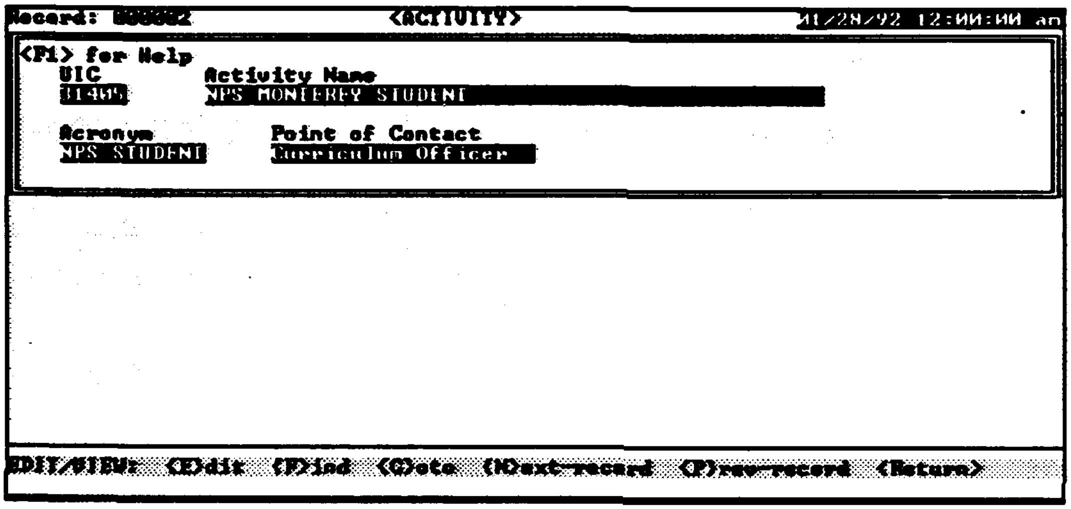
The actions of each of the Edit/view sub-menu commands are as follows:

`{E}dit`	`{E}dit` returns the cursor to the record displayed for further changes; the sub-menu options are not available. Entry of data in edit mode is the same as when appending a new record. Pressing `<Esc>` when in edit mode aborts the edit and the original data is displayed.
`{F}ind`	When editing Members, `{F}ind` enables you to select a specific record by specifying a member's SSN or name. (Part of a name or even a single letter can be used. PTARS will seek the first instance of whatever you type. Specifying the person's full name provides an exact match.) Since a name is not necessarily unique, the first occurrence of a match is shown on the screen. Specify a UIC when editing an Activity and a Curriculum Number when editing a Curriculum.
`{G}oto`	`{G}oto` enables you to go to a specific record number in the database. Record numbers are listed in the top left of the edit screen.
`{N}ext`	`{N}ext-record` brings up the next record. (By default, records are sorted by SSN. When a record is "found" by name, the database is sorted by last-name + first-name.)

`{P}rev` `{P}rev-record` brings up the prior record. Records are sorted as noted above.

`<Return>` `<Return>` brings you back to the Main Menu.

Figures 6 and 7 display the Edit/view forms for the Activity and Curriculum databases, respectively. The Append forms for these databases look the same with the exception of the sub-menus.


Record: 000002
<ACTIVITY>
01/28/92 12:00:00 am
<F1> for Help
EDIT/VIEW: <E>dit <F>ind <G>oto <N>ext-record <P>rev-record <Return>

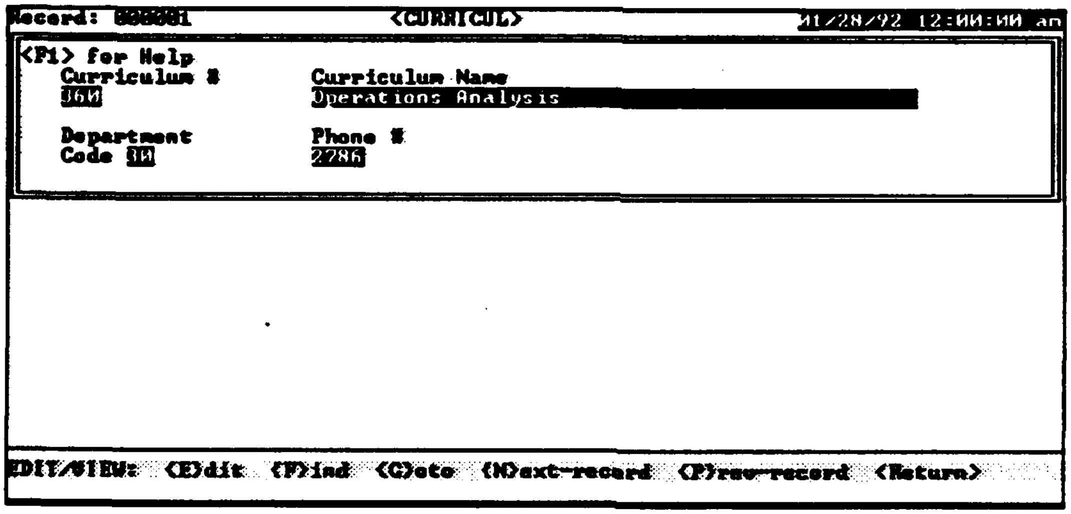
UIC
31405

Activity Name
NPS MONTEREY STUDENT

Acronym
NPS STUDENT

Point of Contact
Curriculum Officer

Figure 6. Edit/view form for Activity.


Record: 000001
<CURRICUL>
01/28/92 12:00:00 am
<F1> for Help
EDIT/VIEW: <E>dit <F>ind <G>oto <N>ext-record <P>rev-record <Return>

Curriculum #
360

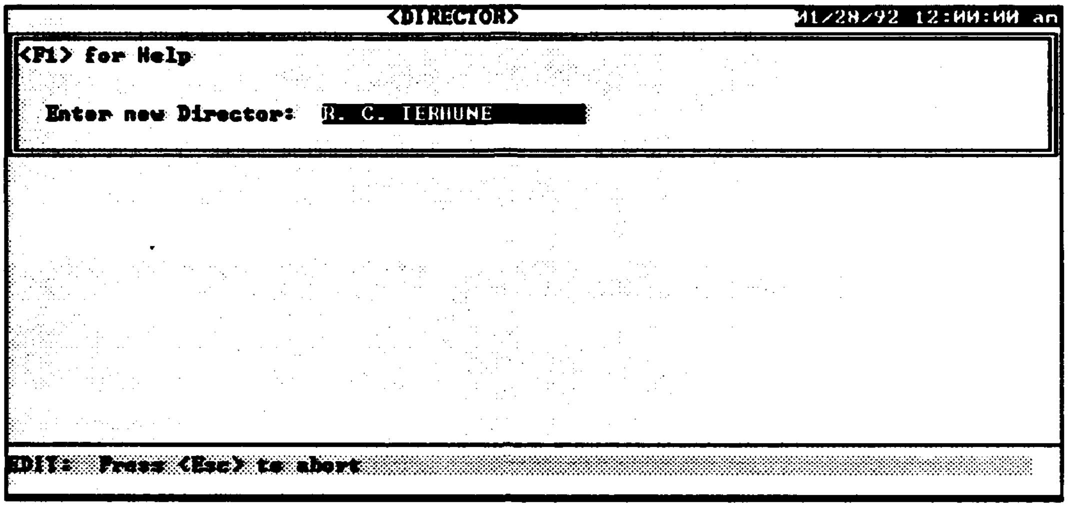
Curriculum Name
Operations Analysis

Department Code 30

Phone #
2786

Figure 7. Edit/view form for Curriculum.

Figure 8 shows the Edit/view form for Director. As discussed, Director can not be appended to or deleted. Hence, you are automatically in edit mode when you select this form. This is because there is only one clinic Director record and it must always contain a signature name.



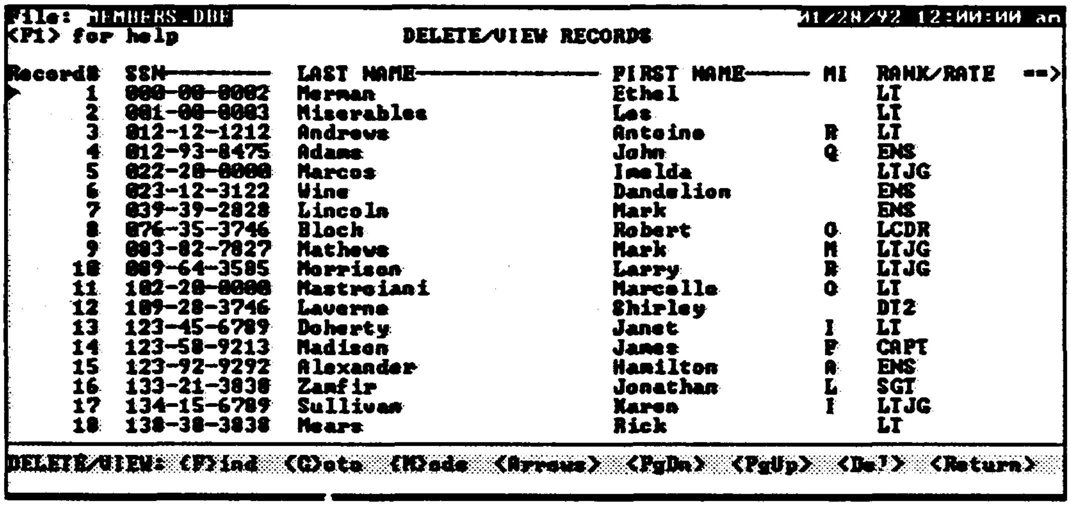
```
                         <DIRECTOR>                     01/28/92 12:00:00 am
<F1> for Help

  Enter new Director: R. C. TERHUNE

EDIT: Press <Esc> to abort
```

Figure 8. Edit/view form for Director.

Deleting/viewing records

Select the "Delete/view" option from the Main Menu to delete record(s) or to view multiple records on one screen. When a record is marked for deletion, an "*" appears to the left of the record. Figure 9 shows the Delete/view screen for Members. The Delete/view screens for Activities and for Curriculums operate in the same fashion as for Members. The only difference is the fields displayed on screen. When the "==>" appears in the upper right of the screen on the field column header line, additional fields exist for viewing. Pressing the right arrow key will pan the screen right to view the additional fields. Press the left arrow key to pan back to the left.

When a record is "Deleted" on the Delete/view screen, the record is not actually physically removed from the database; it is simply "marked" for deletion. This means that the record can still be recovered if you decide later that you want to "undelete" it. See the discussion of the `<Del>` action below for its operation. To permanently (physically) remove record(s) from a database, the database must be "packed". Chapter 6, "Utilities", provides further discussion of packing the database.



```
File: MEMBERS.DBF                                               01/28/92 12:00:00 am
<F1> for help                      DELETE/VIEW RECORDS

Record#  SSN---------  LAST NAME------------------  FIRST NAME------  MI  RANK/RATE  -->
►     1  000-00-0002   Merman                       Ethel                 LT
      2  001-00-0003   Miserables                   Les                   LT
      3  012-12-1212   Andrews                      Antoine           R   LT
      4  012-93-8475   Adams                        John              Q   ENS
      5  022-20-0000   Marcos                       Imelda                LTJG
      6  023-12-3122   Wine                         Dandelion             ENS
      7  039-39-2828   Lincoln                      Mark                  ENS
      8  076-35-3746   Bloch                        Robert            O   LCDR
      9  003-02-7827   Mathews                      Mark              M   LTJG
     10  009-64-3585   Morrison                     Larry             R   LTJG
     11  102-20-0000   Mastroiani                   Marcelle          O   LT
     12  109-28-3746   Laverne                      Shirley               DT2
     13  123-45-6789   Doherty                      Janet             I   LT
     14  123-50-9213   Madison                      James             P   CAPT
     15  123-92-9292   Alexander                    Hamilton          A   ENS
     16  133-21-3830   Zanfir                       Jonathan          L   SGT
     17  134-15-6789   Sullivan                     Karen             I   LTJG
     18  130-30-3830   Mears                        Rick                  LT
DELETE/VIEW: {F}ind {G}oto {M}ode <Arrows> <PgDn> <PgUp> <Del> <Return>
```

Figure 9. Delete/view screen for Members.

The actions of each of the Delete/view sub-menu commands are as follows:

`{F}ind`	Performs the same action as with the Edit/view form.
`{G}oto`	Performs the same action as with the Edit/view form.
`{M}ode`	`{M}ode` pops-up a selection of display modes for EGA and VGA video adapters: EGA, 25 or 43 lines; VGA, 25 or 50 lines. More lines on a screen are useful when deleting many members in a single session.
`<Arrows>`	`<Arrows>` refers to the direction keys for moving sideways to view panels of fields or up and down to place the cursor on different records.
`<PgDn>`	`<PgDn>` takes you to the next screen of consecutive records.
`<PgUp>`	`<PgUp>` takes you to the prior screen of consecutive records.
`<Del>`	`<Del>` toggles a deletion marker for a record. To mark a record for deletion, move the cursor to the record and press `<Del>`. When a record is marked for deletion an "*" appears to the left of the record. To unmark a deletion, make sure the cursor is on the correct marked record and press `<Del>` again.
`<Return>`	`<Return>` brings you back to the Main Menu.

4

Recalls

Recalls are the primary reason for the existence of PTARS. Each of the Service Branches require that members receive an annual dental examination (a "T2" exam), regardless of any prior need for dental treatment. Hence, members require notification prior to expiration of the 12 month period since their last exam (T2 or otherwise). PTARS automates the recall (notification) process by printing initial recall letters (Recall 1) and, if necessary, up to three follow-up letters (Recall 2 to Recall 4) to members.

The following topics are covered in this chapter:

- Printing recalls
- Printing recall lists
- Viewing/editing recall dates

The Recalls Menu is accessed by selecting the "Recalls" option from the Main Menu. As shown in Figure 10, three options are available from the Recalls Menu. Each of these options will be discussed in detail in this chapter.

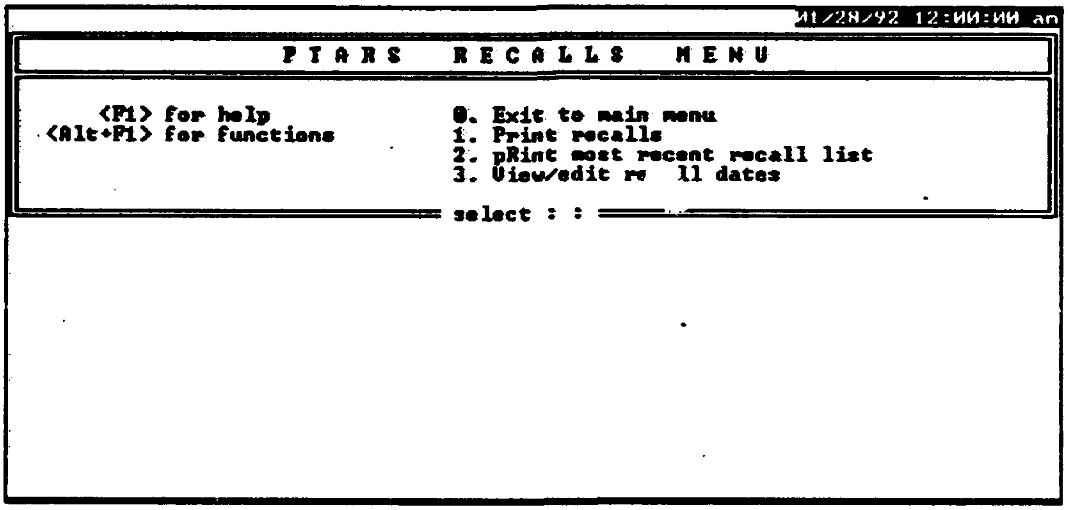
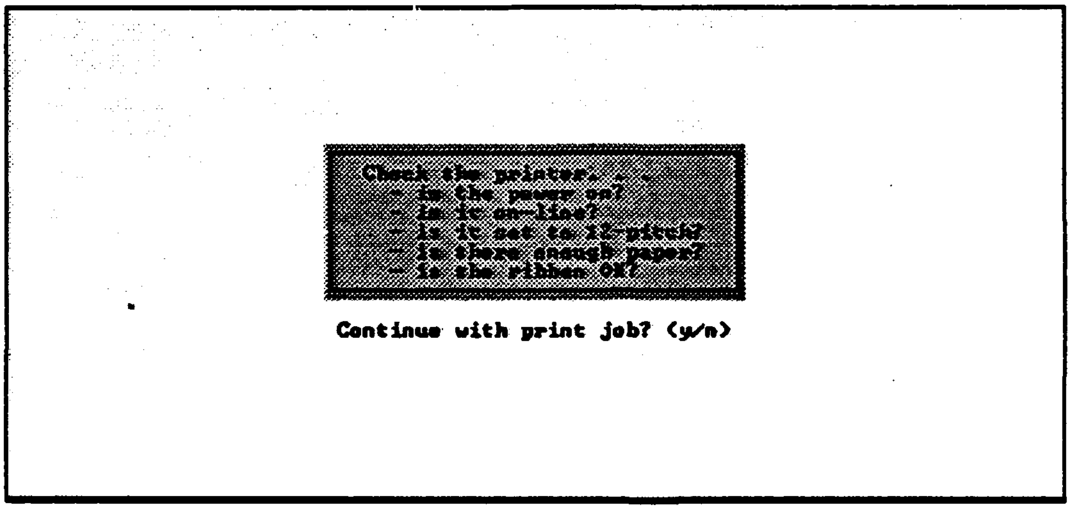

01/28/92 12:00:00 am
PTARS RECALLS MENU
<F1> for help
<Alt+F1> for functions
0. Exit to main menu
1. Print recalls
2. pRint most recent recall list
3. View/edit recall dates
select : :

Figure 10. Recalls Menu.

Printing recalls

Select "Print recalls" from the Recalls Menu to immediately start printing recall letters. Note that PTARS always backs-up the current MEMBERS.DBF to the hard disk prior to beginning its print routine. Also, note that prior to printing something, PTARS always presents a "Check the printer" notification. (See Figure 11.) You are also given the option to abort the print job. It is particularly important to heed this notification prior to printing recalls since the printing volume can be over 200 pages during this process and the print job can last over 45 minutes. Moreover, as discussed below, recall dates are inserted into the Members database. Any disruption of this process is problematic.



```
Check the printer. . .
  - is the power on?
  - is it on-line?
  - is it set to 12-pitch?
  - is there enough paper?
  - is the ribbon OK?

Continue with print job? (y/n)
```

Figure 11. "Check the printer" notification.

It is important that recalls be printed at approximately the same time every month (e.g., the last day of the month or the first day of the month). This will provide consistency in the intervals that members receive follow-up letters, should they be necessary.

When you print recalls, all recall letters are printed and recall letter dates are inserted into MEMBERS.DBF. (Note: The current MEMBERS.DBF is backed-up to the hard disk before printing.) "Print Recalls" also creates a file for each recall letter category which lists members for whom a recall letter is printed (Recall1.lst to Recall4.lst). The previous recall list files are saved with a .BAK extension should they need to be examined from DOS. The logic of recall printing is described following the important section below.

IMPORTANT - The recall letter printing module automatically inserts a new recall letter date into the Members database when a recall letter is printed. It also creates files (RECALL1.LST to RECALL4.LST) containing SSNs and names of members for whom a recall letter was printed. If a printer malfunction occurs or the print job is aborted for some reason, it will be necessary to compare the file listings of the most recent recall letters against the physically printed letters. **Members who are on the file listing, but for whom there is no useable printed recall letter, must have the new recall letter date deleted before the program can print a replacement recall letter.** This is because the printing module checks the existing recall dates to determine if an appropriate recall letter has already been printed.

If for some reason none or relatively few usable recall letters are printed (e.g., the printer was not turned on or there was an early paper jam), you may want to consider restoring the hard disk backup that was created just prior to printing the recalls and starting over. None of the new recall dates will exist on the backup and you can fix the printer and start fresh. See "Restoring backup(s)" in chapter 6. The logic of the recall process is described below:

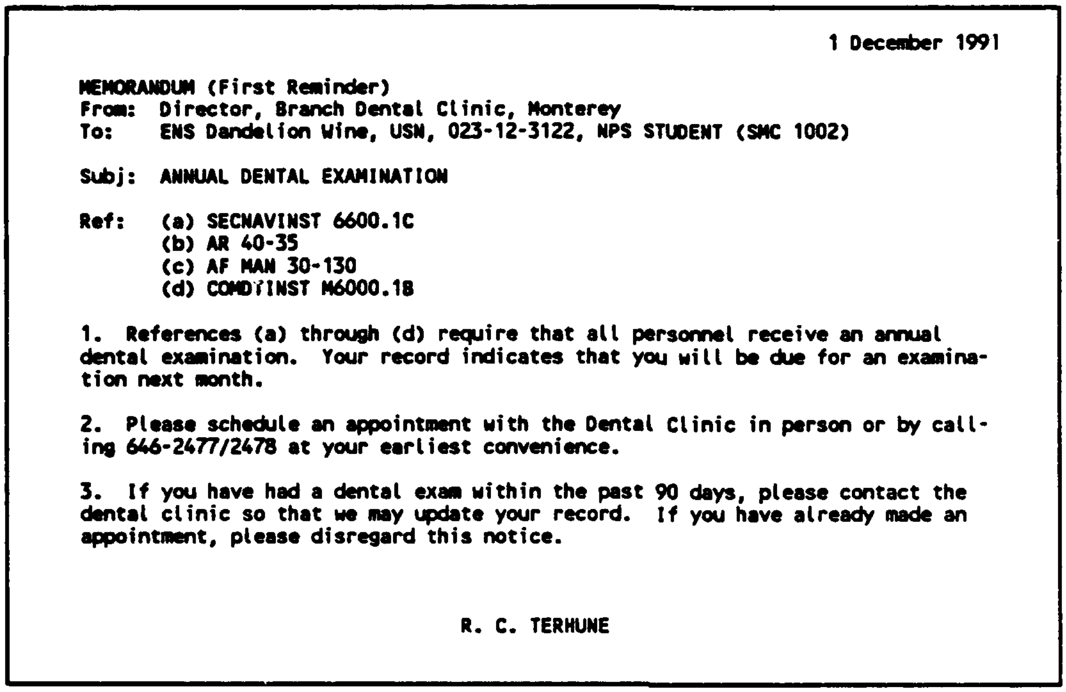
Recall 1 Recall 1 is triggered after at least 10 full months + 1 day have transpired since the member's last T2 exam. Prints a memo to the member and records the print date as Recall 1 date.

Recall 2 Recall 2 is triggered after at least 11 full months + 1 day have transpired since the member's last T2 exam, provided that Recall 1 date is in the database and that at least 25 days have transpired since Recall 1. Prints a memo to the member and records the print date as Recall 2 date.

Recall 3 Recall 3 is triggered after at least 12 full months + 1 day have transpired since the member's last T2 exam, provided that Recall 2 date is in the database and that at least 25 days have transpired since Recall 2. Prints a letter to the member and records the print date as Recall 3 date.

Recall 4 Recall 4 is triggered after at least 13 full months + 1 day have transpired since the member's last T2 exam, provided that Recall 3 date is in the database and that at least 25 days have transpired since Recall 3. Prints the letter to the member's superior (i.e., Curriculum Officer for students or to Activity POC for non-students) and records the print date as Recall 4 date.

Example recall letters 1 through 4 are shown in Figures 11 through 14 on the following three pages. Note that the text of Recall 4 indicates that Recall 3 is included as an enclosure. Thus, when routing Recall 4 letters a copy of Recall 3 should be attached. Copies of recall letters can be made by printing from double-copy paper, or alternatively, Xerox copies of just letters 3 and 4 can be made before routing them. The volume of these two letters is historically very low.



1 December 1991

MEMORANDUM (First Reminder)
From: Director, Branch Dental Clinic, Monterey
To: ENS Dandelion Wine, USN, 023-12-3122, NPS STUDENT (SMC 1002)

Subj: ANNUAL DENTAL EXAMINATION

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

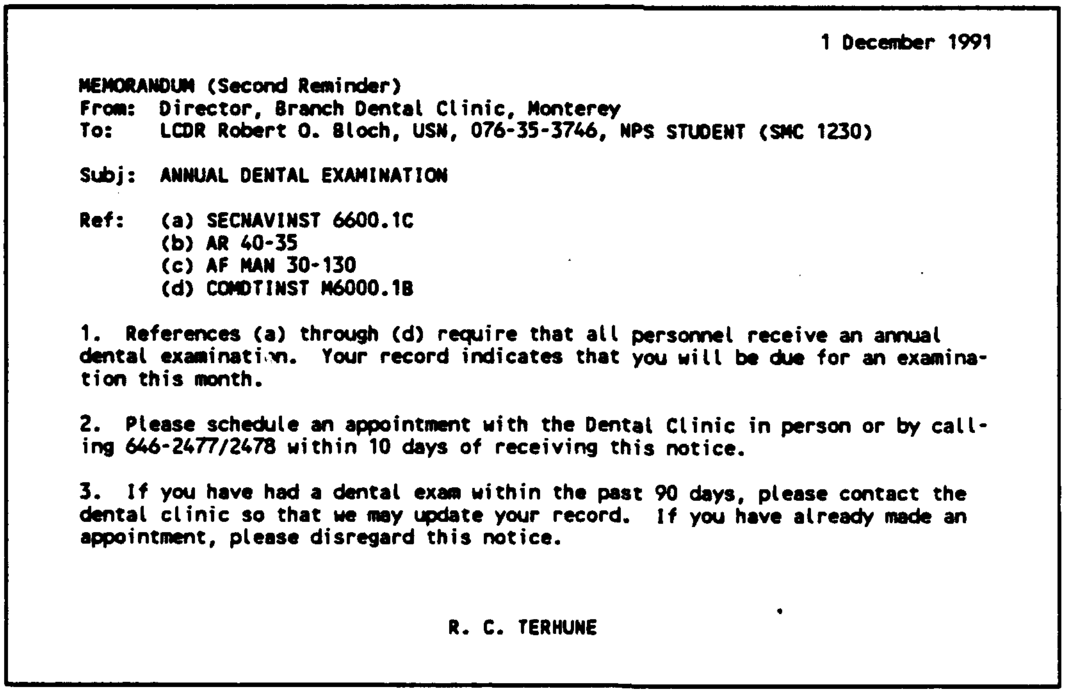
1. References (a) through (d) require that all personnel receive an annual dental examination. Your record indicates that you will be due for an examination next month.

2. Please schedule an appointment with the Dental Clinic in person or by calling 646-2477/2478 at your earliest convenience.

3. If you have had a dental exam within the past 90 days, please contact the dental clinic so that we may update your record. If you have already made an appointment, please disregard this notice.

R. C. TERHUNE

Figure 11. Example Recall 1 memorandum.



1 December 1991

MEMORANDUM (Second Reminder)
From: Director, Branch Dental Clinic, Monterey
To: LCDR Robert O. Bloch, USN, 076-35-3746, NPS STUDENT (SMC 1230)

Subj: ANNUAL DENTAL EXAMINATION

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

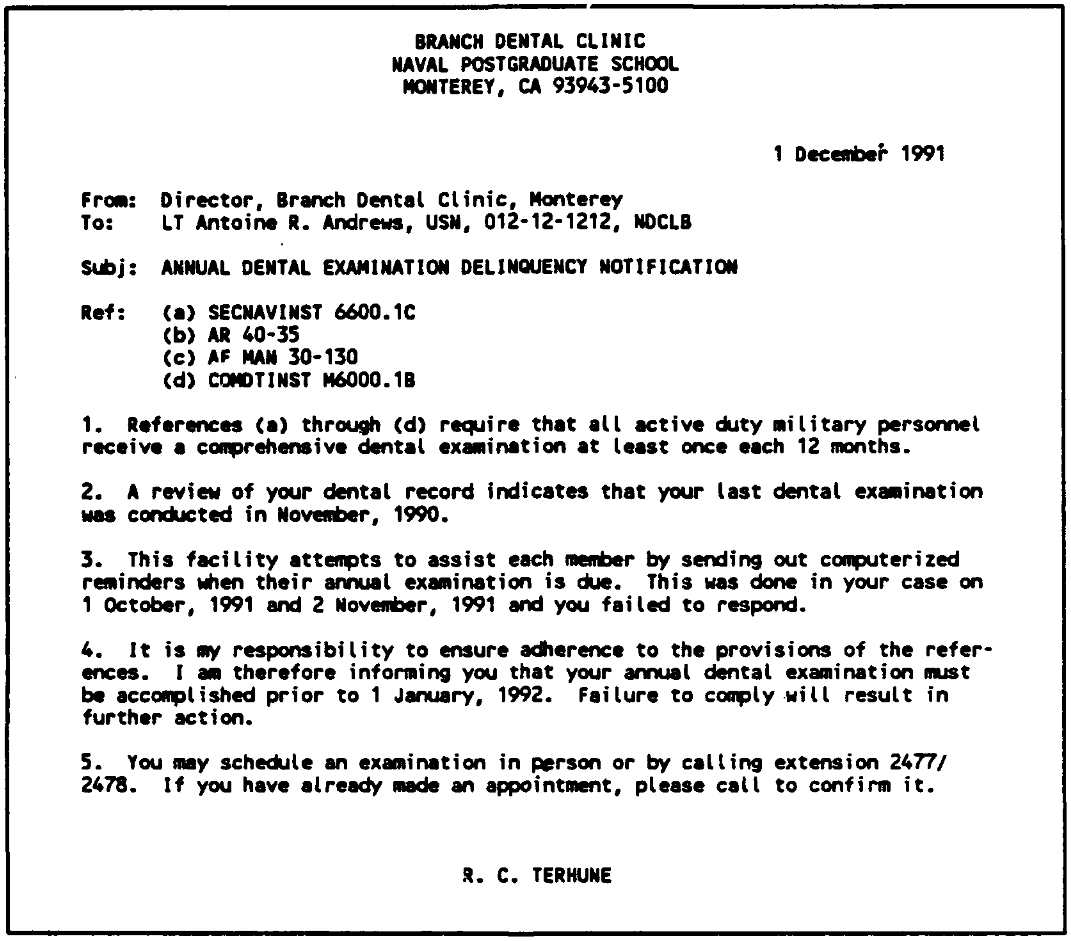
1. References (a) through (d) require that all personnel receive an annual dental examination. Your record indicates that you will be due for an examination this month.

2. Please schedule an appointment with the Dental Clinic in person or by calling 646-2477/2478 within 10 days of receiving this notice.

3. If you have had a dental exam within the past 90 days, please contact the dental clinic so that we may update your record. If you have already made an appointment, please disregard this notice.

R. C. TERHUNE

Figure 12. Example Recall 2 memorandum.



BRANCH DENTAL CLINIC
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5100

1 December 1991

From: Director, Branch Dental Clinic, Monterey
To: LT Antoine R. Andrews, USN, 012-12-1212, NDCLB

Subj: ANNUAL DENTAL EXAMINATION DELINQUENCY NOTIFICATION

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

1. References (a) through (d) require that all active duty military personnel receive a comprehensive dental examination at least once each 12 months.

2. A review of your dental record indicates that your last dental examination was conducted in November, 1990.

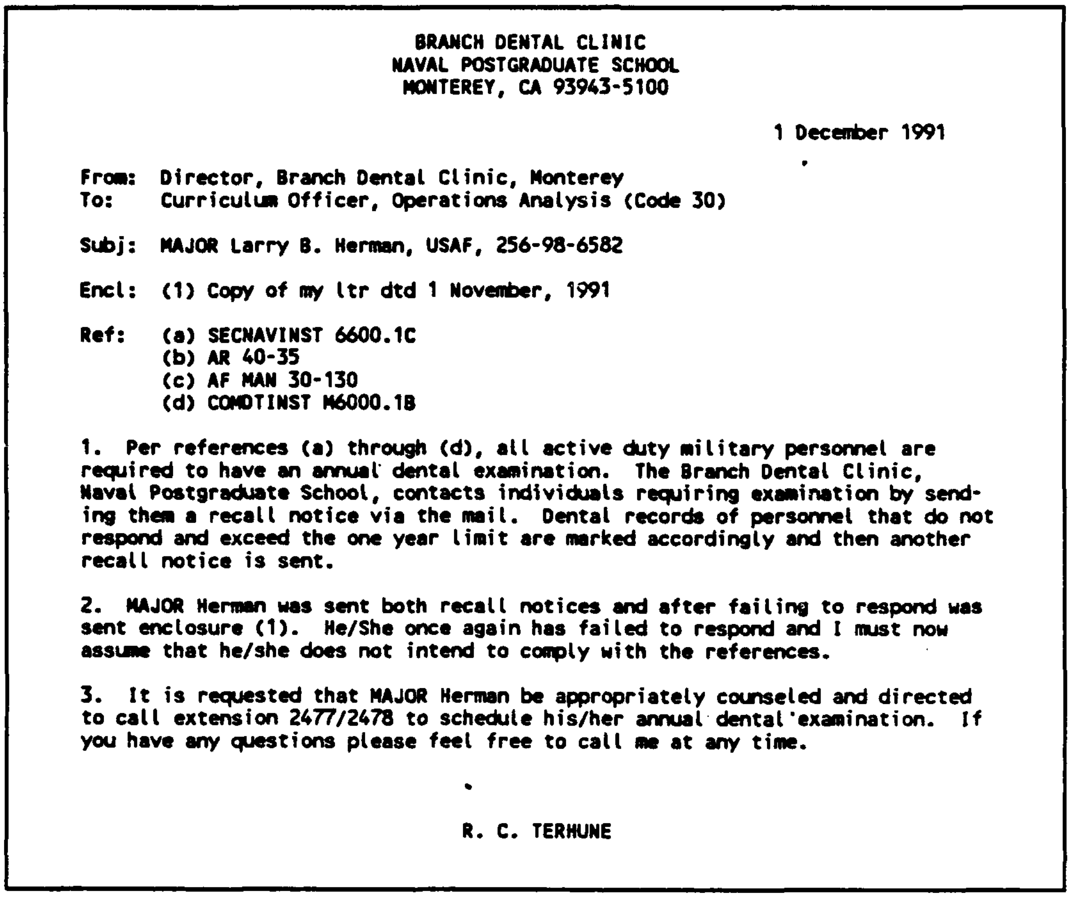
3. This facility attempts to assist each member by sending out computerized reminders when their annual examination is due. This was done in your case on 1 October, 1991 and 2 November, 1991 and you failed to respond.

4. It is my responsibility to ensure adherence to the provisions of the references. I am therefore informing you that your annual dental examination must be accomplished prior to 1 January, 1992. Failure to comply will result in further action.

5. You may schedule an examination in person or by calling extension 2477/2478. If you have already made an appointment, please call to confirm it.

R. C. TERHUNE

Figure 13. Example Recall 3 letter.



BRANCH DENTAL CLINIC
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5100

1 December 1991

From: Director, Branch Dental Clinic, Monterey
To: Curriculum Officer, Operations Analysis (Code 30)

Subj: MAJOR Larry B. Herman, USAF, 256-98-6582

Encl: (1) Copy of my ltr dtd 1 November, 1991

Ref: (a) SECNAVINST 6600.1C
(b) AR 40-35
(c) AF MAN 30-130
(d) COMDTINST M6000.1B

1. Per references (a) through (d), all active duty military personnel are required to have an annual dental examination. The Branch Dental Clinic, Naval Postgraduate School, contacts individuals requiring examination by sending them a recall notice via the mail. Dental records of personnel that do not respond and exceed the one year limit are marked accordingly and then another recall notice is sent.

2. MAJOR Herman was sent both recall notices and after failing to respond was sent enclosure (1). He/She once again has failed to respond and I must now assume that he/she does not intend to comply with the references.

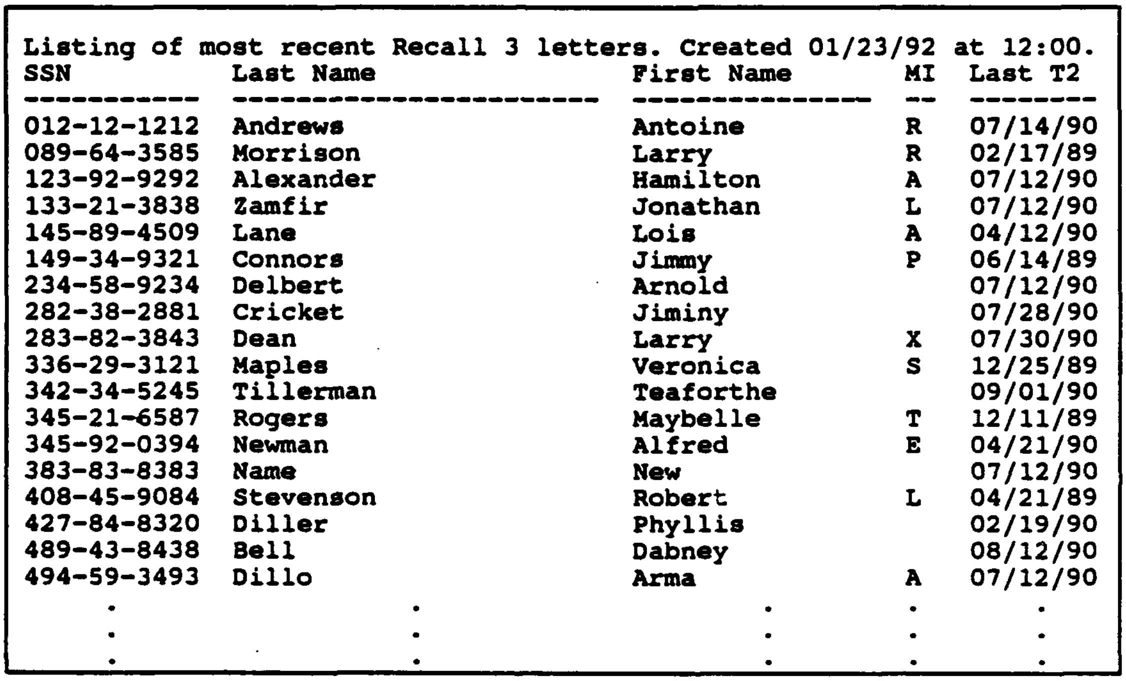
3. It is requested that MAJOR Herman be appropriately counseled and directed to call extension 2477/2478 to schedule his/her annual dental examination. If you have any questions please feel free to call me at any time.

R. C. TERHUNE

Figure 14. Example Recall 4 letter.

Printing recall lists

Select "pRint most recent recall list" from the Recalls Menu. This option lists (to the printer only) the most recent recall letter information. (The same information is listed to the screen during the printing of the recall letters.) Use this option in the event of a printer malfunction when printing recall letters to compare physical letters against what the program "thinks" it printed. Popup options are presented to select which listing to print. Figure 15 depicts an example listing of Recall 3.

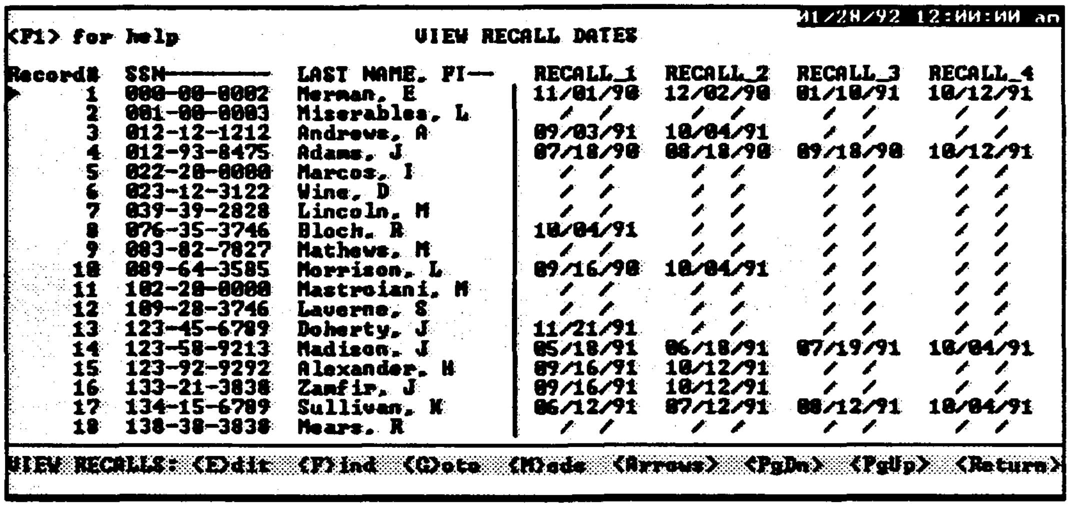


```
Listing of most recent Recall 3 letters. Created 01/23/92 at 12:00.
SSN          Last Name                 First Name        MI  Last T2
-----------  -----------------------   ---------------   --  --------
012-12-1212  Andrews                   Antoine           R   07/14/90
089-64-3585  Morrison                  Larry             R   02/17/89
123-92-9292  Alexander                 Hamilton          A   07/12/90
133-21-3838  Zamfir                    Jonathan          L   07/12/90
145-89-4509  Lane                      Lois              A   04/12/90
149-34-9321  Connors                   Jimmy             P   06/14/89
234-58-9234  Delbert                   Arnold                07/12/90
282-38-2881  Cricket                   Jiminy                07/28/90
283-82-3843  Dean                      Larry             X   07/30/90
336-29-3121  Maples                    Veronica          S   12/25/89
342-34-5245  Tillerman                 Teaforthe             09/01/90
345-21-6587  Rogers                    Maybelle          T   12/11/89
345-92-0394  Newman                    Alfred            E   04/21/90
383-83-8383  Name                      New                   07/12/90
408-45-9084  Stevenson                 Robert            L   04/21/89
427-84-8320  Diller                    Phyllis               02/19/90
489-43-8438  Bell                      Dabney                08/12/90
494-59-3493  Dillo                     Arma              A   07/12/90
     .            .                          .           .       .
     .            .                          .           .       .
     .            .                          .           .       .
```

Figure 15. Example listing of Recall 3.

Viewing/editing recall dates

The "View/edit recall dates" option of the Recalls Menu provides a means for viewing recall letter dates for multiple records and for accessing individual records for recall letter date editing. This facility should be used in conjunction with the previously discussed recall listings in the event of a printer malfunction when printing recall letters. The sub-menu options of the View Recalls screen shown in Figure 16 are the same as the like-named options discussed in Chapter 3 for the Delete/view screen. Since recall dates are a subset of the fields in the Members database, records can not be deleted using View Recalls.



```
                                                           01/28/92 12:00:00 am
<F1> for help                 VIEW RECALL DATES

Record#  SSN-----------  LAST NAME, FI--  RECALL_1  RECALL_2  RECALL_3  RECALL_4
▶     1  000-00-0002     Merman, E       |11/01/90  12/02/90  01/10/91  10/12/91
      2  001-00-0003     Miserables, L   |  /  /      /  /      /  /      /  /
      3  012-12-1212     Andrews, A      |09/03/91  10/04/91    /  /      /  /
      4  012-93-8475     Adams, J        |07/18/90  08/18/90  09/18/90  10/12/91
      5  022-20-0000     Marcos, I       |  /  /      /  /      /  /      /  /
      6  023-12-3122     Wine, D         |  /  /      /  /      /  /      /  /
      7  039-39-2828     Lincoln, M      |  /  /      /  /      /  /      /  /
      8  076-35-3746     Bloch, R        |10/04/91    /  /      /  /      /  /
      9  083-82-7827     Mathews, M      |  /  /      /  /      /  /      /  /
     10  089-64-3585     Morrison, L     |09/16/90  10/04/91    /  /      /  /
     11  102-20-0000     Mastroiani, M   |  /  /      /  /      /  /      /  /
     12  109-28-3746     Laverne, S      |  /  /      /  /      /  /      /  /
     13  123-45-6789     Doherty, J      |11/21/91    /  /      /  /      /  /
     14  123-58-9213     Madison, J      |05/18/91  06/18/91  07/19/91  10/04/91
     15  123-92-9292     Alexander, H    |09/16/91  10/12/91    /  /      /  /
     16  133-21-3838     Zamfir, J       |09/16/91  10/12/91    /  /      /  /
     17  134-15-6789     Sullivan, K     |06/12/91  07/12/91  08/12/91  10/04/91
     18  138-38-3838     Mears, R        |  /  /      /  /      /  /      /  /
VIEW RECALLS: <E>dit <F>ind <G>oto <M>ode <Arrows> <PgDn> <PgUp> <Return>
```

Figure 16. View Recalls screen.

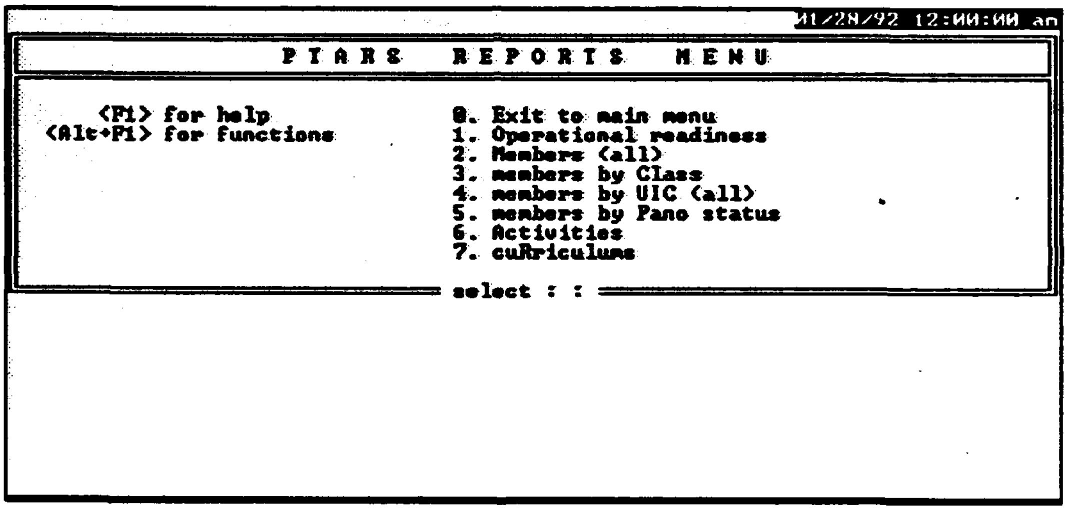
As discussed previously, the purpose of editing recall letter dates is to enable PTARS to print a replacement recall letter. If a recall letter date is present for a given recall letter, the program will only be able to print the *next* letter when the eligibility date for the *next* recall letter arrives. To reprint a letter, the recall letter date *must* be deleted *and* there *must not* be a subsequent recall letter date present. If this sounds confusing, reread the previous coverage of "Printing Recalls".

To edit a member's recall dates, press {E}. The current row of the display will be highlighted and placed into edit mode. Use normal editing and movement keys to edit the date(s). Note that edited dates are checked for chronological consistency as well as general date validity (i.e., can not be later than the current date, must have a prior recall, can not be missing a recall between recalls, values must be chronologically correct for existent recalls).

5

Reports

This chapter discusses the various reports available in PTARS and provides several example figures to preview the look of the reports. The Reports Menu, shown in Figure 17, is accessed from the Main Menu by pressing {P}. The Operational Readiness Report is available to both the screen and the printer. The other reports (rosters) are sent to the printer only.

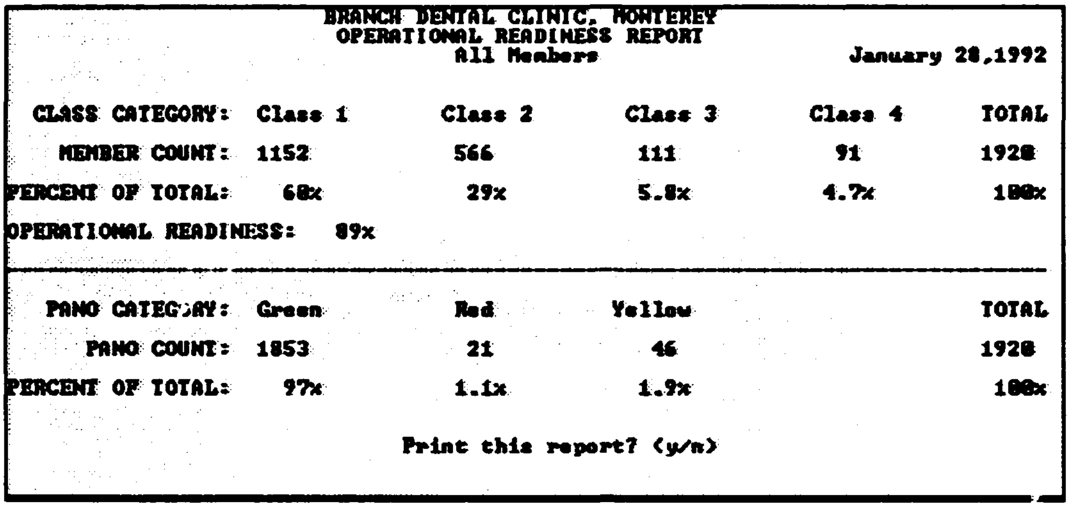


```
                                                        01/28/92 12:00:00 am
                    P T A R S   R E P O R T S   M E N U

      <F1> for help                0. Exit to main menu
 <Alt+F1> for functions            1. Operational readiness
                                   2. Members (all)
                                   3. members by Class
                                   4. members by UIC (all)
                                   5. members by Pano status
                                   6. Activities
                                   7. cuRriculums
                         ========= select : : =========
```

Figure 17. Reports Menu.

Operational readiness

The Operational Readiness Report provides counts and percentages of members in each of the dental CLASS categories. The report is initially displayed to the screen and you are given the option of printing it. Operational Readiness is defined as the percentage of all members served by the clinic who are classified as CLASS 1 or 2. As can be seen in Figure 18, the Operational Readiness percentage is a simple summation of the CLASS 1 and CLASS 2 percentages.



```
                          BRANCH DENTAL CLINIC, MONTEREY
                           OPERATIONAL READINESS REPORT
                                   All Members                  January 28,1992

  CLASS CATEGORY:  Class 1        Class 2        Class 3        Class 4       TOTAL

    MEMBER COUNT:  1152           566            111            91            1920

PERCENT OF TOTAL:    68%           29%           5.8%           4.7%          100%

OPERATIONAL READINESS:   89%
-------------------------------------------------------------------------------------
   PANO CATEGORY:  Green          Red          Yellow                         TOTAL

      PANO COUNT:  1853           21             46                           1920

PERCENT OF TOTAL:    97%          1.1%          1.9%                          100%

                              Print this report? (y/n)
```

Figure 18. Operational Readiness Report to screen.

Also included in the report are counts and percentages of members whose Pano X-rays are in a given status. Three Pano status categories exist and are designated by standard color designations:

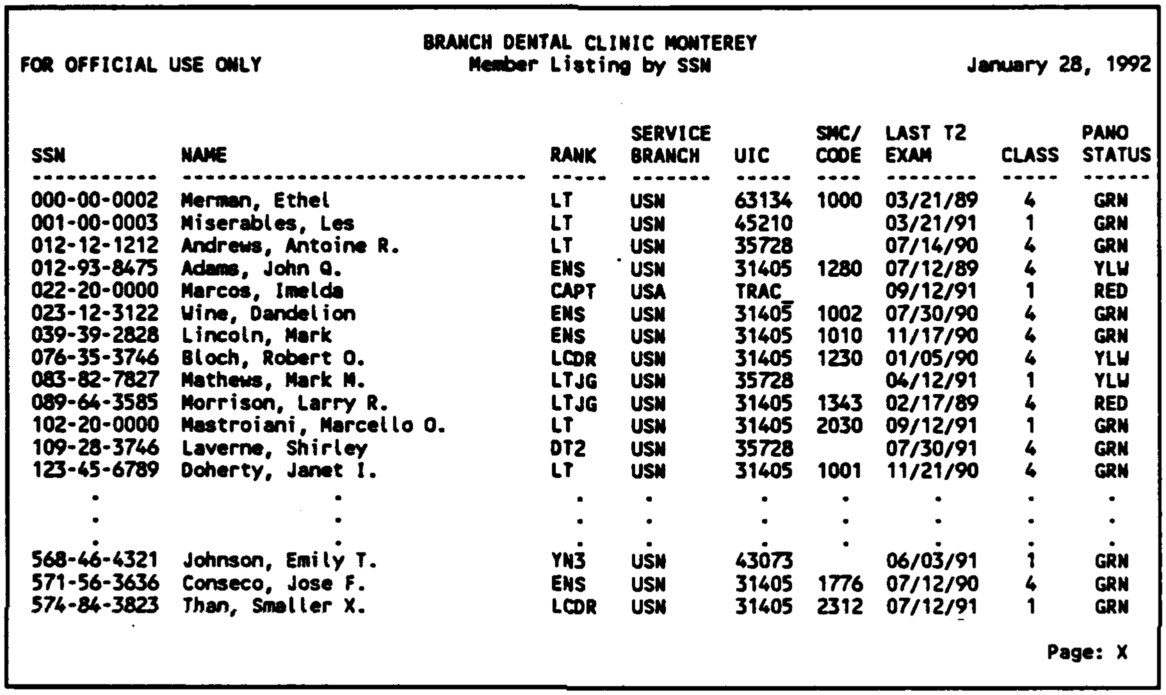
GRN (Green)	Pano *is* accepted and on-file
RED	Pano *has* been duplicated and forwarded
YLW (Yellow)	Pano *is not* on-file and *has not* been duplicated and forwarded

Rosters

The remaining reports available from the Reports Menu are basically rosters sorted on various fields of interest. After selecting any of the Members reports a popup will offer a selection of whether to list members by SSN or alphabetically. If printing Members by dental CLASS, a popup will allow selection of a specific CLASS or all members. If printing Members by Pano status, a popup will allow selection of a specific status or all members. Figure 19 provides an example roster of Members listed by SSN that could be printed by selecting option 2, "Members (all)", from the Reports Menu.

Selections 6 and 7 from the Reports Menu print complete rosters of the Activities and the Curriculums contained in their respective PTARS databases.

Periodic comparison of Member rosters against data from both PSD and the Registrar will help keep member data up-to-date. Current listings of the Curriculums at NPS should also be obtained from the Registrar so that the Curriculum database can be kept up-to-date.



```
                               BRANCH DENTAL CLINIC MONTEREY
FOR OFFICIAL USE ONLY              Member Listing by SSN                 January 28, 1992


                                                 SERVICE        SMC/ LAST T2          PANO
SSN          NAME                          RANK  BRANCH   UIC   CODE EXAM      CLASS STATUS
-----------  ----------------------------- ----- -------  ----- ---- --------  ----- ------
000-00-0002  Merman, Ethel                 LT    USN      63134 1000 03/21/89    4     GRN
001-00-0003  Miserables, Les               LT    USN      45210      03/21/91    1     GRN
012-12-1212  Andrews, Antoine R.           LT    USN      35728      07/14/90    4     GRN
012-93-8475  Adams, John Q.                ENS   USN      31405 1280 07/12/89    4     YLW
022-20-0000  Marcos, Imelda                CAPT  USA      TRAC_      09/12/91    1     RED
023-12-3122  Wine, Dandelion               ENS   USN      31405 1002 07/30/90    4     GRN
039-39-2828  Lincoln, Mark                 ENS   USN      31405 1010 11/17/90    4     GRN
076-35-3746  Bloch, Robert O.              LCDR  USN      31405 1230 01/05/90    4     YLW
083-82-7827  Mathews, Mark M.              LTJG  USN      35728      04/12/91    1     YLW
089-64-3585  Morrison, Larry R.            LTJG  USN      31405 1343 02/17/89    4     RED
102-20-0000  Mastroiani, Marcello O.       LT    USN      31405 2030 09/12/91    1     GRN
109-28-3746  Laverne, Shirley              DT2   USN      35728      07/30/91    4     GRN
123-45-6789  Doherty, Janet I.             LT    USN      31405 1001 11/21/90    4     GRN
     .                        .             .     .        .    .       .      .     .
     .                        .             .     .        .    .       .      .     .
     .                        .             .     .        .    .       .      .     .
568-46-4321  Johnson, Emily T.             YN3   USN      43073      06/03/91    1     GRN
571-56-3636  Conseco, Jose F.              ENS   USN      31405 1776 07/12/90    4     GRN
574-84-3823  Than, Smaller X.              LCDR  USN      31405 2312 07/12/91    1     GRN

                                                                              Page: X
```

Figure 19. Members (all) roster sorted by SSN.

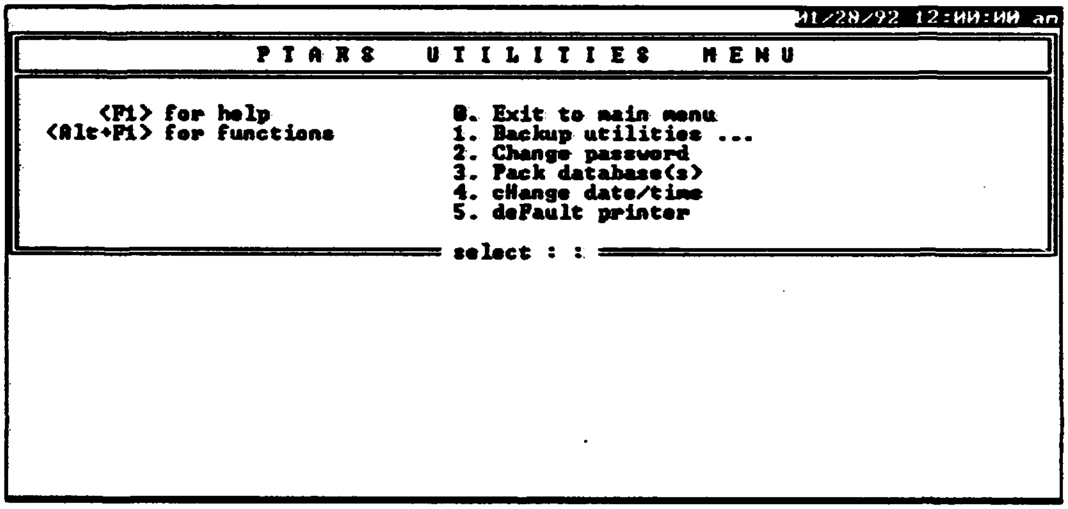
6

Utilities

This Chapter explains the various utilities included with PTARS that support proper maintenance of the databases. The Utilities Menu is accessed by pressing {U} from the Main Menu and is shown in Figure 20.

It contains the following sections:

- Backup utilities
- Changing the password
- Packing the database(s)
- Changing the date or time
- Selecting the default printer



```
                                                         01/28/92 12:00:00 am
                  P T A R S   U T I L I T I E S   M E N U

      <F1> for help                   0. Exit to main menu
  <Alt+F1> for functions              1. Backup utilities ...
                                      2. Change password
                                      3. Pack database(s)
                                      4. cHange date/time
                                      5. deFault printer
                                  select : :
```

Figure 20. Utilities Menu.

Backup utilities

The backup utilities are a collection of utilities related to backing-up and restoring the four database files MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and

DIRECTOR.DBF. The Backup Utilities Menu, shown in Figure 21, is accessed from the Utilities Menu by pressing {B}. Each of the menu selections will be discussed in the sub-sections below.

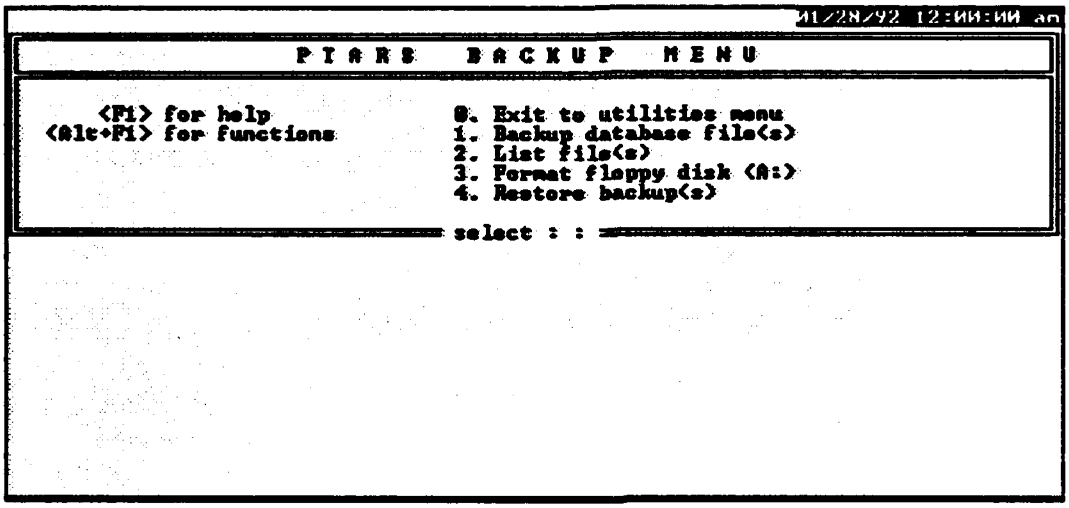


Figure 21. Backup Utilities Menu.

Backing-up database(s)
When you first select Backup, a popup will appear allowing you to select whether you want to back-up to the hard disk or the floppy disk in drive A. Next, another popup appears to let you select which database file(s) (i.e,. MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, DIRECTOR.DBF, or all) to back-up. Once your selection is made, Backup copies the selected file(s) to the destination drive. Backing-up to a floppy keeps a reserve copy of the data that can be restored in case something happens to the computer, hard disk, or the data. Backing-up to the hard disk is convenient for short-term backups, but is *not* sufficient for best reliability. Note that the PTARS program presents the option to back-up the databases to the hard disk prior to quitting a session.

Your data *should* be backed up to a floppy disk weekly and immediately following input or editing sessions involving many records. It is a good idea to keep two or three backup floppies in rotation—writing over the oldest backup each time. *Always* label your backups to floppy disk with the file names and their creation dates. This will help you to identify your backups later if you need to restore them. Hint: use a pencil to label your backups; you can use several floppy disks over and over again by erasing and writing the new information.

Remember, there is only one way to ensure the safety of your data: rigorous adherence to a regular program of backing-up.

Listing files

A popup menu allows selecting the hard disk PTARS subdirectory or floppy disk A: for listing files. Either just database files can be displayed or all files can be displayed. When database files are displayed the following information is included: file name, number of records, last update, file size, total bytes in database files, and bytes remaining on the drive. When all files are displayed, file names are listed and total bytes used in the files and bytes remaining on the drive are presented.

This utility is useful for identifying files that might already exist on a diskette that will be used for backups.

Formatting a floppy disk

Formats a 360Kbyte or a 1.2Mbyte floppy disk (5 ¼") placed in drive A. A popup presents three options:

360K --> 360K	Formats from a 360K capacity drive to a 360K floppy
1.2M --> 360K	Formats from a 1.2M capacity drive to a 360K floppy
1.2M --> 1.2M	Formats from a 1.2M capacity drive to a 1.2M floppy

The first number indicates the actual drive-type. For example, your machine may only be capable of formatting 360K floppy disks, as in the first option. The second number indicates the floppy disk formatted capacity. A new floppy disk must be formatted so that the Disk Operating System (DOS) can read and write data to it.

Restoring backup(s)

When you select "Restore backup(s)", a popup enables selectively replacing database file(s) with backups from the hard disk or a floppy disk.

At the end of every session with PTARS you are presented with the option to backup the databases to the hard disk. If you choose to do so, four backup database files, MEM_BU.DBF, ACT_BU.DBF, CUR_BU.DBF, and DIR_BU.DBF are created in the PTARS subdirectory of your hard drive. These files can be restored (either singly or together) to MEMBERS.DBF, ACTIVITY.DBF, CURRICUL.DBF, and DIRECTOR.DBF, respectively. The restored backups overwrite the current database file(s).

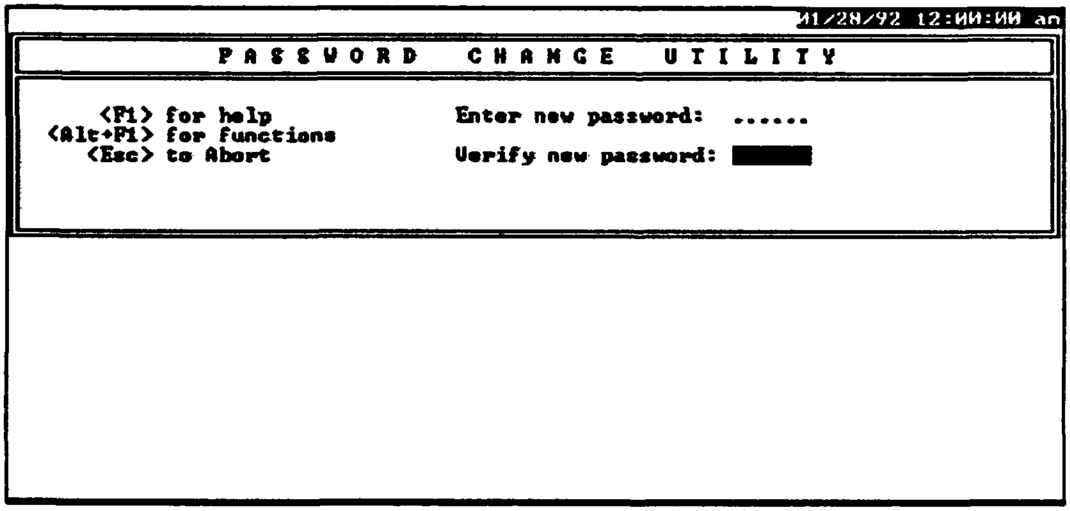
Note that backing-up to the hard drive does not protect your data from hard drive or computer failure since the backups reside on the same machine as the original data. The feature is useful, however, if your original data becomes corrupted for some reason but your backups are still OK. In addition, it may be useful in the event you have experienced a printer malfunction (e.g., your printer ribbon gave up the ghost) and you have many unusable recall letters. Rather than editing recall dates and printing again, it may be advantageous to restore the backup of MEMBERS.DBF (which PTARS always makes before printing recalls) and start over.

A final method of restoring any database is to manually copy the file using DOS commands. This method should never be necessary since the capability is built into PTARS. **If for some reason you should need to manually restore a *.DBF file, be sure that any like-named compound index file (*.CDX) is erased** (e.g., from the DOS prompt: `del c:\ptars\members.cdx`) This is because a unique index file is created and updated by PTARS for each database. If the index file does not "belong" to the specific version of a database, PTARS will not perform properly and will give an error notification.

Changing the password

You can change the current password to a new password (it must have 6 characters). Make sure that you *remember* the new password. If you ever forget your new password, copy the file NPS_MISC.DBF from disk 3 of your *backup copies* of the installation disks to the sub-directory \PTARS (e.g., `copy a:\nps_misc.dbf c:\ptars`). The original password is "zyxabc". This default password should be changed immediately after you install PTARS. (If you can read it here, so can someone else.) Note that the password is encrypted in the file NPS_MISC.DBF and cannot be deciphered if it is forgotten.

Figure 22 shows the screen for changing the password. As you type your new password, a dot will appear for each character typed. As shown in the figure, to verify that you typed what you thought you typed, PTARS prompts for a second entry of your new password. If the two entries do not match, the password change will be aborted.



```
                                                          01/28/92 12:00:00 am
                    P A S S W O R D   C H A N G E   U T I L I T Y

      <F1> for help                   Enter new password:  ......
  <Alt+F1> for functions
     <Esc> to Abort                   Verify new password:
```

Figure 22. Password change screen.

For effective security it is a good idea to periodically change your password. If an unauthorized individual inadvertently (or even deliberately) changes or damages your data, it could be a catastrophe. Regarding security, just think about having to re-enter over 1900 records!

Packing the database(s)

Packing the database(s) *permanently* deletes records "marked" for deletion from one or all of the three primary databases: MEMBERS.DBF, ACTIVITY.DBF, and CURRICUL.DBF. It also physically sorts the databases. MEMBERS.DBF is sorted in ascending order by SSN; ACTIVITY.DBF is sorted in ascending order by UIC; and CURRICUL.DBF is sorted in ascending order by curriculum number. Packing improves the performance of PTARS by reducing the physical size of the database(s) and reorders the records by the primary key. Note that the effects of packing are *not* "undoable". An informational prompt will appear upon quitting a session when 10% or more of the MEMBER.DBF records are marked for deletion. You should heed the prompt and pack the database (unless you have some good reason not to).

Changing the date or time

After selecting the "Change date or time" option a popup for selecting which to change (date or time) appears. After your selection is made you are prompted to enter the date or time using the example format shown on the screen. The system date or time can also be changed when starting the PTARS program. As part of the opening screen routine the user is prompted to verify the system date and time. If the system date or time displayed is incorrect, enter the correct date or time using the example format shown on the screen.

Selecting the default printer

You should select the default printer before printing anything from PTARS for the first time. After choosing this option from the Utilities Menu, PTARS pops-up two common printer emulations for dot matrix printers: (1) Epson E/F/J/RX/LQ emulation and (2) IBM Proprinter emulation. The emulation you select becomes the default for all subsequent sessions. The Epson emulation is supported by the majority of 9 pin dot matrix printers and PTARS uses it as the initial default. The default printer identifier is stored in a field in the NPS_MISC.DBF file.

Optimizing PTARS

This appendix identifies several ways that you can optimize the performance of PTARS if you have certain hardware or software capabilities. It contains the following sections:

- Disk defrag/compress
- Memory
- Config.sys
- Pack the database(s)

Disk defrag/compress

The performance of PTARS can be significantly improved if a disk defragment/compression procedure is performed on your hard drive periodically. Over time the database files will become fragmented as records are appended, edited and deleted. This slows down disk reads and writes since each file is no longer one contiguous piece; files can become many pieces scattered all over the disk. Defragment/compression utilities are available commercially.

Memory

PTARS will take advantage of all types of computer memory. If your computer is configured correctly, PTARS' performance will be enhanced. **Note that if you change your computer's memory configuration or add a disk cache program, you must re-install PTARS so that it operates optimally.**

Personal Computers (PC)s can contain three types of memory: conventional, expanded and extended.

Conventional Memory

All PCs can contain conventional memory (up to 640K). This is the memory that programs typically load into and run in. PTARS requires that you have at least 512K of conventional memory with at least 420K of it free after memory resident programs have been loaded. A minimum of 640K is *strongly* recommended.

Expanded Memory

The 8086 family of microprocessors have a physical address space of 1024K, or 1MB. The first 640K is the conventional memory space discussed above. The remaining 384K is reserved for use by read-only memory (RAM) and hardware device controllers. Also, within this area of memory, a 64K block can be reserved for use by an expanded memory manager which conforms to the Lotus/Intel/Microsoft interface specification (a LIM EMS Memory Manager).

The Expanded Memory Manager (EMM) administers expanded memory as a system resource that can be used by several applications at the same time and services EMS function calls. EMS memory is bank-switched memory that can be larger than the CPU's address space that is mapped into conventional memory via the EMS page frame.

On machines with expanded memory that is LIM 4.0 EMS compatible, PTARS uses the first 64K of expanded memory as "general purpose" memory and any remaining expanded memory to speed file I/O and to cache PTARS code segments.

To check how much EMS is currently being used by PTARS, look in the "About PTARS" box (by pressing `<F4>` or `<Alt+F1>`).

If you run on an 80386 or 80486 you're in luck! There are many inexpensive programs that use extended memory to emulate EMS, such as QEMM from Quarterdeck and 386MAX from Qualitas. MS-DOS 5.0 includes EMM386. On a 386 **always** use QEMM, 386Max, or other expanded memory managers. You'll be glad you did!

If you use a non-80386 processor you have several options. First, you could invest in an EMS board. These pieces of hardware, which usually work with both 8086/88 and 80286 processors, include substantial amounts of memory together with driver programs which provide the software interface to the board.

Extended Memory

Extended memory is memory that lies above the 1MB address range. It can be used directly by some operating systems (OS/2 and UNIX), but standard DOS cannot address it without the use of an Extended Memory Specification (XMS) driver, an interface that allows access to memory beyond 640K. Applications using this address space must be running in protected mode.

Extended memory cannot be used directly by PTARS until it is made to act like EMS. How you make extended memory act like expanded memory is dependent on your system, but typically you install a memory manager -- software that provides an EMS style (LIM 4.0) interface to extended memory. Once the extended memory is emulating EMS memory, PTARS will sense that it is there and make good use of it.

Config.sys

The system configuration file, CONFIG.SYS, contains certain commands that are checked and executed when you start up your computer. These commands change your computer's default configuration.

CONFIG.SYS is not a PTARS file. It's a file that DOS uses to establish the working environment. Because PTARS interacts with this environment, you must be sure that certain settings are properly established. Two CONFIG.SYS statements are of immediate importance to PTARS:

BUFFERS	The BUFFERS statement contains the number of disk buffers that DOS sets aside in memory when your computer is started. A disk buffer is a block of memory (typically 512 bytes) that DOS uses to hold data when reading and writing from disk. For best performance with PTARS, the CONFIG.SYS file should contain a BUFFERS statement with a number between 20 and 40 (e.g., BUFFERS=30).
FILES	The FILES statement sets the number of files that DOS can open and access at one time. This number is directly related to the number of files that PTARS will be able to open. The FILES statement in CONFIG.SYS should always be at least 25 (e.g., FILES=25).

See your DOS manual for complete details on the CONFIG.SYS file and the various statements it can contain.

Pack the database(s)

Packing the databases is covered in Chapter 6.

B

File definitions

The files listed below (with their definitions) are installed by Setup into the "\PTARS" hard disk subdirectory. These files are essential to the operation of PTARS. Three of the files, FOXPRO.ESL, FOXPRO.ESO, and PTAR.EXE are in compressed form on the installation disks and will not work if copied directly from the floppy disk to your hard drive. All of the other files installed by PTARS are in normal form on the installation disks.

PTARS files	
CONFIG.FP	resource pointer file
FOXPRO.ESL	database routines library
FOXPRO.ESO	database routines library
CACHE.COM	extended memory (512K req'd) disk cache utility
NPS_MISC.DBF	contains encrypted password, default printer, backup date
NPS_USER.DBF	contains configuration information
NPS_USER.FPT	memo file for configuration information
PTAR.EXE	PTARS executable program
PTARS.COM	PTARS loader program

NPSDC database files	
ACTIVITY.DBF	contains UIC information
CURRICUL.DBF	contains student Curriculum information
DIRECTOR.DBF	contains current Director signature name
MEMBERS.DBF	contains Member information

The following files are created during the operation of PTARS and may or may not be present at any given time:

ACTIVITY.CDX	compound index file for ACTIVITY.DBF
CURRICUL.CDX	compound index file for CURRICUL.DBF
MEMBERS.CDX	compound index file for MEMBERS.DBF
ACT_BU.DBF	hard disk backup of ACTIVITY.DBF
CUR_BU.DBF	hard disk backup of CURRICUL.DBF
DIR_BU.DBF	hard disk backup of DIRECTOR.DBF
MEM_BU.DBF	hard disk backup of MEMBERS.DBF

RECALL1.LST	most recent listing of members receiving recall 1 letter
RECALL2.LST	most recent listing of members receiving recall 2 letter
RECALL3.LST	most recent listing of members receiving recall 3 letter
RECALL4.LST	most recent listing of members receiving recall 4 letter
RECALL1.BAK	previous listing of members receiving recall 1 letter
RECALL2.BAK	previous listing of members receiving recall 2 letter
RECALL3.BAK	previous listing of members receiving recall 3 letter
RECALL4.BAK	previous listing of members receiving recall 4 letter
RELATE1.VUE	PTARS environment file
RELATE2.VUE	PTARS environment file

C

Database specifications

Members.dbf

Field-name	Type	Length	Usage
SSN	Character	11	Social Security Number -- unique, mandatory, key field
LAST_NAME	Character	23	Last Name -- mandatory
FIRST_NAME	Character	15	First Name -- mandatory
MI	Character	1	Middle Initial -- if available
RANK_RATE	Character	5	Rank or Rate -- mandatory
BRANCH	Character	4	Service Branch -- mandatory, popup list
LAST_T2	Date	8	Last-T2-Exam date -- mandatory
CLASS	Numeric	1	Dental Class -- mandatory, range (1 - 4), PTARS updated
PANO	Character	3	Pano X-ray status -- mandatory, popup list
UIC	Character	5	Unit Identification Code -- mandatory, popup list, linked with ACTIVITY.DBF (used in "To:" line of recall letters to students)
CURR_NUM	Character	3	Curriculum Number -- mandatory for UIC 31405, popup list, linked with CURRICUL.DBF
SMC/CODE	Character	4	Student Mail Center number/Department Code -- if available (used in "To:" line of recall letters)
RECALL_1	Date	8	Recall 1 letter date -- PTARS created, editable
RECALL_2	Date	8	Recall 2 letter date -- PTARS created, editable
RECALL_3	Date	8	Recall 3 letter date -- PTARS created, editable
RECALL_4	Date	8	Recall 4 letter date -- PTARS created, editable

Activity.dbf

Field-name	Type	Length	Usage
UIC	Character	5	Unit Identification Code -- unique, mandatory, key field
ACRONYM	Character	11	Acronym for UIC -- mandatory (used in "To:" line of recall letters 1 - 3)
ACT_NAME	Character	47	UIC Name -- mandatory (used in "To:" line of recall 4 letter)
POC	Character	20	UIC Point of Contact -- mandatory (used in "To:" line of recall 4 letter)

Curricul.dbf

Field-name	Type	Length	Usage
CURR_NUM	Character	3	Curriculum Number -- unique, mandatory, key field
CURR_NAME	Character	46	Curriculum Name -- mandatory (used in "To:" line of recall 4 letter applicable to students)
DEPT_CODE	Character	2	Department Code of Curriculum -- mandatory (used in "To:" line of recall 4 letter applicable to students)
PHONE_NO	Character	4	Curriculum Office Phone Number -- mandatory

Director.dbf

Field-name	Type	Length	Usage
DIRECTOR	Character	20	Director signature -- mandatory (format as per signature line of recall letters)

APPENDIX D: RELATION DEFINITIONS

MEMBER

Item	Type	Length
SSN	Character	11
Last-name	Character	23
First-name	Character	15
MI	Character	1
Rank_rate	Character	5
Branch	Character	4
Last_T2	Date	8
Class	Numeric	1
Pano	Character	3
UIC	Character	5
Curr-num	Character	3
SMC/Code	Character	4
Recall_1	Date	8
Recall_2	Date	8
Recall_3	Date	8
Recall_4	Date	8

ACTIVITY

Item	Type	Length
UIC	Character	5
Acronym	Character	11
Act-name	Character	47
POC	Character	20

CURRICULUM

Item	Type	Length
Curr-num	Character	3
Curr-name	Character	46
Dept_code	Character	2
Phone_no	Character	4

APPENDIX E: PROGRAM CODE

```
System: NPSDC PATIENT TRACKING & RECALL SYSTEM
Author: LCDR Timothy P. Steele, MSC, USN
01/28/92   09:01:11
File List

------------------------------------------------------------------------------

Programs and procedures:
   ABOUT               (procedure in C:\PTARS\NPS_PROC.PRG)
   ADISPLAY            (procedure in C:\PTARS\NPS_BROW.PRG)
   BAR1                (procedure in C:\PTARS\NPS_PROC.PRG)
   BAR2                (procedure in C:\PTARS\NPS_PROC.PRG)
   BAR3                (procedure in C:\PTARS\NPS_PROC.PRG)
   BAR4                (procedure in C:\PTARS\NPS_PROC.PRG)
   BDISPLAY            (procedure in C:\PTARS\NPS_BROW.PRG)
   BU                  (procedure in C:\PTARS\NPS_BU.PRG)
   BU_ALL              (procedure in C:\PTARS\NPS_BU.PRG)
   BU_MENU             (procedure in C:\PTARS\NPS_BU.PRG)
   BU_SELECT           (procedure in C:\PTARS\NPS_BU.PRG)
   BU_SINGLE           (procedure in C:\PTARS\NPS_BU.PRG)
   CDISPLAY            (procedure in C:\PTARS\NPS_BROW.PRG)
   CLASSPOP            (procedure in C:\PTARS\NPS_REPO.PRG)
   CONFIG_CHECK        (procedure in C:\PTARS\NPSDC.PRG)
   DATE_TIME           (procedure in C:\PTARS\NPS_UTIL.PRG)
   DBF_BU              (procedure in C:\PTARS\NPS_PROC.PRG)
   DEACTPOP            (procedure in C:\PTARS\NPS_PROC.PRG)
   DELAY               (procedure in C:\PTARS\NPS_INTR.PRG)
   DIRECTOR            (procedure in C:\PTARS\NPS_EDIT.PRG)
   DOCONT              (procedure in C:\PTARS\NPS_PROC.PRG)
   DOGOTO              (procedure in C:\PTARS\NPS_PROC.PRG)
   DOLOCATE            (procedure in C:\PTARS\NPS_PROC.PRG)
   DOPACK              (procedure in C:\PTARS\NPS_PACK.PRG)
   DO REST_FLOP        (procedure in C:\PTARS\NPS_BU.PRG)
   DO REST_HARD        (procedure in C:\PTARS\NPS_BU.PRG)
   EDIT                (procedure in C:\PTARS\NPS_RECA.PRG)
   ENV_SETUP           (procedure in C:\PTARS\NPSDC.PRG)
   FLOPPY_SIZE         (procedure in C:\PTARS\NPS_BU.PRG)
   FMT                 (procedure in C:\PTARS\NPS_BU.PRG)
   FUNC_LIST           (procedure in C:\PTARS\NPS_PROC.PRG)
   GETKEY              (procedure in C:\PTARS\NPS_PROC.PRG)
   GORECALLS           (procedure in C:\PTARS\NPS_RECA.PRG)
   GOTOREC             (procedure in C:\PTARS\NPS_PROC.PRG)
   HLP                 (procedure in C:\PTARS\NPS_PROC.PRG)
   LISTPOP             (procedure in C:\PTARS\NPS_REPO.PRG)
   LIST_FILES          (procedure in C:\PTARS\NPS_BU.PRG)
   LIST_SELECT         (procedure in C:\PTARS\NPS_BU.PRG)
   MAIN_MENU           (procedure in C:\PTARS\NPSDC.PRG)
   NPS_AGET            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_AREA            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_AREP            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_ASAY            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_ASTO            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_BGET            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_BREP            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_BSAY            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_BSTO            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_CGET            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_CHECK           (procedure in C:\PTARS\NPS_APPE.PRG)
   NPS_CPAS            (procedure in C:\PTARS\NPS_UTIL.PRG)
   NPS_CREC            (procedure in C:\PTARS\NPS_RECA.PRG)
   NPS_CREP            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_CSAY            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_CSTO            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_FILE            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_FORMA1          (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_FORMA2          (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_FORMB           (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_FORMC           (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_GETS            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_KEYS            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_NDIS            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_REPL            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_SAY1            (procedure in C:\PTARS\NPS_APPE.PRG)
   NPS_SAY2            (procedure in C:\PTARS\NPS_APPE.PRG)
   NPS_SAY3            (procedure in C:\PTARS\NPS_APPE.PRG)
   NPS_SAYS            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_SEEK            (procedure in C:\PTARS\NPS_PROC.PRG)
   NPS_STOR            (procedure in C:\PTARS\NPS_PROC.PRG)
   PACK                (procedure in C:\PTARS\NPS_PACK.PRG)
   PACK_CHECK          (procedure in C:\PTARS\NPSDC.PRG)
   PACK_POP            (procedure in C:\PTARS\NPS_UTIL.PRG)
   PAINT               (procedure in C:\PTARS\NPS_BROW.PRG)
   PANPOP              (procedure in C:\PTARS\NPS_REPO.PRG)
   PASSTRAN            (procedure in C:\PTARS\NPS_PROC.PRG)
   POPCAL              (procedure in C:\PTARS\NPS_PROC.PRG)
   POPTRIS.PRG
   PRNCHR              (procedure in C:\PTARS\NPS_PROC.PRG)
   PRNDEFAULT          (procedure in C:\PTARS\NPS_UTIL.PRG)
   PRNPROB             (procedure in C:\PTARS\NPS_PROC.PRG)
   PRNRPT              (procedure in C:\PTARS\NPS_REPO.PRG)
   PRNSTOP             (procedure in C:\PTARS\NPS_PROC.PRG)
   PROMPTBU            (procedure in C:\PTARS\NPSDC.PRG)
   QUITBU              (procedure in C:\PTARS\NPSDC.PRG)
   RECALLS_MENU        (procedure in C:\PTARS\NPS_RECA.PRG)
   RECDISPLAY          (procedure in C:\PTARS\NPS_RECA.PRG)
   RECPRN              (procedure in C:\PTARS\NPS_RECA.PRG)
   REC_LISTS           (procedure in C:\PTARS\NPS_RECA.PRG)
   REPORT_MENU         (procedure in C:\PTARS\NPS_REPO.PRG)
   RESTOR  .           (procedure in C:\PTARS\NPS_BU.PRG)
   REST_POP            (procedure in C:\PTARS\NPS_BU.PRG)
   REST_SELECT         (procedure in C:\PTARS\NPS_BU.PRG)
   SAYEOF              (procedure in C:\PTARS\NPS_PROC.PRG)
   SAYLINE             (procedure in C:\PTARS\NPS_PROC.PRG)
   SAYREC              (procedure in C:\PTARS\NPS_PROC.PRG)
   SETPRN              (procedure in C:\PTARS\NPS_PROC.PRG)
   SET_POP             (procedure in C:\PTARS\NPSDC.PRG)
   SPACE_CHECK         (procedure in C:\PTARS\NPS_OPEN.PRG)
   STATLINE            (procedure in C:\PTARS\NPS_PROC.PRG)
   UPDATE              (procedure in C:\PTARS\NPS_OPEN.PRG)
   UPDATEREC           (procedure in C:\PTARS\NPS_EDIT.PRG)
   UTILITY_MENU        (procedure in C:\PTARS\NPS_UTIL.PRG)
   VAL_BRAN            (procedure in C:\PTARS\NPS_PROC.PRG)
   VAL_CURR            (procedure in C:\PTARS\NPS_PROC.PRG)
   VAL_PANO            (procedure in C:\PTARS\NPS_PROC.PRG)
   VAL_REC1            (procedure in C:\PTARS\NPS_RECA.PRG)
   VAL_REC2            (procedure in C:\PTARS\NPS_RECA.PRG)
   VAL_REC3            (procedure in C:\PTARS\NPS_RECA.PRG)
   VAL_REC4            (procedure in C:\PTARS\NPS_RECA.PRG)
   VAL_UIC             (procedure in C:\PTARS\NPS_PROC.PRG)
   VIEWBROW            (procedure in C:\PTARS\NPS_PROC.PRG)
   WARNING             (procedure in C:\PTARS\NPS_UTIL.PRG)
   WARN_DBF            (procedure in C:\PTARS\NPS_OPEN.PRG)
   WARN_WINDOW         (procedure in C:\PTARS\NPS_RECA.PRG)

Procedure files:
   NPS_PROC.PRG
   NPSDC.PRG
   NPS_INTR.PRG
   NPS_OPEN.PRG
   NPS_APPE.PRG
   NPS_EDIT.PRG
   NPS_BROW.PRG
   NPS_RECA.PRG
   NPS_REPO.PRG
   NPS_UTIL.PRG
   NPS_BU.PRG
   NPS_PACK.PRG

Databases:
   &BAKFILE.DBF
   &BU.DBF
   &BUFILE
   &BUFILE.DBF
   &ORIGFILE
   &ORIGFILE.DBF
   &ORIGINDEX.DBF
   &ORIGNDX.DBF
   &PACK_FIL
   &TEMPFILE
   &TEMPFILE.DBF
   ACTIVITY.DBF
   CURRICUL.DBF
   DIRECTOR.DBF
   MEMBERS.DBF
   NPS_MISC.DBF

Multiple index files:
   ACTIVITY.CDX
   CURRICUL.CDX
   MEMBERS.CDX

Report forms:
   &RPTNAME
   CLAS_CTS.FRX
   RECALL&CNT.FRX

Other files:
   &RECTYPE
   NPS_HLP
   NPS_USCR

                    FILELIST.DOC    1 of 1
```

```
System: NPSDC PATIENT TRACKING & RECALL SYSTEM
Author: LCDR Timothy P. Steele, MSC, USN
01/28/92   09:00:53
Token Cross-Reference Report

-------------------------------------------------------------------------------

401 tokens are included in this report.

Legend for context symbols:
   (blank) reference does not change the variable or field value.
   * variable or field is changed in an assignment statement.
   ! PROCEDURE or FUNCTION statement.
   x variable is released.
   A array is declared.
   G GET or MENU statement changes variable or field.
   P variable is declared PUBLIC.
   R field is replaced.
   U database is USEd
   V variable is declared PRIVATE.
   & variable is referenced in a macro--takes precedence over all others.
   ? reference is of unknown type.

File types appear next to tokens that are used as file names.  Some of these
tokens may also be used in other ways.

ABOUT                    (procedure in NPS_PROC.PRG)
   NPSDC.PRG       212
   NPS_PROC.PRG    179!  183   186   235   237x  314

ACRONYM
   NPS_PROC.PRG   1350R
   NPS_BROW.PRG    322

ACTIVITY                 (database)
   NPS_PROC.PRG   1453
   NPS_OPEN.PRG    110U  170U  176
   NPS_REPO.PRG     36

ACTIVITY.ACRONYM
   NPS_PROC.PRG   1069G 1146

ACTIVITY.ACT_NAME
   NPSDC.PRG       275
   NPS_PROC.PRG   1068G 1145

ACTIVITY.POC
   NPS_PROC.PRG   1070G 1147

ACTIVITY.UIC
   NPSDC.PRG       275
   NPS_PROC.PRG   1067G 1144
   NPS_BROW.PRG    167

ACT_NAME
   NPS_PROC.PRG   1349R
   NPS_BROW.PRG    322

ACT_UIC
   NPS_OPEN.PRG    114

ADISPLAY                 (procedure in NPS_BROW.PRG)
   NPS_BROW.PRG    251!  378

ALL_NAM
   NPS_REPO.PRG     37

ALL_SSN
   NPS_REPO.PRG     38

ALT
   NPSDC.PRG       213

BACKUP_UTILS
   NPS_PROC.PRG    158

BAKFILE
   NPS_RECA.PRG    282*  285&  287&

BAR1                     (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    259   274!

BAR2                     (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    260   287!

BAR3                     (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    261   300!

BAR4                     (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    262   313!

BDISPLAY                 (procedure in NPS_BROW.PRG)
   NPS_BROW.PRG    297!  380

BRANCH
   NPSDC.PRG       255   258   259   260   261   262   263   264
   NPS_PROC.PRG   1126  1326  1414
   NPS_BROW.PRG    283

BRANID
   NPS_PROC.PRG   1409

BU                       (procedure in NPS_BU.PRG)
   NPSDC.PRG        56    61   100   400*
   NPS_BU.PRG       98   142   145   173!  176   179   180   181   182   183
                   184   187   310x  348   379   382   389&  393   406

BUF
   NPSDC.PRG       444V  453*  454   458*  459   463*  464   467*  468
   NPS_OPEN.PRG     43V
   NPS_RECA.PRG     34V   38*   72
   NPS_BU.PRG      174V  198*  199   203*  204   208*  209   212*  213   268*
                   269   271*  272   280*  281   283*  284   292*  293   295*
                   296   303*  304   306*  307   327V  336*  337   341*  342
                   346*  347   350*  351   660V  682*  683   686*  687   690*
                   691   694*  695   699*  703   706*  707   710*  711   714*
                   715   722*  723   725*  729   734*  737   739*  743   750*
                   751   753*  757   764*  765   767*  771

BUFILE
   NPS_PROC.PRG    401   413   415&  418   421&  426   428&  430   439&  443&
                   448
   NPS_BU.PRG       34V  793   802   804&  809   839   856   865   867&  872
                   902

BU_ALL                   (procedure in NPS_BU.PRG)
   NPS_BU.PRG      241   253   326!

BU_DISK
   NPS_BU.PRG      128V  141*  144*  193   214   267   270   279   282   291
                   294   302   305

BU_MENU                  (procedure in NPS_BU.PRG)
   NPS_BU.PRG       38   791

BU_SELECT                (procedure in NPS_BU.PRG)
   NPS_BU.PRG       54   127!  129   132   133   134   137   147x

BU_SINGLE                (procedure in NPS_BU.PRG)
   NPS_BU.PRG      272   284   296   307   367!

CAL
   NPS_PROC.PRG     73    76    77    96x

CALENDAR
   NPS_PROC.PRG     77    78

CDISPLAY                 (procedure in NPS_BROW.PRG)
   NPS_BROW.PRG    335!  382

CHOICE
   NPSDC.PRG        58    63   109*  110   111   296   352*  353   355*  356G
                   361   362*  362   397*  398   399
   NPS_PROC.PRG    180V  200*  201   202   215   216   229   230   500   502*
                   503   506*  615   618   621   624   627   670*  671   675
                   677  1602* 1603  1604
   NPS_INTR.PRG     20V  134*  135   136   147*  148   149
   NPS_OPEN.PRG     52
   NPS_APPE.PRG     55*   59   121   123   127   132   150   155
   NPS_BROW.PRG     88    90    92   114   118   131   133   134   148   154
                   158   204   214   224   227   232   237
   NPS_RECA.PRG     48    49    51    51    62    62   117   117   121   121
                   382*  383   385   387   409   413   421   429   447   449
                   450   464   470   480
   NPS_REPO.PRG     53    54    56    56    65    65    94    95   100   100
                   109   109   119   119   128   128   138   138   146   146
   NPS_UTIL.PRG     32    33    35    35    39    43    43    49    49    56
                    56    63    63    67    67   216   281   282
   NPS_BU.PRG       40    41    43    43    43    47    51    51    56    56
                    60    60    64    64   228   229   383   384   588*  607
                   609
   NPS_PACK.PRG     39    40

CLASN
   NPS_OPEN.PRG     86

CLASS
   NPS_PROC.PRG   1128  1328
   NPS_OPEN.PRG     85    85    86   226R
   NPS_BROW.PRG    283
   NPS_REPO.PRG     73    74    75    76   282   284   286   288   290   290
                   292   292

CLASSPOP                 (procedure in NPS_REPO.PRG)
   NPS_REPO.PRG    117   264!  266   269   270   271   272   273   274   275
                   276   279   296x

CLAS_CTP
   NPS_REPO.PRG     39

CLAS_CTS
   NPS_REPO.PRG     91

CLAS_NAM
   NPS_REPO.PRG     40

CLAS_SSN
   NPS_REPO.PRG     41

COLGET
   NPS_INTR.PRG     20V   20V   93*   98
   NPS_UTIL.PRG    299V  324*  330   354

COLSAY
   NPS_INTR.PRG     94*  100
   NPS_UTIL.PRG    299V  325*  341   363

COL_CNTR
   NPS_INTR.PRG     19V

CONFIG_CHECK             (procedure in NPSDC.PRG)
   NPSDC.PRG       219   228!

COUNTER
   NPS_INTR.PRG     19V   25*   26    28    30    32    34    36    38    40
                    55*   55    87*  108*  108   110   111   113   114
   NPS_EDIT.PRG     32V
   NPS_RECA.PRG     34V  214*  217   218*  218   219

CURRICUL                 (database)
   NPS_PROC.PRG   1491
   NPS_OPEN.PRG    126U  172U  177
   NPS_REPO.PRG     42

CURRICUL.CURR_NAME
   NPSDC.PRG       281
   NPS_PROC.PRG   1084G 1160

CURRICUL.CURR_NUM
   NPSDC.PRG       281
   NPS_PROC.PRG   1083G 1159  1367R
   NPS_BROW.PRG    184

CURRICUL.DEPT_CODE
   NPS_PROC.PRG   1085G 1161

CURRICUL.PHONE_NO
   NPS_PROC.PRG   1086G 1162

CURRNUM
   NPS_PROC.PRG   1488 1492

CURR_NAME
   NPS_PROC.PRG   1348R
   NPS_BROW.PRG    353

CURR_NUM
   NPS_PROC.PRG   1131  1331
   NPS_OPEN.PRG     89   130   159   177
   NPS_EDIT.PRG    178R
   NPS_BROW.PRG    284   353

CURR_NUMC
   NPS_OPEN.PRG    130

CURR_NUMS
   NPS_OPEN.PRG     89

DATE_TIME                (procedure in NPS_UTIL.PRG)
   NPS_UTIL.PRG     66   161!  163   166   167   168   171   180x

DBFAREA
   NPSDC.PRG        57    62    83*  305   308   311   314
   NPS_PROC.PRG    695&  697   700   703   706   734   746*  763   776   779
```

```
                   801   826   835   840   848   877   1008  1010  1012  1103
                   1106  1107  1178  1180  1182  1300  1302  1304
   NPS_APPE.PRG     42    44    46    48    68    70    72    97   100   103
                   241   256   264
   NPS_EDIT.PRG     42    44    46    48   122   124   126   151   155   159
                   175
   NPS_BROW.PRG     29    57    59    61   164   166   183   377   379   381
   NPS_RECA.PRG    60&   339*
   NPS_REPO.PRG    63&
   NPS_UTIL.PRG    55&   150&  228&  237&  246&  255&
   NPS_BU.PRG      55&   777&

DBFCHOICE
   NPS_PROC.PRG    734*  737*  739*  741*  743*  746

DBFILE
   NPS_PROC.PRG    723   726   727   728   729   730   733   745x

DBFNAME
   NPSDC.PRG        57    62
   NPS_PROC.PRG    523   523   698*  701*  704*  707*
   NPS_BROW.PRG    375

DBFPAGEMAX
   NPSDC.PRG        57    62

DBF_BU                   (procedure in NPS_PROC.PRG)
   NPSDC.PRG       454   459   464   468
   NPS_PROC.PRG    399!
   NPS_RECA.PRG     72
   NPS_BU.PRG      199   204   209   213   269   281   293   304   337   342
                   347   351   393   406

DBF_TOP
   NPS_OPEN.PRG    196V  199*  200

DEACTPOP                 (procedure in NPS_PROC.PRG)
   NPSDC.PRG       264   272   278   284
   NPS_PROC.PRG    343   730  1396!
   NPS_RECA.PRG    187
   NPS_REPO.PRG    217   276   317
   NPS_UTIL.PRG    131   168   203
   NPS_BU.PRG      134   184   457   488   550   630   669

DELAY                    (procedure in NPS_INTR.PRG)
   NPS_INTR.PRG     54   168!

DELRECORD
   NPSDC.PRG        56    61
   NPS_OPEN.PRG     49*
   NPS_APPE.PRG    121   123
   NPS_BROW.PRG     42   158

DELRECS
   NPS_PACK.PRG     81V   89    91

DEPT_CODE
   NPS_PROC.PRG   1369R
   NPS_BROW.PRG    353

DOCONT                   (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    652   667!

DOGOTO                   (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    607!
   NPS_EDIT.PRG     92
   NPS_BROW.PRG    149
   NPS_RECA.PRG    465

DOLOCATE                 (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    640!

DONE
   NPS_PROC.PRG   1560*

DOPACK                   (procedure in NPS_PACK.PRG)
   NPS_PACK.PRG     97   105   119!

DOWNARROW
   NPS_BROW.PRG     25V   38*   41    90
   NPS_RECA.PRG    333V  346*  347   385

DO_REST_FLOP             (procedure in NPS_BU.PRG)
   NPS_BU.PRG      703   707   711   715   729   743   757   771   855!

DO_REST_HARD             (procedure in NPS_BU.PRG)
   NPS_BU.PRG      683   687   691   695   723   737   751   765   792!

DRIVE
   NPS_BU.PRG      424V  426*  435&  438&

DSPACE
   NPS_BU.PRG      369V  423   432*  433

EDITCHOICE
   NPS_EDIT.PRG     32V   52    59    65    67    75    90    95   105   115

EDIT_RECALLS
   NPS_PROC.PRG    140

ENDKEY
   NPS_BROW.PRG     25V   36*   42   237
   NPS_RECA.PRG    333V

ENV_SETUP                (procedure in NPSDC.PRG)
   NPSDC.PRG        34   172!

ESC
   NPS_PROC.PRG   1410  1420  1431  1440

EXPR
   NPSDC.PRG        59    64
   NPS_PROC.PRG    641   645&  650&  795V  808*  809   810   821*  822   823
                   831*  832   833   840*  841   842   855   856*  866*  875*
                   884*
   NPS_OPEN.PRG     52
   NPS_APPE.PRG     37*   85    90
   NPS_EDIT.PRG     35*  133   134   142

F1
   NPSDC.PRG       213   354
   NPS_APPE.PRG     58
   NPS_EDIT.PRG     56   226
   NPS_BROW.PRG     75
   NPS_RECA.PRG     46   363   544
   NPS_REPO.PRG     51
   NPS_UTIL.PRG     28   315
   NPS_BU.PRG       36

F2
   NPSDC.PRG       210

F3
   NPSDC.PRG       211

F4
   NPSDC.PRG       212

FILESIZE
   NPS_BU.PRG      369V  423

FILE_SIZE
   NPS_OPEN.PRG    196V  200*  201   205
   NPS_BU.PRG      424V  428*  430*  430   433

FIRST_NAME
   NPS_PROC.PRG   1123  1323
   NPS_OPEN.PRG     84    86    88    91
   NPS_BROW.PRG    281
   NPS_RECA.PRG    523   549

FLOP
   NPS_BU.PRG      327V  328*  337   342   347   351   369V  371*  393   406

FLOPPY
   NPS_PROC.PRG    401   414   442

FLOPPY_SIZE              (procedure in NPS_BU.PRG)
   NPS_BU.PRG      239   251   391   404   422!

FMT                      (procedure in NPS_BU.PRG)
   NPS_BU.PRG       63   543!  544   547   548   549   550   553   574x

FUNCS
   NPS_PROC.PRG    252   255   256   257   258   259   260   261   262   263
                   276   289   302   315

FUNCTION
   NPSDC.PRG       202   203   204   205   206   207   208   209
   NPS_PROC.PRG   1203  1206  1207  1212  1217  1219  1407  1428  1448  1486
   NPS_RECA.PRG    598   618   638   657

FUNCTIONS
   NPS_PROC.PRG    143

FUNC_LIST                (procedure in NPS_PROC.PRG)
   NPSDC.PRG       213
   NPS_PROC.PRG    251!

GETKEY                   (procedure in NPS_PROC.PRG)
   NPSDC.PRG       110   398
   NPS_PROC.PRG    201   215   229   499!  615   675   769  1603
   NPS_INTR.PRG    135   148
   NPS_APPE.PRG    121
   NPS_EDIT.PRG     65
   NPS_BROW.PRG     88   114
   NPS_RECA.PRG     48   383   409
   NPS_REPO.PRG     53    94
   NPS_UTIL.PRG     32   281
   NPS_BU.PRG       40   228   383   607
   NPS_PACK.PRG     39

GORECALLS                (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG     90    96   102   107   279!

GOTOREC                  (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    579!  628

HDG
   NPS_INTR.PRG     22*   23    62
   NPS_UTIL.PRG     82V   87*   88    92
   NPS_BU.PRG       80V   83*   84    91

HLP                      (procedure in NPS_PROC.PRG)
   NPSDC.PRG       354
   NPS_PROC.PRG    119!  275
   NPS_APPE.PRG     58
   NPS_EDIT.PRG     56   226
   NPS_BROW.PRG     75
   NPS_RECA.PRG     46   363   544
   NPS_REPO.PRG     51
   NPS_UTIL.PRG     28   315
   NPS_BU.PRG       36

HLPTOPIC
   NPSDC.PRG        46    49*
   NPS_PROC.PRG    120   124   127   130   133   136   139   142   145   148
                   151   154   157   160   163

HOME
   NPS_BROW.PRG     25V   35*   41   224
   NPS_RECA.PRG    333V

I
   NPS_PROC.PRG     72V   80*   82    84    84    86    88    90    92  1537
                  1538  1565  1566
   NPS_INTR.PRG     97    98   100
   NPS_UTIL.PRG    328   330   341   352   354   363

ISBLANK
   NPS_PROC.PRG    855   857*  865*  874*  883*
   NPS_APPE.PRG     30V   34    85    86   110
   NPS_EDIT.PRG     33V   36   133   137   168

ISDELETED
   NPS_PROC.PRG    519   524
   NPS_APPE.PRG     30V   34    78   111*  115   125*  125   126   129*  134
                   137   236*  247*  252*  254   262*  269*
   NPS_EDIT.PRG     33V   36   169*

ISEDITED
   NPSDC.PRG        57    62   133*  134
   NPS_APPE.PRG     30V   35*  130*  145*  230
   NPS_EDIT.PRG     31    36    87*   94*  103*  113*  117*
   NPS_BROW.PRG     27V   53*
   NPS_RECA.PRG    335V  358   369   372*  422   435   436   444*  541   545*
                   580*

ISFILEOK
   NPS_PROC.PRG    402V  425*  431*  434
   NPS_BU.PRG      794V  817*  822*  825   857V  880*  885*  888
   NPS_PACK.PRG     21V   50*   65    80   132*  137*  140*  144

ISUNIQUE
   NPS_PROC.PRG    855   858*  867*  876*  885*
   NPS_APPE.PRG     30V   34    85    86
   NPS_EDIT.PRG     33V   36   133   137

ISVALID
   NPSDC.PRG        59    64

KBYTES
   NPS_OPEN.PRG    196V  205*  207   209
   NPS_PACK.PRG     21V   58*   59
```

```
KEYCHARS
   NPS_PROC.PRG    500  503

KEYCODE
   NPS_PROC.PRG    501V  504*  505   506
   NPS_INTR.PRG     20V

KEYNOM
   NPS_EDIT.PRG     33V  123*  125*  127*  134

KEYSTROKES
   NPS_PROC.PRG    180V  187*  201   215   229
   NPS_BROW.PRG     26V   41*   88   114
   NPS_RECA.PRG    334V  347*  383   409

LASTPAGE
   NPS_EDIT.PRG     32V

LASTREC
   NPSDC.PRG        59    64   122
   NPS_PROC.PRG    710*
   NPS_OPEN.PRG     51
   NPS_APPE.PRG     78   115   126   144*  164
   NPS_EDIT.PRG     92
   NPS_BROW.PRG    149
   NPS_RECA.PRG    465
   NPS_UTIL.PRG     39
   NPS_BU.PRG       47

LASTRECNUM
   NPS_PROC.PRG    580  585   588   608  619   628

LAST_NAME
   NPS_PROC.PRG   1122  1322
   NPS_OPEN.PRG     84    86    88    91
   NPS_BROW.PRG    281
   NPS_RECA.PRG    522   548

LAST_T2
   NPS_PROC.PRG   1127  1327
   NPS_OPEN.PRG    225
   NPS_EDIT.PRG    195
   NPS_BROW.PRG    283
   NPS_RECA.PRG     85    92    98   104

LEFTARROW
   NPS_BROW.PRG     25V   39*   41   227
   NPS_RECA.PRG    333V

LISTPOP                (procedure in NPS_REPO.PRG)
   NPS_REPO.PRG    107   116   126   135   2111  212   215   216   217   220
                   253x

LISTRECS
   NPS_BROW.PRG    253  254   281   283   286   299  300   322   324   337
                   338   353
   NPS_RECA.PRG    502  503   522

LISTTYPE
   NPS_REPO.PRG     34*  106*  115*  125*  134*  224   227   230   233   239
                   242   245   248

LIST_DISK
   NPS_BU.PRG      451V  464*  467*  496   502   514   520

LIST_FILES             (procedure in NPS_BU.PRG)
   NPS_BU.PRG      465   468   4821  483   486   487   488   491   531x

LIST_SELECT            (procedure in NPS_BU.PRG)
   NPS_BU PRG       59   4501  452   455   456   457   460   470x

LSTFILE
   NPS_RECA.PRG    281*  287G  289G

MACRONYM
   NPSDC PRG        71    76
   NPS_PROC.PRG   1146  1264G 1350

MACT_NAME
   NPSDC PRG        71    76
   NPS_PROC.PRG   1145  1261G 1262  1349
   NPS_APPE.PRG    257

MAIN
   NPS_PROC.PRG    125

MAINCHOICE
   NPSDC.PRG        58    63    85*   89    92   122   126   130   135   139
```

```
                   143   147   150

MAIN_MENU              (procedure in NPSDC.PRG)
   NPSDC.PRG        89   2951

MBRANCH
   NPSDC.PRG        68    75
   NPS_PROC.PRG   1126* 1211G 1211G 1326  1411  1411  1412  1412  1412  1415*
                  1416G

MBU
   NPSDC.PRG        47    54*
   NPS_BU.PRG      775   795*  858*

MCLASS
   NPSDC.PRG        68    75
   NPS_PROC.PRG   1128* 1216G 1216G 1328
   NPS_APPE.PRG    243

MCNTPANO1
   NPS_REPO.PRG     27V   31    78    81    87

MCNTPANO2
   NPS_REPO.PRG     27V   31    79    81    88

MCNTPANO3
   NPS_REPO.PRG     27V   31    80    81    89

MCOUNT
   NPSDC.PRG        46    50*  426   427

MCOUNT1
   NPS_REPO.PRG     26V   30    73    77    82    86

MCOUNT2
   NPS_REPO.PRG     26V   30    74    77    83    86

MCOUNT3
   NPS_REPO.PRG     26V   30    75    77    84

MCOUNT4
   NPS_REPO.PRG     24V   30    76    77    85

MCURR_NAME
   NPSDC.PRG        72    77
   NPS_PROC.PRG   1160  1279G 1280  1368
   NPS_APPE.PRG    265

MCURR_NUM
   NPSDC.PRG        69    75
   NPS_PROC.PRG   1131* 1240G 1240G 1331  1511* 1512G
   NPS_APPE.PRG    249
   NPS_BROW.PRG    184*  187

MCURR_NUMC
   NPSDC.PRG        72    77
   NPS_PROC.PRG    837*  838G  840   879G  880   883   884  1159  1367
   NPS_APPE.PRG    265
   NPS_EDIT.PRG    127

MDEPT_CODE
   NPSDC.PRG        72    77
   NPS_PROC.PRG   1161  1282G 1369

MDIRECTOR
   NPS_EDIT.PRG    212V  215*  227G  228   230

MEMBERS                (database)
   NPSDC.PRG       425U
   NPS_PROC.PRG   1456  1469  1494  1507
   NPS_OPEN.PRG     71U  157   159   169U  171   173
   NPS_BU.PRG      195U  263U  333U

MEMBERS.BRANCH
   NPS_PROC.PRG   1031G

MEMBERS.CLASS
   NPS_PROC.PRG   1033G

MEMBERS.CURR_NUM
   NPS_PROC.PRG   1043G

MEMBERS.FIRST_NAME
   NPS_PROC.PRG   1028G

MEMBERS.LAST_NAME
   NPS_PROC.PRG   1027G
```

```
MEMBERS.LAST_T2
   NPS_PROC.PRG   1032G

MEMBERS.MI
   NPS_PROC.PRG   1029G

MEMBERS.PANO
   NPS_PROC.PRG   1034G

MEMBERS.RANK_RATE
   NPS_PROC.PRG   1030G

MEMBERS.RECALL_1
   NPS_PROC.PRG   1051G
   NPS_RECA.PRG     99

MEMBERS.RECALL_2
   NPS_PROC.PRG   1052G
   NPS_RECA.PRG     93

MEMBERS.RECALL_3
   NPS_PROC.PRG   1053G

MEMBERS.RECALL_4
   NPS_PROC.PRG   1054G

MEMBERS.SMC
   NPS_PROC.PRG   1044G 1048G

MEMBERS.SSN
   NPS_PROC.PRG   1026G

MEMBERS.UIC
   NPS_PROC.PRG   1035G 1036

MENX
   NPS_INTR.PRG     20V   95*   98   101
   NPS_UTIL.PRG    299V  329*  330   331   342   353*  354   364

MENUCHOICE
   NPSDC.PRG        58    63
   NPS_OPEN.PRG     51

MENUHDG
   NPSDC.PRG       319*  320   324
   NPS_RECA.PRG    141*  142   146
   NPS_REPO.PRG    165V  170*  171   175
   NPS_UTIL.PRG    306*  307   311

MFIRST_NAM
   NPSDC.PRG        67    74
   NPS_PROC.PRG    814*  817G  821  1123* 1204G 1323

MFIRST_NAME
   NPS_PROC.PRG   1206

MI
   NPS_PROC.PRG   1124  1324
   NPS_BROW.PRG    281

MISSING_DBF
   NPS_OPEN.PRG     43V   58*   60    65*   67   104*  106   120*  122   135*
                   137   238  245   246   243

MLAST_NAME
   NPSDC.PRG        67    74
   NPS_PROC.PRG    813*  815G  821  1122* 1201G 1202  1322

MLAST_T2
   NPSDC.PRG        68    79
   NPS_PROC.PRG   1127* 1213G 1213G 1214  1214  1327
   NPS_APPE.PRG    243

MMI
   NPSDC.PRG        67    74
   NPS_PROC.PRG   1124* 1207G 1324

MNEWPASS
   NPS_UTIL.PRG    299V  313*  326*  342*  342   345   345   371   372

MODE
   NPSDC.PRG        46    51
   NPS_PROC.PRG    336   339   340   341   342   343   345   380x

MPANO
   NPSDC.PRG        68    75
   NPS_PROC.PRG   1129* 1218G 1218G 1329  1432  1432  1432  1435* 1436G
```

```
MPASSWRD
   NPS_INTR.PRG     19V   89*   96*  101*  101   104

MPCT_CNT1
   NPS_REPO.PRG     20V   32    82*

MPCT_CNT2
   NPS_REPO.PRG     20V   32    83*

MPCT_CNT3
   NPS_REPO.PRG     20V   32    84*

MPCT_CNT4
   NPS_REPO.PRG     20V   32    85*

MPCT_OR
   NPS_REPO.PRG     20V   32    86*

MPCT_PANO1
   NPS_REPO.PRG     29V   33    87*

MPCT_PANO2
   NPS_REPO.PRG     29V   33    88*

MPCT_PANO3
   NPS_REPO.PRG     29V   33    89*

MPHONE_NO
   NPSDC.PRG        72    77
   NPS_PROC.PRG   1162  1283G 1370

MPOC
   NPSDC.PRG        71    76
   NPS_PROC.PRG   1147  1265G 1264  1351
   NPS_APPE.PRG    258

MPRINT
   NPSDC.PRG        46
   NPS_PROC.PRG   1617  1618* 1635*
   NPS_RECA.PRG     80    83   191*  217   247   250   253   280   297   301
   NPS_REPO.PRG    357   360

MPRINTCNT
   NPSDC.PRG        47    53
   NPS_RECA.PRG     80    83   108*  247   250   258*
   NPS_REPO.PRG    357   360   365*

MPROMPT
   NPSDC.PRG        46
   NPS_PROC.PRG    344*  347   352   357   368   731*  734   738   740   742
                  1397* 1413* 1415  1433* 1435  1464* 1466  1473  1502* 1504
                  1511
   NPS_RECA.PRG    188*  193   198   203   208   213
   NPS_REPO.PRG    218*  222   237   277*  281   283   285   287   289   291
                   293   318*  322   324   326   328
   NPS_UTIL.PRG    132*  138   139   140   141   146   169*  175   177   204*
                   211   229   238   247
   NPS_BU.PRG      135*  140   143   185*  191   261   274   284   298   458*
                   463   466   489*  494   512   551*  555   561   567   631*
                   637   640   670*  677   717   731   745   759

MRANK_RATE
   NPSDC.PRG        67    74
   NPS_PROC.PRG   1125* 1208G 1209  1325

MRECALL_1
   NPSDC.PRG        69    79
   NPS_RECA.PRG    557*  565G  565G  584   601   604   604   605   605   625
                   625   644   660

MRECALL_2
   NPSDC.PRG        69    79
   NPS_RECA.PRG    558*  567G  567G  585   601   605   606   621   624   624
                   625   626   644   644   660

MRECALL_3
   NPSDC.PRG        70    79
   NPS_RECA.PRG    559*  569G  569G  586   602   621   626   626   641   643
                   643   644   645   660   661

MRECALL_4
   NPSDC.PRG        70    79
   NPS_RECA.PRG    560*  571G  571G  587   602   622   641   645   645   659
                   659   660

MSMC
   NPSDC.PRG        69    75
   NPS_PROC.PRG   1132* 1231G 1246G 1332

MSSN
   NPSDC.PRG        67    74
   NPS_PROC.PRG    805*  806G  808   862G  863   865   866  1121* 1321
   NPS_APPE.PRG    232   242
   NPS_EDIT.PRG    123

MTOTCOUNT
   NPS_REPO.PRG     26V   30    77*   82    83    84    85    86

MTOTPANO
   NPS_REPO.PRG     27V   31    81*   87    88    89

MUIC
   NPSDC.PRG        68    75
   NPS_PROC.PRG   1130* 1220G 1220G 1227  1233  1330  1473* 1474G
   NPS_APPE.PRG    242   249
   NPS_BROW.PRG    167*  170

MUICB
   NPSDC.PRG        71    76
   NPS_PROC.PRG    828*  829G  831   870G  871   874   875  1144  1348
   NPS_APPE.PRG    257
   NPS_EDIT.PRG    125

MVERIFY
   NPS_UTIL.PRG    299V  314*  351*  364*  364   371

NAME
   NPS_OPEN.PRG     84

NDICHOICE
   NPS_PROC.PRG    761   769   771   774   778*  781*
   NPS_EDIT.PRG     32V   52    77    78

NDIORDER
   NPSDC.PRG        57    62
   NPS_PROC.PRG    699*  702*  705*  774*  775G  796   804   812
   NPS_BROW.PRG    172G  180G  189G  197G

NFILES
   NPSDC.PRG       230*  231   231   237

NORELATE1
   NPSDC.PRG        47    53
   NPS_OPEN.PRG    155   163*

NORELATE2
   NPSDC.PRG        47    53
   NPS_OPEN.PRG    167   180*

NO_DBF
   NPSDC.PRG        46    48    94   122
   NPS_OPEN.PRG     59*   66*  105*  121*  136*  186*
   NPS_UTIL.PRG     39
   NPS_BU.PRG       47

NO_SIZE
   NPS_BU.PRG      174V  218*  239   240   251   252   369V  370*  391   392
                   404   405   423   434*

NPS_AGET              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1179  1198!

NPS_APPE              (procedure file)
   NPSDC.PRG       129

NPS_AREA              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    694!  747
   NPS_OPEN.PRG    185
   NPS_RECA.PRG    340

NPS_AREP              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1301  1315!

NPS_ASAY              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1009  1023!

NPS_ASTO              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1104  1118!

NPS_BGET              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1181  1258!

NPS_BREP              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1303  1342!

NPS_BROW              (procedure file)
   NPSDC.PRG       138

NPS_BSAY              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1011  1064!

NPS_BSTO              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1106  1141!

NPS_BU                (procedure file)
   NPSDC.PRG       402
   NPS_UTIL.PRG     48

NPS_CGET              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1183  1277!

NPS_CHECK             (procedure in NPS_APPE.PRG)
   NPS_APPE.PRG    135   228!

NPS_CPAS              (procedure in NPS_UTIL.PRG)
   NPS_UTIL.PRG     54   297!

NPS_CREC              (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG    125   330!

NPS_CREP              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1305  1361!

NPS_CSAY              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1013  1080!

NPS_CSTO              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1108  1156!

NPS_EDIT              (procedure file)
   NPSDC.PRG       134

NPS_FILE              (procedure in NPS_PROC.PRG)
   NPSDC.PRG       149
   NPS_PROC.PRG    722!

NPS_FORMA1            (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    896!  932
   NPS_APPE.PRG     43

NPS_FORMA2            (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    931!
   NPS_EDIT.PRG     43

NPS_FORMB             (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    952!
   NPS_APPE.PRG     45
   NPS_EDIT.PRG     45

NPS_FORMC             (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    977!
   NPS_APPE.PRG     47
   NPS_EDIT.PRG     47

NPS_GETS              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1176!
   NPS_APPE.PRG    113
   NPS_EDIT.PRG    171

NPS_HLP               (other file)
   NPSDC.PRG       190
   NPS_PROC.PRG    122

NPS_INTR              (procedure file)
   NPSDC.PRG        43

NPS_KEYS              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    854!
   NPS_APPE.PRG     85
   NPS_EDIT.PRG    133

NPS_MISC              (database)
   NPS_INTR.PRG     80U
   NPS_OPEN.PRG    153U  175U

NPS_MISC.BU_DATE
   NPSDC.PRG       379   390
   NPS_PROC.PRG    452R

NPS_MISC.PRINTER
   NPS_PROC.PRG   1650

NPS_NDIS              (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    760!
   NPS_EDIT.PRG     77
   NPS_BROW.PRG    133
   NPS_RECA.PRG     449

NPS_OPEN                    (procedure file)
   NPSDC.PRG         84
   NPS_UTIL.PRG     227   236   245   254
   NPS_BU.PRG       776

NPS_PACK                    (procedure file)
   NPSDC.PRG        429
   NPS_UTIL.PRG     221   224   285

NPS_PROC                    (procedure file)
   NPSDC.PRG         36
   NPS_PROC.PRG     259   260   261   262

NPS_RECA                    (procedure file)
   NPSDC.PRG        142

NPS_REPL                    (p···edure in NPS_PROC.PRG)
   NPS_PROC.PRG    1298!
   NPS_APPE.PRG     144
   NPS_EDIT.PRG     174

NPS_REPO                    (procedure file)
   NPSDC.PRG        146

NPS_SAY1                    (procedure in NPS_APPE.PRG)
   NPS_APPE.PRG      69   175!

NPS_SAY2                    (procedure in NPS_APPE.PRG)
   NPS_APPE.PRG      71   198!

NPS_SAY3                    (procedure in NPS_APPE.PRG)
   NPS_APPE.PRG      73   213!

NPS_SAYS                    (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG     470  1006!

NPS_SEEK                    (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG     793!
   NPS_EDIT.PRG      80
   NPS_BROW.PRG     136
   NPS_RECA.PRG     452

NPS_STOR                    (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    1101!
   NPS_APPE.PRG      75
   NPS_EDIT.PRG     118

NPS_USER                    (other file)
   NPSDC.PRG        185

NPS_UTIL                    (procedure file)
   NPSDC.PRG        153

OFFSET
   NPS_PROC.PRG   1528V 1534* 1535  1545  1550* 1562* 1567  1568  1570  1572*

OLDCOLOR
   NPS_INTR.PRG      20V

OLDCUR
   NPS_PROC.PRG     180V  181*  2346

OLDRECNUM
   NPSDC.PRG         59    64
   NPS_PROC.PRG     540   551   552   642V  643*  647   669V  672*  682
   NPS_OPEN.PRG      51
   NPS_EDIT.PRG      79*   84    97*  100   107*  110   119*  146   165

OPEN_CHECKS
   NPSDC.PRG         46    48
   NPS_OPEN.PRG      96   228*

ORIG
   NPS_BU.PRG       368   393   406

ORIGF
   NPSDC.PRG        444V  452*  454   457*  459   462*  464   466*  468
   NPS_OPEN.PRG      43V
   NPS_RECA.PRG      34V   37*   72
   NPS_BU.PRG       174V  197*  199   202*  204   207*  209   211*  213   265*
                    269   272   277*  281   284   289*  293   296   300*  304
                    307   327V  335*  337   340*  342   345*  347   349*  351
                    660V  680*  683   684*  687   688*  691   692*  695   697*
                    703   704*  707   708*  711   712*  715   718*  723   729
                    732*  737   743   746*  751   757   760*  765   771
   NPS_PACK.PRG      43*
```

```
ORIGFI
   NPS_UTIL.PRG     270   285

ORIGFIL
   NPS_UTIL.PRG     196V  213*  215   219*  221   222*  224   231*  233   240*
                    242   249*  251

ORIGFILE
   NPS_PROC.PRG     401   418
   NPS_BU.PRG       793   808   834&  835&  856   871   897&  898&
   NPS_PACK.PRG      80    88&   92    96   100   104   151&  152&

ORIGINDEX
   NPS_PACK.PRG      80   154&

ORIGN
   NPS_BU.PRG       681*  683   685*  687   689*  691   693*  695   698*  703
                    705*  707   709*  711   713*  715   719*  723   729   733*
                    737   743   747*  751   757   761*  765   771
   NPS_PACK.PRG      44*

ORIGND
   NPS_UTIL.PRG     270   285

ORIGNDX
   NPS_UTIL.PRG     195V  214*  215   220*  221   223*  224   232*  233   241*
                    242   250*  251
   NPS_BU.PRG       660V  793   834&  856   899&

PACK_CHECK                  (procedure in NPSDC.PRG)
   NPSDC.PRG         98   419!

PACK_FIL
   NPS_PACK.PRG      20    52    53&   65

PACK_FILE
   NPSDC.PRG        420V  421*  429

PACK_NDX
   NPSDC.PRG        420V  422*  429
   NPS_PACK.PRG      20    65

PACK_POP                    (procedure in NPS_UTIL.PRG)
   NPS_UTIL.PRG      62   194!  196   199   200   201   202   203   209   257x

PAGEPAINT
   NPS_BROW.PRG      26V   52*   76   144*  152*  157*  203*  212*  222*  225*
                    230*  235*  238*  390*
   NPS_RECA.PRG     334V  357*  364   373*  424*  445*  460*  468*  478*  488*

PAINT                       (procedure in NPS_BROW.PRG)
   NPS_BROW.PRG      79   367!

PANCOL
   NPS_BROW.PRG      24V   54*  225   226*  228   229*  229   233   234*  234
                    238   239*  258   262   266   280   282   285   306   309
                    321   323   385   389

PANLAST
   NPS_BROW.PRG      24V   55*  202*  385   389*

PANMAX
   NPS_BROW.PRG      24V   58*   60*   62*  233   238   239

PANO
   NPSDC.PRG        266   269   270   271   272
   NPS_PROC.PRG    1129  1329  1434
   NPS_OPEN.PRG      90    91
   NPS_BROW.PRG     284
   NPS_REPO.PRG      78    79    80   323   325   327

PANOCOLOR
   NPS_PROC.PRG    1430

PANON
   NPS_OPEN.PRG      91

PANOS
   NPS_OPEN.PRG      90

PANPOP                      (procedure in NPS_REPO.PRG)
   NPS_REPO.PRG     136   308!  310   313   314   315   316   317   320   331x

PAN_NAM
   NPS_REPO.PRG      43

PAN_SSN
   NPS_REPO.PRG      44
```

```
PASSTRAN                    (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    1525!
   NPS_INTR.PRG     104
   NPS_UTIL.PRG     372

PASSWORD
   NPS_PROC.PRG     155
   NPS_INTR.PRG     104
   NPS_UTIL.PRG     372R

PASS_STR
   NPS_PROC.PRG    1527 1530  1531  1556  1558  1559  1562  1566

PGDN
   NPSDC.PRG         56    61
   NPS_OPEN.PRG      46*
   NPS_BROW.PRG      42   204
   NPS_RECA.PRG     342*  347   470

PGUP
   NPSDC.PRG         56    61
   NPS_OPEN.PRG      47*
   NPS_BROW.PRG      42   214
   NPS_RECA.PRG     343*  347   480

PHONE_NO
   NPS_PROC.PRG    1370R
   NPS_BROW.PRG     353

POC
   NPS_PROC.PRG    1351R
   NPS_BROW.PRG     324

POPCAL                      (procedure in NPS_PROC.PRG)
   NPSDC.PRG        210
   NPS_PROC.PRG     711!  288

POPCURR
   NPSDC.PRG        280   284
   NPS_PROC.PRG    1503

POPTRIS                     (program file)
   NPSDC.PRG        211
   NPS_PROC.PRG     301

POPUIC
   NPSDC.PRG        274   278
   NPS_PROC.PRG    1465

PRNCHK                      (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    1590!
   NPS_RECA.PRG      67   195   200   205   210   215
   NPS_REPO.PRG     351

PRNDEFAULT                  (procedure in NPS_UTIL.PRG)
   NPS_UTIL.PRG      70   124!  126   129   130   131   134   143x

PRNPROB                     (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    1633!
   NPS_RECA.PRG      81   216   248
   NPS_REPO.PRG     358

PRNRPT                      (procedure in NPS_REPO.PRG)
   NPS_REPO.PRG      98   108   118   127   137   145   153   348!

PRNSTAT
   NPSDC.PRG         46    52*
   NPS_PROC.PRG    1604*
   NPS_RECA.PRG      68   243
   NPS_REPO.PRG     352

PRNSTOP                     (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    1615!
   NPS_RECA.PRG      82   249
   NPS_REPO.PRG     359

PROCEDURE
   NPSDC.PRG         36   172   228   253   295   375   419   442
   NPS_PROC.PRG      71   119   179   251   274   287   300   313   327   399
                    467   499   518   539   566   579   607   640   667   694
                    722   760   793   854   896   931   952   977  1006  1023
                   1064  1080  1101  1118  1141  1156  1176  1198  1258  1277
                   1298  1315  1342  1361  1396  1525  1590  1615  1633  1649
   NPS_INTR.PRG     168
   NPS_OPEN.PRG     195   220   237
   NPS_APPE.PRG     175   198   213   228
   NPS_EDIT.PRG     194   211
   NPS_BROW.PRG     251   297   335   367
   NPS_RECA.PRG     135   177   240   279   330   501   540   673
   NPS_REPO.PRG     164   211   264   308   348
   NPS_UTIL.PRG      81   124   161   194   269   297
   NPS_BU.PRG        79   127   173   326   367   422   450   482   543   586
                    623   659   792   855
   NPS_PACK.PRG      79   119

PROMPTBAR
   NPSDC.PRG         58    63
   NPS_PROC.PRG     904   960   985
   NPS_OPEN.PRG      54=
   NPS_APPE.PRG      82
   NPS_EDIT.PRG     221
   NPS_BROW.PRG      83
   NPS_RECA.PRG     377

PROMPTBU                 (procedure in NPSDC.PRG)
   NPSDC.PRG         96   375|

PROMPTROW
   NPSDC.PRG         58    63
   NPS_PROC.PRG     904   960   985
   NPS_OPEN.PRG      53=
   NPS_APPE.PRG      32    82
   NPS_EDIT.PRG      34   221

QUITBU                   (procedure in NPSDC.PRG)
   NPSDC.PRG        113   442|

RANK_RATE
   NPS_PROC.PRG    1125  1325
   NPS_BROW.PRG     281

RCD_CNT
   NPS_PROC.PRG     402V  406=  431
   NPS_BU.PRG       794V  806=  822   857V  849=  885

RECALL&CNT
   NPS_RECA.PRG     302

RECALL1
   NPS_RECA.PRG      41

RECALL2
   NPS_RECA.PRG      42

RECALL3
   NPS_RECA.PRG      43

RECALL4
   NPS_RECA.PRG      44

RECALLS
   NPS_PROC.PRG     149

RECALLS_MENU             (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG      47   135|

RECALL_&CNT
   NPS_RECA.PRG     303R

RECALL_1
   NPS_EDIT.PRG     195   196R
   NPS_BROW.PRG     286
   NPS_RECA.PRG     100   105   523   551   552   553   554   557   584

RECALL_2
   NPS_EDIT.PRG     197R
   NPS_BROW.PRG     286
   NPS_RECA.PRG      94   100   523   558   585

RECALL_3
   NPS_EDIT.PRG     198R
   NPS_BROW.PRG     286
   NPS_RECA.PRG      86    87    94   524   559   586

RECALL_4
   NPS_EDIT.PRG     199R
   NPS_BROW.PRG     286
   NPS_RECA.PRG      88   524   560   587

RECDISPLAY               (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG     367   501|

RECNUM
   NPSDC.PRG         59    64
   NPS_PROC.PRG     519   521   580   581=  587   591   592   608   609=  619=
                    623=  626=  628  1451V 1452= 1458  1471  1489V 1490= 1496
                   1509
   NPS_OPEN.PRG      51
   NPS_APPE.PRG      30V
   NPS_EDIT.PRG      53=   92
   NPS_BROW.PRG      50=   51    77=  149   150   384
   NPS_RECA.PRG     355=  356   365=  368   465   466

RECNUMLAST
   NPS_BROW.PRG      24V  135=  141
   NPS_RECA.PRG     332V  430=  436   442   451=  457

RECNUMOFS
   NPS_APPE.PRG      30V   33=   61=   61    78   115   126   139=  139   164

RECNUMTOP
   NPS_PROC.PRG     349=  354=  364=  375=
   NPS_BROW.PRG      24V   51=   78   143=  151=  206   211=  216   221<
   NPS_RECA.PRG     332V  356=  366   459=  467=  472   477=  482   487=

RECOK
   NPS_RECA.PRG     599   619   639   658

RECPRN                   (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG     196   201   206   211   220   240|

RECTYPE
   NPS_RECA.PRG      34V   35=  194=  196   199=  201   204=  206   209=  211
                    219=  220   241   254   257&

REC_LISTS                (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG     120   177|  179   182   183   184   185   186   187   190
                    225x

RELATE1
   NPS_OPEN.PRG     162   183
   NPS_RECA.PRG      59
   NPS_REPO.PRG      62

RELATE2
   NPS_OPEN.PRG     179
   NPS_RECA.PRG      74
   NPS_REPO.PRG      48

REPORT_MENU              (procedure in NPS_REPO.PRG)
   NPS_REPO.PRG      52   164|

REQ_SPACE
   NPSDC.PRG         47
   NPS_PACK.PRG      21V   54=   55    58

REST_DISK
   NPS_BU.PRG       624V  638=  641=  679   696   721   724   735   738   749
                    752   763   766

REST_POP                 (procedure in NPS_BU.PRG)
   NPS_BU.PRG       639   642   659|  661   664   665   666   667   668   669
                    675   774x

REST_SELECT              (procedure in NPS_BU.PRG)
   NPS_BU.PRG       610   623|  625   628   629   630   634   644x

RETURNKEY
   NPSDC:PRG         56    61    92
   NPS_PROC.PRG     615   618   675   769   771
   NPS_INTR.PRG      20V
   NPS_OPEN.PRG      48=
   NPS_APPE.PRG      55    59   121   132   150
   NPS_EDIT.PRG      65    67
   NPS_BROW.PRG      42   118
   NPS_RECA.PRG     344=  347   413
   NPS_REPO.PRG      94
   NPS_UTIL.PRG     281
   NPS_BU.PRG       228   383
   NPS_PACK.PRG      39

RIGHTARROW
   NPS_BROW.PRG      25V   40=   42   232
   NPS_RECA.PRG     333V

ROWBOTTOM
   NPS_PROC.PRG     351=  356=  366=  377=  381   382
   NPS_BROW.PRG      26V   44=   45    46   107   119   119   273   314   347
   NPS_RECA.PRG     334V  349=  350   351   402   414   414   516

ROWPROMPT
   NPS_PROC.PRG     336   336   381=
   NPS_BROW.PRG      26V   45=   83    84   133   136   138   140   147   149
   NPS_RECA.PRG     334V  350=  377   378   449   452   454   456   463   465
                    561

ROWTOP
   NPS_PROC.PRG     382
   NPS_BROW.PRG      26V   43=   46    49    72    97   259   260   263   264
                    267   268   273   303   304   314   341   342   347   373
                    378   380   382   387
   NPS_RECA.PRG     334V  348=  351   354   367   370   392   510   511   516
                    547

ROW_HARD_BU
   NPSDC.PRG        101=  103=  105   107   108

RPTNAME
   NPSDC.PRG         47    51
   NPS_REPO.PRG      96=   98   108   118   127   137   144=  145   152=  153
                    226=  229=  232=  235=  241=  244=  247=  250=  349   364&

SAYEOF                   (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG     539|  647   682
   NPS_EDIT.PRG     100   110

SAYLINE                  (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG     546|  644   674
   NPS_EDIT.PRG      82
   NPS_BROW.PRG     138
   NPS_RECA.PRG     454

SAYREC                   (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG     467|  673
   NPS_EDIT.PRG      57    86    93   102   112

SCRNFLD
   NPS_APPE.PRG      34

SETPRN                   (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    1649|
   NPS_RECA.PRG     284
   NPS_REPO.PRG     363

SET_POP                  (procedure in NPSDC.PRG)
   NPSDC.PRG         81   253|

SKIPRECS
   NPS_PROC.PRG     382=
   NPS_BROW.PRG      24V   46=  207   217   378   380   382
   NPS_RECA.PRG     332V  351=  367   473   483

SMC
   NPS_PROC.PRG    1132  1239  1332
   NPS_BROW.PRG     284

SPACE_CHECK              (procedure in NPS_OPEN.PRG)
   NPS_OPEN.PRG      98   195|

SSN
   NPS_PROC.PRG    1121  1321
   NPS_OPEN.PRG      83    83    85    87    90
   NPS_BROW.PRG     281   283   286
   NPS_RECA.PRG     522   550

STATLINE                 (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG     469   518|
   NPS_APPE.PRG      78   115   126

STRG
   NPS_PROC.PRG     567   570

TEMP
   NPS_RECA.PRG     542V  548=  550

TEMPF
   NPS_PACK.PRG      21V   22=   65

TEMPFILE
   NPS_PROC.PRG     403=  410   411&  415&  441   443&  447&
   NPS_BU.PRG       794V  797=  798   799&  812&  818   819&  829   830&  835&
                    857V  860=  861   862&  875&  881   882&  892   893&  898&
   NPS_PACK.PRG      80    84    85&  124&  133   134&  152&

UIC
   NPS_PROC.PRG    1130  1330  1348R
   NPS_OPEN.PRG      87    88   114   157   176
   NPS_EDIT.PRG     177
   NPS_BROW.PRG     284   322   324

UICID
   NPS_PROC.PRG    1450 1454
```

```
UICN
   NPS_OPEN.PRG     88

UICS
   NPS_OPEN.PRG     87

UIC_NAM
   NPS_REPO.PRG     45

UIC_SSN
   NPS_REPO.PRG     46

UNDELRECS
   NPS_PACK.PRG     81V  128   137   140

UPARROW
   NPS_BROW.PRG     25V   37*   41    90    92
   NPS_RECA.PRG    333V  345*  347   385   387

UPDATEREC                (procedure in NPS_EDIT.PRG)
   NPS_EDIT.PRG    176   194|

UTILITIES
   NPS_PROC.PRG    164

UTILITY_MENU             (procedure in NPS_UTIL.PRG)
   NPS_UTIL.PRG     30    81|

VAL_BRAN                 (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1211  1407|

VAL_CURR                 (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1240  1486|

VAL_PAND                 (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1218  1428|

VAL_REC1                 (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG    565   598|

VAL_REC2                 (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG    567   618|

VAL_REC3                 (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG    569   638|

VAL_REC4                 (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG    571   657|

VAL_UIC                  (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG   1220  1448|

VIEWBROW                 (procedure in NPS_PROC.PRG)
   NPS_PROC.PRG    327|
   NPS_BROW.PRG    156
   NPS_RECA.PRG    423

VIEW_RECALLS
   NPS_PROC.PRG    131

WARN
   NPSDC.PRG        47    51   428*
   NPS_UTIL.PRG     61*
   NPS_PACK.PRG     24

WARNING                  (procedure in NPS_UTIL.PRG)
   NPS_UTIL.PRG    215   233   242   251   269|

WARN_CNT
   NPS_RECA.PRG    335V  358   431   432   674  678*

WARN_DBF                 (procedure in NPS_OPEN.PRG)
   NPS_OPEN.PRG     60    67   106   122   137   237|  239   243   250   251x

WARN_WINDOW              (procedure in NPS_RECA.PRG)
   NPS_RECA.PRG    432   673|  676   679   686   687x

WORD_LNGTH
   NPS_PROC.PRG   1527 1531  1537  1553  1559* 1565  1575

WRK_CHAR
   NPS_PROC.PRG   1528V 1532  1538* 1545* 1545  1546  1550  1557  1566* 1567
                  1568  1570  1572

WRK_PASS
   NPS_PROC.PRG   1528V 1532  1535* 1546* 1546  1553* 1557  1568* 1568  1570*
                  1570  1575* 1578

WRK_STRING
   NPS_PROC.PRG   1528V 1531* 1538  1558*

Y
   NPS_INTR.PRG    169  171

Z
   NPS_INTR.PRG    170V  171*  172

-------------------------------------------------------------------------------

System: NPSDC PATIENT TRACKING & RECALL SYSTEM
Author: LCDR Timothy P. Steele, MSC, USN
01/28/92   09:01:11
Macro Summary

-------------------------------------------------------------------------------

   There were no macros defined to FoxDoc

   ----------------------------------------------------------------------------
   Macros Not Defined to FoxDoc
   ----------------------------------------------------------------------------

   &BAKFILE                                   &BU
   &BUFILE                                    &DBFAREA
   &DRIVE                                     &EXPR
   &LSTFILE                                   &NDXORDER
   &OLDCUR                                    &ORIGFILE
   &ORIGINDEX                                 &ORIGNDX
   &PACK_FIL                                  &RECTYPE
   &RPTNAME                                   &TEMPFILE

-------------------------------------------------------------------------------

System: NPSDC PATIENT TRACKING & RECALL SYSTEM
Author: LCDR Timothy P. Steele, MSC, USN
01/28/92   09:01:11
Array Summary

-------------------------------------------------------------------------------

FoxDoc found no arrays in this system

-------------------------------------------------------------------------------
```

```
*:*****************************************************************************
*:
*: Procedure file: C:\PTARS\NPSDC.PRG
*:
*:  Procs & Fncts: ENV_SETUP
*:               : SET_POP
*:               : MAIN_MENU
*:               : PROMPTBU
*:               : PACK_CHECK
*:               : QUITBU
*:               : CONFIG_CHECK
*:
*:          Calls: ENV_SETUP          (procedure in NPSDC.PRG)
*:               : NPS_INTR.PRG
*:               : SET_POP            (procedure in NPSDC.PRG)
*:               : NPS_OPEN.PRG
*:               : MAIN_MENU          (procedure in NPSDC.PRG)
*:               : PROMPTBU           (procedure in NPSDC.PRG)
*:               : PACK_CHECK         (procedure in NPSDC.PRG)
*:               : GETKEY             (procedure in NPS_PROC.PRG)
*:               : QUITBU             (procedure in NPSDC.PRG)
*:               : NPS_APPE.PRG
*:               : NPS_EDIT.PRG
*:               : NPS_BROW.PRG
*:               : NPS_RECA.PRG
*:               : NPS_REPO.PRG
*:               : NPS_FILE           (procedure in NPS_PROC.PRG)
*:               : NPS_UTIL.PRG
*:
*:      Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:*****************************************************************************
*Program: npsdc.prg (PTARS main module)
* ---Setup foxpro environment
DO env_setup
* ---Open PROCEDURE file.
SET PROCEDURE TO nps_proc
* ---SET COLOR TO values.
*ScreenAtr = "W/N,N/W"
*StatusAtr = "W+/N,N/W"
*PromptAtr = "W+/N,N/W"
*HiliteAtr = "W+/N*"
* ---Begin introductory screen, password test, date/time check
DO nps_intr
* --~Declare Global memory variables.
PRIVATE;
   open_checks,no_dbf,mprompt,hlptopic,mode,prnstat,mprint,mcount, ;
   mprintcnt,rptname,warn,req_space,norelate1,norelate2,mbu
STORE .F. TO open_checks,no_dbf
hlptopic = SPACE(6)
STORE 0 TO mcount
STORE "" TO mode,rptname,warn
STORE "N" TO prnstat
STORE .T. TO mprintcnt,norelate1,norelate2
STORE .T. TO mbu          && parameter to nps_open
PRIVATE;
   bu,pgdn,pgup,returnkey,delrecord,APPE,;
   dbfname,dbfarea,dbfpagemax,ndxorder,isedited,;
   promptbar,promptrow,mainchoice,menuchoice,choice,;
   lastrec,recnum,oldrecnum,expr,isvalid
STORE .F. TO;
   bu,pgdn,pgup,returnkey,delrecord,APPE,;
   dbfname,dbfarea,dbfpagemax,ndxorder,isedited,;
   promptbar,promptrow,mainchoice,menuchoice,choice,;
   lastrec,recnum,oldrecnum,expr,isvalid
* ---Declare field memory variables.
PRIVATE;
   mssn,mlast_name,mfirst_nam,mmi,mrank_rate,;
   mbranch,mlast_t2,mclass,mpano,muic,;
   mcurr_num,msec,mrecall_1,mrecall_2,;
   mrecall_3,mrecall_4,;
   muicb,mact_name,macronym,mpoc,;
   mcurr_numc,mcurr_name,mdept_code,mphone_no
STORE " " TO;
   mssn,mlast_name,mfirst_nam,mmi,mrank_rate,;
   mbranch,mclass,mpano,muic,mcurr_num,msec,;
   muicb,mact_name,macronym,mpoc,;
   mcurr_numc,mcurr_name,mdept_code,mphone_no
STORE CTOD("  /  /  ") TO;
   mlast_t2,mrecall_1,mrecall_2,mrecall_3,mrecall_4
* ---Define popups
DO set_pop
* ---Initialize Global memory variables and OPEN file(s).
dbfarea = "1"
DO nps_open
mainchoice = "*"
* ---Execute main loop.
DO WHILE .T.
   * ---Display main menu and get choice.
   DO main_menu WITH mainchoice
   SET COLOR TO W/N,N/W
   DO CASE
   CASE mainchoice $ "Q"+returnkey
      SET COLOR TO W+/N,N/W
      IF ( ! no_dbf )     &&if no database files are missing do quit sequen
*> ce
         * ---Prompt for backup before quitting
         DO promptbu
         * ---Prompt for PACK if more than 10% records deleted before quit
*> ting
         DO pack_check
         * ---Backup the databases before quitting
         IF ( bu )                  &&if backing up to floppy before quitting
            row_hard_bu = 11
         ELSE
            row_hard_bu = 14
         ENDIF
         @ row_hard_bu,33 SAY "              " COLOR G+/N
         ?? CHR(7)
         @ row_hard_bu + 1,0 CLEAR TO 24,79
         @ row_hard_bu + 2,14 SAY "Backup databases to hard disk before qu
*> itting? (y/n)"
         choice = "*"
         DO getkey WITH choice, "YN"
         IF choice = "Y"
            SET CURSOR OFF
            DO quitbu
         ENDIF
      ENDIF
      * --- Quit the program
      SET COLOR TO W/N
      CLOSE ALL
      CLEAR
      @ 0,0 SAY "PTARS 1.0 - Normal shutdown." COLOR W+/N
      QUIT
   CASE .NOT. (mainchoice $ "U") .AND. ((no_dbf = .T.) .OR. (lastrec = 0))
      ?? CHR(7)
      WAIT "MISSING DATABASE(S):  Use Backup Utilities to restore "+;
         "database(s)." WINDOW TIMEOUT 3
   CASE mainchoice = "A"
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---DO APPEND.
      DO nps_appe
   CASE mainchoice = "E"
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---DO EDIT/VIEW.
      isedited = .F.
      DO nps_edit WITH isedited
   CASE mainchoice = "B"
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---BROWSE the database file.
      DO nps_brow
   CASE mainchoice = "R"
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---Print Recalls
      DO nps_reca
   CASE mainchoice = "P"
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---DO Reports
      DO nps_repo
   CASE mainchoice = "S"
      * ---Select a database
      DO nps_file
   CASE mainchoice = "U"
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---Utilities Menu
      DO nps_util
   ENDCASE
ENDDO
*!*****************************************************************************
*!
*!       Procedure: ENV_SETUP
*!
*!       Called by: NPSDC.PRG
*!
*!           Calls: POPCAL             (procedure in NPS_PROC.PRG)
*!                : POPTRIS.PRG
*!                : ABOUT              (procedure in NPS_PROC.PRG)
*!                : FUNC_LIST          (procedure in NPS_PROC.PRG)
*!                : CONFIG_CHECK       (procedure in NPSDC.PRG)
*!
*!     Other Files: NPS_USER
*!                : NPS_HLP
*!
*!*****************************************************************************
PROCEDURE env_setup
CLOSE ALL
SET TALK OFF
SET DEFAULT TO SYS(2004)
SET PATH TO SYS(2004)
IF .NOT. FILE( "NPS_USER.DBF" )
   SET COLOR TO W/N
   ?? CHR(7)
   WAIT "NPS_USER.DBF file not found.  Program quitting..." WINDOW TIMEOUT
*>  2
   CLEAR
   @ 0,0 SAY "NPS_USER.DBF not found.  Check User's Manual." COLOR W+/N
   QUIT
ENDIF
SET RESOURCE TO nps_user
SET COLOR SET TO DEFAULT
SET COLOR OF SCHEME 1 TO W/N
SET CURSOR OFF
CLEAR
SET HELP TO nps_hlp
SET MOUSE OFF
SET SYSMENU OFF
SET STATUS OFF
SET BELL OFF
SET CARRY OFF
SET MENUS OFF
SET DOHISTORY OFF
SET SAFETY OFF
SET ESCAPE OFF
SET SCOREBOARD OFF
SET CLOCK OFF
SET FUNCTION "2" TO
SET FUNCTION "3" TO
SET FUNCTION "4" TO
SET FUNCTION "5" TO
SET FUNCTION "6" TO
SET FUNCTION "7" TO
SET FUNCTION "8" TO
SET FUNCTION "9" TO
ON KEY LABEL f2 DO popcal
ON KEY LABEL f3 DO poptris
ON KEY LABEL f4 DO about
ON KEY LABEL alt+f1 DO func_list
* ---Set Capslock off
= CAPSLOCK(.F.)
* ---Set Numlock off
= NUMLOCK(.F.)
* ---Check config.sys files available
DO config_check
RETURN
*!*****************************************************************************
*!
*!       Procedure: CONFIG_CHECK
*!
*!       Called by: ENV_SETUP          (procedure in NPSDC.PRG)
*!
*!*****************************************************************************
PROCEDURE config_check
* ---Check config.sys files available
STORE VAL(SYS(2010)) TO nfiles
IF nfiles>0 .AND. nfiles<25
   SET COLOR TO W/N
   CLEAR
   ?? CHR(7)
   WAIT 'Config.sys file error' WINDOW TIMEOUT 2
   @ 0,0 SAY 'Invalid Config.sys FILES= setting.  Your setting is '+;
      'FILES='+LTRIM(STR(nfiles,3)) COLOR W+/N
   @ 1,0 SAY 'The config.sys file must be set for a minimum of '+;
      'FILES=25' COLOR W+/N
   @ 2,0 SAY 'Use a text editor to modify and then reboot.' COLOR W+/N
   QUIT
ENDIF
RETURN
*!*****************************************************************************
*!
*!       Procedure: SET_POP
*!
*!       Called by: NPSDC.PRG
*!
*!           Calls: DEACTPOP           (procedure in NPS_PROC.PRG)
*!
*!*****************************************************************************
PROCEDURE set_pop
* ---Sets up "picklist" for branch
DEFINE POPUP branch FROM 4,43 TO 11,49 ;
   MESSAGE "Scroll  or press highlightud letter to locate and select "+;
   "a Service Branch" COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF branch PROMPT "US\<N "
DEFINE BAR 2 OF branch PROMPT "US\<MC"
DEFINE BAR 3 OF branch PROMPT "US\<A "
DEFINE BAR 4 OF branch PROMPT "USA\<F"
DEFINE BAR 5 OF branch PROMPT "US\<CG"
DEFINE BAR 6 OF branch PROMPT "N\<OAA"
ON SELECTION POPUP branch DO deactpop
* ---Sets up "picklist" for pano
DEFINE POPUP pano FROM 7,20 TO 11,47 TITLE "Select Status" ;
   MESSAGE "Scroll  or press highlighted letter to select a Pano Status"
*> ;
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF pano PROMPT "\<GRN - accepted/on-file    "
DEFINE BAR 2 OF pano PROMPT "\<RED - duplicated/forwarded"
DEFINE BAR 3 OF pano PROMPT "\<YLW - neither of above    "
ON SELECTION POPUP pano DO deactpop
* ---Sets up "picklist" for uic
DEFINE POPUP popuic FROM 1,17 TO 21,69 TITLE "Select UIC" ;
   PROMPT FIELD activity.uic+" "+activity.act_name SCROLL ;
   MESSAGE "Scroll  to locate and select a valid UIC";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R
ON SELECTION POPUP popuic DO deactpop
* ---picklist for curriculum number
DEFINE POPUP popcurr FROM 1,29 TO 21,79 TITLE "Select Curriculum" ;
   PROMPT FIELD curricul.curr_num+" "+curricul.curr_name SCROLL ;
   MESSAGE "Scroll  to locate and select a valid Curriculum";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R
ON SELECTION POPUP popcurr DO deactpop
RETURN
*|**********************************************************************
*|
*|       Procedure: MAIN_MENU
*|
*|       Called by: NPSDC.PRG
*|
*| .         Calls: HLP                    (procedure in NPS_PROC.PRG)
*|
*|**********************************************************************
PROCEDURE main_menu
PARAMETER choice
PRIVATE COL
SET COLOR TO W/N,N/W
CLEAR
* ---Object MEMBER is default.
@  0, 0 SAY "MEMBER  ACTIVITY  CURRICULUM  DIRECTOR"
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
DO CASE
CASE dbfarea = "1"
   SET COLOR TO W+/Nm
   @ 0,0 SAY "MEMBER"
CASE dbfarea = "2"
   SET COLOR TO W+/Nm
   @ 0,8 SAY "ACTIVITY"
CASE dbfarea = "3"
   SET COLOR TO W+/Nm
   @ 0,18 SAY "CURRICULUM"
CASE dbfarea = "4"
   SET COLOR TO W+/Nm
   @ 0,30 SAY "DIRECTOR"
ENDCASE
* ---Center the menu heading.
menuhdg = "P T A R S   M A I N   M E N U"
COL = (80 - LEN(menuhdg)) / 2
@  1,0 TO 14,79 DOUBLE COLOR G+/N
@  3, 0 SAY "                                           " COLOR G+/N
@  3,40 SAY "                                           " COLOR G+/N
@  2,COL SAY menuhdg COLOR GR+/N
SET COLOR TO W+/N,N/W
@  5, 3 SAY "    <F1> for help"
@  5, 8 SAY "F1" COLOR GR+/N
@  6, 3 SAY "<Alt+F1> for functions"
@  6, 4 SAY "Alt+F1" COLOR GR+/N
COL = 34
@  5,COL SAY "0. Quit"
@  5,COL + 3 SAY "Q" COLOR GR+/N
@  6,COL SAY "1. Append"
@  6,COL + 3 SAY "A" COLOR GR+/N
@  7,COL SAY "2. Edit/view"
@  7,COL + 3 SAY "E" COLOR GR+/N
@  8,COL SAY "3. Delete/view"
@  8,COL + 3 SAY "D" COLOR GR+/N
@  9,COL SAY "4. Recalls ..."
@  9,COL + 3 SAY "R" COLOR GR+/N
@ 10,COL SAY "5. rePorts ..."
@ 10,COL + 5 SAY "P" COLOR GR+/N
@ 11,COL SAY "6. Select database"
@ 11,COL + 3 SAY "S" COLOR GR+/N
@ 12,COL SAY "7. Utilities ..."
@ 12,COL + 3 SAY "U" COLOR GR+/N
@ 14,33 SAY " select : : "
@ 14,42 SAY ""
* ---Allow either the number or the first letter of a main menu item.
SET INTENSITY OFF
SET CURSOR ON
choice = "*"
DO WHILE .NOT. (choice $ "01234567QAEDRPSU")
   ON KEY LABEL f1 DO hlp WITH "Main"
   choice = " "
   @ 14,42 GET choice PICTURE "!"
   READ
ENDDO
SET INTENSITY ON
* ---Number becomes first letter of menu selection.
IF choice $ "01234567"
   choice = SUBSTR( "QAEDRPSU",VAL( choice ) + 1,1 )
ENDIF
RETURN
*|**********************************************************************
*|
*|       Procedure: PROMPTBU
*|
*|       Called by: NPSDC.PRG
*|
*|           Calls: GETKEY             (procedure in NPS_PROC.PRG)
*|                : NPS_BU.PRG
*|
*|**********************************************************************
PROCEDURE promptbu
* ---Prompt for backup of database if more than 1 month since last backup
SELECT 5
GO TOP
IF nps_misc.bu_date < GOMONTH(DATE(),-1)
   @ 14,33 SAY "              " COLOR G+/N
   SET COLOR TO W+/N
   @ 15,0 CLEAR TO 24,79
   ?? CHR(7)
   @ 16, 1 TO 21,78 COLOR W+/RB
   @ 17, 2 SAY " It has been more than one month since MEMBERS.DBF was "+;
      "backed up to floppy  " COLOR  /RB
   @ 17,70 SAY "floppy" COLOR R+/RB
   @ 18, 2 SAY " disk on          .  To provide insurance against data "+;
      "loss you should back  " COLOR W+/RB
   @ 18,11 SAY nps_misc.bu_date COLOR W+/RB
   @ 19, 2 SAY " up the database to floppy disk at least once a month "+;
      "and whenever exten-   " COLOR W+/RB
   @ 19,34 SAY "at least" COLOR R+/RB
   @ 20, 2 SAY " sive changes are made to the database.  "+SPACE(35) ;
      COLOR W+/RB
   @ 23,17 SAY "Back up the database(s) to floppy disk? (y/n)" COLOR W+/N
   choice = "n"
   DO getkey WITH choice, "YN"
   IF choice = "Y"
      bu = .T.
      @ 24,0 SAY "WORKING ..." COLOR W/Nm
      DO nps_bu
   ENDIF
ENDIF
RETURN
*|**********************************************************************
*|
*|       Procedure: PACK_CHECK
*|
*|       Called by: NPSDC.PRG
*|
*|           Calls: NPS_PACK.PRG
*|
*|            Uses: MEMBERS.DBF
*|
*|       CDX files: MEMBERS.CDX
*|
*|**********************************************************************
PROCEDURE pack_check
PRIVATE pack_file,pack_ndx
pack_file = "MEMBERS.DBF"
pack_ndx = "MEMBERS.CDX"
* ---Prompt for PACK if more than 10% records deleted before quitting
SELECT 1
USE members
COUNT TO mcount FOR DELETED()
IF mcount/RECCOUNT() >= .1
   warn = "Quit"
   DO nps_pack WITH pack_file,pack_ndx
   WAIT "" TIMEOUT 1
ENDIF
RETURN
*|**********************************************************************
*|
*|       Procedure: QUITBU
*|
*|       Called by: NPSDC.PRG
*|
*|           Calls: DBF_BU             (procedure in NPS_PROC.PRG)
*|
*|**********************************************************************
PROCEDURE quitbu
* ---Backup databases prior to quitting
PRIVATE ROW,origf,buf
ROW = 17
SET CURSOR OFF
SET COLOR TO W+/N,N/W
@ 16,0 CLEAR TO 23,79
@ 16,0 SAY "Backing up database files to hard disk ..."
SELECT 1
SET ORDER TO 1
origf = "MEMBERS.DBF"
buf = "MEM_BU.DBF"
DO dbf_bu WITH ROW,origf,buf
SELECT 2
SET ORDER TO 1
origf = "ACTIVITY.DBF"
buf = "ACT_BU.DBF"
DO dbf_bu WITH ROW,origf,buf
SELECT 3
SET ORDER TO 1
origf = "CURRICUL.DBF"
buf = "CUR_BU.DBF"
DO dbf_bu WITH ROW,origf,buf
SELECT 4
origf = "DIRECTOR.DBF"
buf = "DIR_BU.DBF"
DO dbf_bu WITH ROW,origf,buf
RETURN
*: EOF: NPSDC.PRG
```

```
*:*****************************************************************************
*:
*: Procedure file: C:\PTARS\NPS_INTR.PRG
*:
*:  Procs & Fncts: DELAY
*:
*:         Set by: NPSDC.PRG
*:
*:          Calls: DELAY              (procedure in NPS_INTR.PRG)
*:               : PASSTRAN           (procedure in NPS_PROC.PRG)
*:               : GETKEY             (procedure in NPS_PROC.PRG)
*:
*:           Uses: NPS_MISC.DBF
*:
*:      Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:*****************************************************************************
* Program: NPS_INTR.PRG (introductory screen, password test,
* date/time check module)
PRIVATE COL,col_cntr,counter,mpasswrd,;
   oldcolor,keycode,choice,returnkey,colget,colget,mmax,X
* ---Center the intro heading
hdg = "NPSDC PATIENT TRACKING AND RECALL SYSTEM (PTARS)"
COL = (80 - LEN(hdg)) / 2
X = .175
counter = 0
DO WHILE counter < 7
   DO CASE
   CASE counter = 0
      SET COLOR TO BG+/N
   CASE counter = 1
      SET COLOR TO G+/N
   CASE counter = 2
      SET COLOR TO RB+/N
   CASE counter = 3
      SET COLOR TO R+/N
   CASE counter = 4
      SET COLOR TO B+/N
   CASE counter = 5
      SET COLOR TO W+/N
   CASE counter = 6
      SET COLOR TO GR+/N
   ENDCASE
   @  5,15 SAY "                                                  "
   @  6,15 SAY "                                                  "
   @  7,15 SAY "                                                  "
   @  8,15 SAY "                                                  "
   @  9,15 SAY "                                                  "
   @ 10,15 SAY "                                                  "
   @ 11,15 SAY "                                                  "
   @ 12,15 SAY "                                                  "
   @ 13,15 SAY "                                                  "
   @ 14,15 SAY "                                                  "
   @ 15,15 SAY "                                                  "
   DO delay WITH X
   counter = counter + 1
ENDDO
@ 17,COL-1 SAY "NPSDC Patient Tracking and Recall System (PTARS)" ;
   COLOR GR+/N
@ 18,33 SAY "Version 1.0" COLOR GR+/N
WAIT "" TIMEOUT 2
CLEAR
@ 2,COL SAY hdg COLOR GR+/N
@ 3,33 SAY "Access Screen" COLOR GR+/N
@ 1,0 TO 8,79 DOUBLE COLOR G+/N
@ 4,0 SAY "                                        " COLOR G+/N
@ 4,40 SAY "                                        " COLOR G+/N
SET COLOR TO W+/N,N/W
* ---Begin password validation
* ---Check for existence of PASSWORD.DBF
IF .NOT. FILE( "NPS_MISC.DBF" )
   ?? CHR(7)
   WAIT "Password file not found.  Program quitting..." WINDOW TIMEOUT 3
   SET COLOR TO W/N,N/W
   CLEAR
   @ 0,0 SAY "NPS_MISC.DBF not found.  Check User's Manual." COLOR W+/N
   CLOSE ALL
   QUIT
ENDIF
* ---Using NPS_MISC.DBF
USE nps_misc IN E
SELECT E
@11,35 SAY "            " COLOR GR+/N
@12,35 SAY "            " COLOR GR+/N
@13,35 SAY "            " COLOR GR+/N
@14,35 SAY "    1.0     " COLOR GR+/N
* ---Password routine
counter = 0
DO WHILE .T.
   mpasswrd = SPACE(6)
   @ 6,25 SAY "Please type the password:"
   @ 6,52 SAY "      " COLOR N/W
   ?? CHR(7)
   colget = 59
   colsay = 51
   mmax = SPACE(1)
   mpasswrd = ""
   FOR i = 1 TO 6
      @ 6,colget + i GET mmax COLOR  ,X
      READ
      @ 6,colsay + i SAY "."
      mpasswrd = mpasswrd+mmax
      CLEAR GETS
   ENDFOR
   IF UPPER(mpasswrd) == passtran(password)
      EXIT
   ELSE
      ?? CHR(7)
      counter = counter + 1
      DO CASE
      CASE counter < 3
         WAIT "Invalid Password on attempt "+LTRIM(STR(counter))+;
            ".  Try again ..." WINDOW TIMEOUT 1
      CASE counter = 3
         WAIT "Invalid Password on attempt "+LTRIM(STR(counter))+;
            ".  Quitting program ..." WINDOW TIMEOUT 1
         SET COLOR TO W/N
         CLEAR
         @ 0,0 SAY "Password failure entering PTARS.  User rejected." ;
            COLOR W+/N
         QUIT
      ENDCASE
      LOOP
   ENDIF
ENDDO
USE
* ---Check system date
SET CURSOR ON
@ 6,2 CLEAR TO 6,77
?? CHR(7)
SET COLOR TO W+/N
@ 6,11 SAY "The system date is          ."
@ 6,30 SAY DATE() COLOR W+/N*
@ 6,41 SAY "Is this date correct? (y/n)"
choice = "n"
DO getkey WITH choice, "YN"
IF choice = "N"
   @ 11,0 CLEAR TO 20,79
   @ 11,0 SAY ''
   RUN DATE
   @ 11,0 CLEAR TO 23,79
ENDIF
@ 6,2 CLEAR TO 6,77
?? CHR(7)
@ 6,11 SAY "The system time is          ."
@ 6,30 SAY TIME() COLOR W+/N*
@ 6,41 SAY "Is this time correct? (y/n)"
choice = "n"
DO getkey WITH choice, "YN"
IF choice = "N"
   @ 11,0 SAY ''
   RUN TIME
   @ 11,0 CLEAR TO 23,79
ENDIF
SET CURSOR OFF
SET COLOR TO W/N,N/W
@ 1,0 CLEAR TO 10,79
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
SET CLOCK ON
RETURN
*|*****************************************************************************
*|
*|       Procedure: DELAY
*|
*|       Called by: NPS_INTR.PRG
*|
*|*****************************************************************************
PROCEDURE delay
PARAMETER y
PRIVATE z
z = SECONDS() + y
DO WHILE SECONDS() < z
ENDDO
RETURN
*: EOF: NPS_INTR.PRG
```

```
*:*****************************************************************
*:
*: Procedure file: C:\PTARS\NPS_OPEN.PRG
*:
*:  Procs & Fncts: WARN_DBF
*:               : SPACE_CHECK
*:               : UPDATE
*:
*:         Set by: NPSDC.PRG
*:               : PACK_POP            (procedure in NPS_UTIL.PRG)
*:               : REST_POP            (procedure in NPS_BU.PRG)
*:
*:          Calls: WARN_DBF            (procedure in NPS_OPEN.PRG)
*:               : SPACE_CHECK         (procedure in NPS_OPEN.PRG)
*:               : UPDATE              (procedure in NPS_OPEN.PRG)
*:               : NPS_AREA            (procedure in NPS_PROC.PRG)
*:
*:           Uses: MEMBERS.DBF
*:               : ACTIVITY.DBF
*:               : CURRICUL.DBF
*:               : DIRECTOR.DBF
*:               : NPS_MISC.DBF
*:
*:        Indexes: SSN                      (tag in MEMBERS.CDX)
*:               : NAME                     (tag in MEMBERS.CDX)
*:               : CLASS                    (tag in MEMBERS.CDX)
*:               : CLASN                    (tag in MEMBERS.CDX)
*:               : UICS                     (tag in MEMBERS.CDX)
*:               : UICN                     (tag in MEMBERS.CDX)
*:               : CURR_NUMS                (tag in MEMBERS.CDX)
*:               : PANOS                    (tag in MEMBERS.CDX)
*:               : PANON                    (tag in MEMBERS.CDX)
*:               : ACT_UIC                  (tag in ACTIVITY.CDX)
*:               : CURR_NUMC                (tag in CURRICUL.CDX)
*:
*:      CDX files: MEMBERS.CDX
*:               : ACTIVITY.CDX
*:               : CURRICUL.CDX
*:
*:      Documented 01/28/92 at 09:00              FoxDoc  version 2.10
*:*****************************************************************
*Program: NPS_OPEN.PRG (open files, index, relations module)
PRIVATE ROW,origf,buf,missing_dbf
SET COLOR TO W+/N,N/W
* ---INKEY() constant values.
pgdn = CHR(3)
pgup = CHR(18)
returnkey = CHR(13)
delrecord = CHR(7)
* ---Initialize global variables.
STORE 0 TO lastrec,oldrecnum,recnum,menuchoice
STORE " " TO choice,expr
promptrow = 22
promptbar = REPLICATE( CHR( 196 ),80 )
* ---Check if ALL database files missing
IF  ( ! FILE('MEMBERS.DBF')) .AND.( ! FILE('ACTIVITY.DBF')) .AND. ;
      ( ! FILE('CURRICUL.DBF')) .AND. ( ! FILE('DIRECTOR.DBF'))
   missing_dbf = "ALL databases"
   no_dbf = .T.
   DO warn_dbf WITH missing_dbf
   RETURN
ENDIF
* ---Open members database file.
IF NOT FILE( "MEMBERS.DBF" )
   missing_dbf = "MEMBERS.DBF"
   no_dbf = .T.
   DO warn_dbf WITH missing_dbf
   RETURN
ENDIF
SELECT 1
USE members
* ---Open INDEX file(s).
IF ( ! FILE('members.cdx')) .OR. ( ! FILE('activity.cdx')) .OR. ;
      ( ! FILE('curricul.cdx'))
   @ 19,0 CLEAR TO 23,79
   @ 19,0 SAY "Checking/creating structural index file(s) ..." COLOR W+/N
   WAIT "" TIMEOUT 1
ENDIF
ROW = ROW()
IF NOT FILE( "MEMBERS.CDX" )
   @ ROW,0 SAY [Creating structural index "MEMBERS.CDX" ...      ]
   @ 24,0 SAY "WORKING ..." COLOR W/N*
   INDEX ON ssn TAG ssn ADDITIVE
   INDEX ON last_name-first_name          TAG name ADDITIVE
   INDEX ON STR(class)+ssn TAG class ADDITIVE
   INDEX ON STR(class)+last_name-first_name TAG clasn ADDITIVE
   INDEX ON uic+ssn TAG uics ADDITIVE
   INDEX ON uic+last_name-first_name TAG uicn ADDITIVE
   INDEX ON curr_num TAG curr_nums ADDITIVE
   INDEX ON pano+ssn TAG panos ADDITIVE
   INDEX ON pano+last_name-first_name TAG panon ADDITIVE
   WAIT "" TIMEOUT 1
ENDIF
SET ORDER TO 1
* ---Do initial startup checks
IF ( ! open_checks )
   * ---Check if sufficient diskspace available
   DO space_check
   * ---Change members.class to 4 if full 12 mos since last_t2
   DO UPDATE
ENDIF
* ---Open activity database file.
IF NOT FILE( "ACTIVITY.DBF" )
   missing_dbf = "ACTIVITY.DBF"
   no_dbf = .T.
   DO warn_dbf WITH missing_dbf
   RETURN
ENDIF
SELECT 2
USE activity
* ---Open INDEX file.
IF .NOT. FILE( "ACTIVITY.CDX" )
   @ ROW,0 SAY [Creating structural index "ACTIVITY.CDX" ...      ]
   INDEX ON uic TAG act_uic ADDITIVE
   WAIT "" TIMEOUT 1
ENDIF
SET ORDER TO 1
* ---Open curriculum database file.
IF NOT FILE( "CURRICUL.DBF" )
   missing_dbf = "CURRICUL.DBF"
   no_dbf = .T.
   DO warn_dbf WITH missing_dbf
   RETURN
ENDIF
SELECT 3
USE curricul
* ---Open INDEX file.
IF .NOT. FILE( "CURRICUL.CDX" )
   @ ROW,0 SAY [Creating structural index "CURRICUL.CDX" ...      ]
   INDEX ON curr_num TAG curr_numc ADDITIVE
ENDIF
SET ORDER TO 1
* ---Open director database file.
IF NOT FILE( "DIRECTOR.DBF" )
   missing_dbf = "DIRECTOR.DBF"
   no_dbf = .T.
   DO warn_dbf WITH missing_dbf
   RETURN
ENDIF
SELECT 4
USE DIRECTOR
* ---Open nps_misc database file.
IF NOT FILE( "nps_misc.dbf" )
   SET COLOR TO W/N
   ?? CHR(7)
   WAIT "NPS_MISC.DBF not found.  Program quitting ..." WINDOW TIMEOUT 2
   CLEAR
   @ 0,0 SAY "NPS_MISC.DBF file not found.  Check the User's Manual ..."
*)                COLOR W+/N,N/W
   CLOSE ALL
   QUIT
ENDIF
SELECT 5
USE nps_misc
* ---Set relation(1).
IF NOT FILE( "RELATE1.VUE" ) .OR. norelate1 = .T.
   SELECT 2
   SET RELATION TO uic INTO members ADDITIVE
   SELECT 3
   SET RELATION TO curr_num INTO members ADDITIVE
   SELECT 1
   * ---Save environment to view file 1.
   CREATE VIEW relate1 FROM ENVIRONMENT
   norelate1 = .F.
ENDIF
CLOSE DATABASES
* ---Set up second environment
IF NOT FILE( "RELATE2.VUE" ) .OR. norelate2 = .T.
   SELECT 1
   USE members ORDER 1
   USE activity IN B ORDER 1
   SET RELATION OFF INTO members
   USE curricul IN C ORDER 1
   SET RELATION OFF INTO members
   USE DIRECTOR IN D
   USE nps_misc IN E
   SET RELATION TO uic INTO activity ADDITIVE
   SET RELATION TO curr_num INTO curricul ADDITIVE
   * ---Save environment to view file 2
   CREATE VIEW relate2 FROM ENVIRONMENT
   norelate2 = .F.
ENDIF
* ---Initialize database variables for current workarea
SET VIEW TO relate1
* ---Set up work area definitions
DO nps_area
no_dbf = .F.      &&All database files are present
RETURN
*|*****************************************************************
*|
*|       Procedure: SPACE_CHECK
*|
*|       Called by: NPS_OPEN.PRG
*|
*|*****************************************************************
PROCEDURE space_check
PRIVATE dbf_top,file_size,kbytes
* ---Check if sufficient diskspace available for backups and
* ---temporary sort files
dbf_top = HEADER()
file_size =  dbf_top + (RECSIZE() * RECCOUNT() + 1)
IF DISKSPACE() < (file_size * 4)
   SET COLOR TO W/N
   CLEAR
   ?? CHR(7)
   kbytes = "(minimum "+SUBSTR(LTRIM(STR(file_size)),1,3)+"K req'd)."
   WAIT "Insufficient disk space remains to safely run PTARS "+;
      kbytes WINDOW TIMEOUT 2
   @ 0,0 SAY "Insufficient disk space remains to safely run PTARS "+;
      kbytes COLOR W+/N
   QUIT
ENDIF
RETURN
*|*****************************************************************
*|
*|       Procedure: UPDATE
*|
*|       Called by: NPS_OPEN.PRG
*|
*|*****************************************************************
PROCEDURE UPDATE
* ---Change members.class to 4 if full 12 mos since last_t2
@ 15,0 CLEAR TO 24,79
@ 16,20 SAY "Updating Class 4 status.  Please wait ..." COLOR W+/N
@ 24,0 SAY "WORKING ..." COLOR W/N*
SCAN FOR last_t2 < GOMONTH(DATE(),-12)
   REPLACE class WITH 4
ENDSCAN
open_checks = .T.
RETURN
*|*****************************************************************
*|
*|       Procedure: WARN_DBF
*|
*|       Called by: NPS_OPEN.PRG
*|
*|*****************************************************************
PROCEDURE warn_dbf
PARAMETERS missing_dbf
DEFINE WINDOW warn_dbf FROM 5,14 TO 13,65 DOUBLE NOFLOAT NOGROW ;
   IN SCREEN COLOR W+/R,,GR+/R,W+/R
@ 16,0 CLEAR TO 24,79
?? CHR(7)
ACTIVATE WINDOW warn_dbf
@ 0,19 SAY " WARNING " COLOR GR+/R*
@ 2,15 SAY  "Missing "+missing_dbf COLOR GR+/R
@ 3,3 SAY "Use PTARS Backup Menu to restore "+missing_dbf COLOR GR+/R
@ 4,4 SAY missing_dbf+" MUST be restored to run PTARS" COLOR GR+/R
@ 6,10 SAY "Press any key to continue ..."
WAIT ""
DEACTIVATE WINDOW warn_dbf
RELEASE WINDOW warn_dbf
RETURN
*: EOF: NPS_OPEN.PRG
```

```
*:*****************************************************************************
*:
*: Procedure file: C:\PTARS\NPS_APPE.PRG
*:
*:   Procs & Fncts: NPS_SAY1
*:                : NPS_SAY2
*:                : NPS_SAY3
*:                : NPS_CHECK
*:
*:          Set by: NPSDC.PRG
*:
*:           Calls: NPS_FORMA          (procedure in NPS_PROC.PRG)
*:                : NPS_FORMB          (procedure in NPS_PROC.PRG)
*:                : NPS_FORMC          (procedure in NPS_PROC.PRG)
*:                : HLP                (procedure in NPS_PROC.PRG)
*:                : NPS_SAY1           (procedure in NPS_APPE.PRG)
*:                : NPS_SAY2           (procedure in NPS_APPE.PRG)
*:                : NPS_SAY3           (procedure in NPS_APPE.PRG)
*:                : NPS_STOR           (procedure in NPS_PROC.PRG)
*:                : STATLINE           (procedure in NPS_PROC.PRG)
*:                : NPS_KEYS           (procedure in NPS_PROC.PRG)
*:                : NPS_GETS           (procedure in NPS_PROC.PRG)
*:                : GETKEY             (procedure in NPS_PROC.PRG)
*:                : NPS_CHECK          (procedure in NPS_APPE.PRG)
*:                : NPS_REPL           (procedure in NPS_PROC.PRG)
*:
*:      Documented 01/28/92 at 09:00                FoxDoc  version 2.10
*:*****************************************************************************
*Program: NPS_APPE.PRG (append record module)
PRIVATE ROW,recnum,recnumofs,isedited,isblank,isunique,isdeleted
* ---Initialize local memory variables.
ROW = promptrow+1
recnumofs = 0
STORE .F. TO isblank,isunique,isdeleted,scrnfld
isedited = .F.
APPE = .T.
expr = ""
* ---Set Capslock off
= CAPSLOCK(.F.)
* ---Select input form
DO CASE
CASE dbfarea = "1"
   DO nps_forma
CASE dbfarea = "2"
   DO nps_formb
CASE dbfarea = "3"
   DO nps_formc
CASE dbfarea = "4"
   ?? CHR(7)
   WAIT "DIRECTOR can only be edited" WINDOW TIMEOUT 2
   RETURN
ENDCASE
SET COLOR TO W+/N,N/W
* ---Start by adding one record.
choice = returnkey
* ---The following loop is really a "REPEAT/UNTIL <cond>".
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Appe"
   IF (choice = returnkey)
      * ---Add another record.
      recnumofs = recnumofs + 1
      * ---Initialize memory variables with blanks.
      GOTO BOTTOM
      IF .NOT. EOF()
         SKIP
      ENDIF
      DO CASE
      CASE dbfarea = "1"
         DO nps_say1
      CASE dbfarea = "2"
         DO nps_say2
      CASE dbfarea = "3"
         DO nps_say3
      ENDCASE
      DO nps_stor
      GOTO BOTTOM
   ENDIF
   DO statline WITH lastrec+recnumofs,isdeleted
   @ 0,50 SAY "*BLANK*  "
   * ---Check for duplicate record.
   DO WHILE .T.
      @ promptrow,0 SAY promptbar
      @ ROW,0 SAY "APPEND:  Press <Esc> to abort"+SPACE(51) COLOR W+/BG
      * ---Enter key field values.
      DO nps_keys WITH expr,isblank,isunique
      IF isblank .OR. .NOT. isunique
         EXIT
      ENDIF
      * ---Check for duplicate key in master file.
      SEEK expr
      IF EOF()
         * ---No duplicate key found, so leave.
         EXIT
      ELSE
         * ---Found a duplicate record in the file.
         DO CASE
         CASE dbfarea = "1"
            ?? CHR(7)
            WAIT "DUPLICATE SSN.  Change value to proceed." WINDOW TIMEOUT
*) 2
         CASE dbfarea = "2"
            ?? CHR(7)
            WAIT "DUPLICATE UIC.  Change value to proceed." WINDOW TIMEOUT
*) 2
         CASE dbfarea = "3"
            ?? CHR(7)
            WAIT "DUPLICATE Curriculum.  Change value to proceed." ;
               WINDOW TIMEOUT 2
         ENDCASE
      ENDIF
   ENDDO
   IF isblank
      isdeleted = .T.
   ELSE
      DO nps_gets                     && prompts for field inputs
   ENDIF
   DO statline WITH lastrec+recnumofs,isdeleted
   * ---Loop until Add, Edit, or Finished is selected.
   DO WHILE .T.
      SET CURSOR OFF
      @ ROW,0 SAY "APPEND:  <Return>:add-another  (E)dit  (F)inished  "+;
         "<Del>"+SPACE(24) COLOR W+/BG
      DO getkey WITH choice,"EF"+delrecord+returnkey
      DO CASE
      CASE choice = delrecord
         * ---Toggle IsDeleted flag.
         isdeleted = .NOT. isdeleted
         DO statline WITH lastrec+recnumofs,isdeleted
      CASE choice = "E"
         * ---Re-edit the record.
         isdeleted = .F.
         isedited = .T.
         SET CURSOR ON
      CASE choice $ "F"+returnkey
         * ---Finished, Add-another.
         IF .NOT. isdeleted
            DO nps_check
         ENDIF
         IF isdeleted .OR. LASTKEY() = 27
            * ---Reset offset so as not to increment.
            recnumofs = recnumofs - 1
            *                                  set cursor on
         ELSE
            * ---Save the memvar values.
            APPEND BLANK
            DO nps_repl
            isedited = .F.
            SET CURSOR ON
         ENDIF
      ENDCASE
      * ---Condition to exit inner loop.
      IF choice $ "EF"+returnkey
         EXIT
      ENDIF
   ENDDO
   * ---Condition to exit outer loop.
   IF choice = "F"
      ******************
      *CLOSE DATABASES          These lines inactive for speed on 80286
      *SET VIEW TO RELATE1
      *SELECT &dbfarea
      ******************
      EXIT
   ENDIF
ENDDO
lastrec = lastrec + recnumofs
GOTO TOP
APPE = .F.
RETURN
*!*****************************************************************************
*!
*!       Procedure: NPS_SAY1
*!
*!       Called by: NPS_APPE.PRG
*!
*!*****************************************************************************
PROCEDURE nps_say1
* ---Using MEMBERS.DBF
SET COLOR TO N/W
@  3,16 SAY "   -  -    "
@  5,13 SAY SPACE(22)
@  5,51 SAY SPACE(15)
@  5,75 SAY "  "
@  7,13 SAY "    "
@  7,37 SAY "    "
@  7,58 SAY "  /  /  "
@  7,76 SAY "  "
@  9,15 SAY "    "
@ 12,10 SAY "     "
@ 11,30 SAY SPACE(45) COLOR N/N,N/N
@ 12,30 SAY SPACE(45) COLOR N/N,N/N
@ 13,30 SAY SPACE(49) COLOR N/N,N/N
RETURN
*!*****************************************************************************
*!
*!       Procedure: NPS_SAY2
*!
*!       Called by: NPS_APPE.PRG
*!
*!*****************************************************************************
PROCEDURE nps_say2
* ---Using ACTIVITY.DBF
SET COLOR TO N/W
@ 4, 4 SAY "     "
@ 4,15 SAY SPACE(47)
@ 7, 4 SAY SPACE(10)
@ 7,20 SAY SPACE(20)
RETURN
*!*****************************************************************************
*!
*!       Procedure: NPS_SAY3
*!
*!       Called by: NPS_APPE.PRG
*!
*!*****************************************************************************
PROCEDURE nps_say3
* ---Using CURRICUL.DBF
SET COLOR TO N/W
@ 4, 4 SAY "   "
@ 4,23 SAY SPACE(46)
@ 7, 9 SAY "  "
@ 7,23 SAY "    "
RETURN
*!*****************************************************************************
*!
*!       Procedure: NPS_CHECK
*!
*!       Called by: NPS_APPE.PRG
*!
*!*****************************************************************************
PROCEDURE nps_check
* ---Makes sure important fields correct before appending record
IF isedited
   * ---Check again for duplicate key in master file.
   SEEK mssn
   IF NOT EOF()
      ?? CHR(7)
      WAIT "DUPLICATE key encountered.  Record not saved." WINDOW TIMEOUT
*) 2
      isdeleted = .T.
      EXIT
   ENDIF
ENDIF
DO CASE
CASE dbfarea = "1"
   IF (LEN(ALLTRIM(mssn)) < 11) .OR. (LEN(ALLTRIM(muic)) < 5) ;
        .OR. (EMPTY(mlast_t2)) .OR. (EMPTY(mclass))
      ?? CHR(7)
      WAIT "Incomplete or missing data.  Record not saved." ;
         WINDOW TIMEOUT 2
      isdeleted = .T.
   ENDIF
   IF (muic = "31405") .AND. (LEN(ALLTRIM(mcurr_num)) < 3)
      ?? CHR(7)
      WAIT "Error in Curriculum.  Record not saved." WINDOW TIMEOUT 2
      isdeleted = .T.
   ENDIF
   IF isdeleted
   ENDIF
CASE dbfarea = "2"
   IF (LEN(ALLTRIM( muicb )) < 5) .OR. (EMPTY(ALLTRIM( mact_name ))) ;
        (EMPTY(ALLTRIM( mpoc )))
      ?? CHR(7)
      WAIT "Incomplete or missing data.  Record not saved." ;
```

```
            WINDOW TIMEOUT 2
         isdeleted = .T.
      ENDIF
CASE dbfarea = "3"
   IF (LEN(ALLTRIM(mcurr_numc)) < 3) .OR. (EMPTY(ALLTRIM( mcurr_name )))
         ?? CHR(7)
         WAIT "Incomplete or missing data.  Record not saved." ;
            WINDOW TIMEOUT 2
         isdeleted = .T.
      ENDIF
ENDCASE
RETURN
*: EOF: NPS_APPE.PRG
```

```
*:*********************************************************************
*:
*: Procedure file: C:\PTARS\NPS_EDIT.PRG
*:
*:  Procs & Fncts: DIRECTOR
*:               : UPDATEREC
*:
*:          Set by: NPSBC.PRG
*:
*:           Calls: NPS_FORMA2         (procedure in NPS_PROC.PRG)
*:                : NPS_FORMB          (procedure in NPS_PROC.PRG)
*:                : NPS_FORMC          (procedure in NPS_PROC.PRG)
*:                : DIRECTOR           (procedure in NPS_EDIT.PRG)
*:                : HLP                (procedure in NPS_PROC.PRG)
*:                : SAYREC             (procedure in NPS_PROC.PRG)
*:                : GETKEY             (procedure in NPS_PROC.PRG)
*:                : NPS_NDIS           (procedure in NPS_PROC.PRG)
*:                : NPS_SEEK           (procedure in NPS_PROC.PRG)
*:                : SAYLINE            (procedure in NPS_PROC.PRG)
*:                : DOGOTO             (procedure in NPS_PROC.PRG)
*:                : SAYEOF             (procedure in NPS_PROC.PRG)
*:                : NPS_STOR           (procedure in NPS_PROC.PRG)
*:                : NPS_KEYS           (procedure in NPS_PROC.PRG)
*:                : NPS_GETS           (procedure in NPS_PROC.PRG)
*:                : NPS_REPL           (procedure in NPS_PROC.PRG)
*:                : UPDATEREC          (procedure in NPS_EDIT.PRG)
*:
*:       Documented 01/28/92 at 09:00              FoxDoc  version 2.10
*:*********************************************************************
*Program: NPS_EDIT.PRG (edit/view records module)
PARAMETER isedited
PRIVATE ROW,lastpage,editchoice,ndxchoice,counter, ;
   isblank,isunique,isdeleted,keynow
ROW = promptrow+1
expr = ""
STORE .F. TO isedited,isblank,isunique,isdeleted
* ---Set Capslock off
= CAPSLOCK(.F.)
SET DELETED ON
GO TOP
DO CASE
CASE dbfarea = "1"
   DO nps_forma2
CASE dbfarea = "2"
   DO nps_formb
CASE dbfarea = "3"
   DO nps_formc
CASE dbfarea = "4"
   DO DIRECTOR
   RETURN
ENDCASE
STORE "*" TO editchoice,ndxchoice
recnum = RECNO()
* ---Loop until <Return> is pressed.
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Edit"
   DO sayrec
   SET COLOR TO W+/N,N/W
   IF .NOT. (editchoice $ "NP")
      @ ROW,0 CLEAR
   ENDIF
   SET CURSOR OFF
   @ ROW,0 SAY "EDIT/VIEW:  (E)dit  (F)ind  (G)oto  (N)ext-record  "+;
      "(P)rev-record  <Return>      " COLOR W+/BG
   DO getkey WITH editchoice,"EFGNPS"+returnkey
   DO CASE
   CASE editchoice = returnkey
      *********************
      *CLOSE DATABASES          These lines excluded for speed on 80286
      *SET VIEW TO RELATE1
      *SELECT &dbfarea
      *********************
      SET DELETED OFF
      EXIT
   CASE editchoice = "F"
      * ---Find a record.
      DO nps_ndxs WITH ROW,ndxchoice
      IF ndxchoice > "0"
         oldrecnum = RECNO()
         DO nps_seek WITH ROW
         IF EOF()
            DO sayline WITH ROW,"No find."
            WAIT
            GOTO oldrecnum
         ELSE
            DO sayrec
            isedited = .F.
         ENDIF
      ENDIF
   CASE editchoice = "G"
      * ---Goto a record
      DO dogoto WITH ROW,recnum,lastrec
      DO sayrec
      isedited = .F.
   CASE editchoice = "N"
      * ---Next record.
      oldrecnum = RECNO()
      SKIP + 1
      IF EOF()
         DO sayeof WITH ROW,oldrecnum
      ELSE
         DO sayrec
         isedited = .F.
      ENDIF
   CASE editchoice = "P"
      * ---Previous record.
      oldrecnum = RECNO()
      SKIP -1
      IF BOF()
         DO sayeof WITH ROW,oldrecnum
      ELSE
         DO sayrec
         isedited = .F.
      ENDIF
   CASE editchoice = "E"
      * ---Edit the record.
      isedited = .T.
      DO nps_stor
      oldrecnum = RECNO()
      * ---Check for duplicate record.
      DO CASE
      CASE dbfarea = "1"
         keynow = mssn
      CASE dbfarea = "2"
         keynow = muicb
      CASE dbfarea = "3"
         keynow = mcurr_numc
      ENDCASE
      DO WHILE .T.
         @ ROW, 0 SAY "EDIT:  Press <Esc> to abort"+SPACE(52) COLOR W+/BG
         SET CURSOR ON
         * ---Enter key field values.
         DO nps_keys WITH expr,isblank,isunique
         IF expr = keynow          &&if original key value
            EXIT
         ENDIF
         IF isblank .OR. .NOT. isunique
            EXIT
         ENDIF
         * ---Check for duplicate key in master file.
         SET DELETED OFF
         SEEK expr
         IF EOF()
            * ---No duplicate key found, so leave.
            SET DELETED ON
            GO oldrecnum
            EXIT
         ELSE
            * ---Found a duplicate record in the file.
            DO CASE
            CASE dbfarea = "1"
               ?? CHR(7)
               WAIT "DUPLICATE SSN.  Change value to proceed." WINDOW ;
                  TIMEOUT 2
            CASE dbfarea = "2"
               ?? CHR(7)
               WAIT "DUPLICATE UIC.  Change value to proceed." WINDOW ;
                  TIMEOUT 2
            CASE dbfarea = "3"
               ?? CHR(7)
               WAIT "DUPLICATE Curriculum.  Change value to proceed." ;
                  WINDOW ;
                  TIMEOUT 2
            ENDCASE
            GO oldrecnum
         ENDIF
      ENDDO
      IF isblank
         isdeleted = .T.
      ELSE
         DO nps_gets
      ENDIF
      IF NOT LASTKEY() = 27
         DO nps_repl
         IF dbfarea = '1'
            DO updaterec                    &&Clears recall dates with new la
=> st_t2
            IF (uic <> "31405")
               REPLACE curr_num WITH ""
            ENDIF
         ENDIF
      ELSE
         WAIT "Record edit aborted." WINDOW TIMEOUT 1
      ENDIF
   ENDCASE
ENDDO
RETURN
*!*********************************************************************
*!
*!       Procedure: UPDATEREC
*!
*!       Called by: NPS_EDIT.PRG
*!
*!*********************************************************************
PROCEDURE updaterec
IF last_t2 > recall_1
   REPLACE recall_1 WITH CTOD("")
   REPLACE recall_2 WITH CTOD("")
   REPLACE recall_3 WITH CTOD("")
   REPLACE recall_4 WITH CTOD("")
ENDIF
RETURN
*!*********************************************************************
*!
*!       Procedure: DIRECTOR
*!
*!       Called by: NPS_EDIT.PRG
*!
*!           Calls: HLP                  (procedure in NPS_PROC.PRG)
*!
*!*********************************************************************
PROCEDURE DIRECTOR
PRIVATE adirector
* ---Using DIRECTOR.DBF
SELECT B
adirector = DIRECTOR
CLEAR
SET COLOR TO W+/N,N/W
@  0,30 SAY "<DIRECTOR>"
@  3, 1 SAY "<F1> for Help"
@  3, 2 SAY "F1" COLOR GR+/N
@  promptrow , 0 SAY promptbar
@  ROW, 0 SAY "EDIT:  Press <Esc> to abort"+SPACE(51) COLOR W+/BG
SET COLOR TO W/N,N/W
@ 1,0 TO 7,79 DOUBLE COLOR B+/N
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Dire"
   @  5,3 SAY "Enter new Director:  " GET adirector PICTURE "@!" ;
      VALID(NOT EMPTY(ALLTRIM( adirector )))
   READ
   REPLACE DIRECTOR WITH adirector
   RETURN .F.
ENDDO
RETURN
*: EOF: NPS_EDIT.PRG
```

```
*:*****************************************************************************
*:
*: Procedure file: C:\PTARS\NPS_BROW.PRG
*:
*:  Procs & Fncts: PAINT
*:               : ADISPLAY
*:               : BDISPLAY
*:               : CDISPLAY
*:
*:         Set by: NPSDC.PRG
*:
*:          Calls: HLP                     (procedure in NPS_PROC.PRG)
*:               : PAINT                   (procedure in NPS_BROW.PRG)
*:               : GETKEY                  (procedure in NPS_PROC.PRG)
*:               : NPS_NDXS                (procedure in NPS_PROC.PRG)
*:               : NPS_SEEK                (procedure in NPS_PROC.PRG)
*:               : SAYLINE                 (procedure in NPS_PROC.PRG)
*:               : DOGOTO                  (procedure in NPS_PROC.PRG)
*:               : VIEWBROW                (procedure in NPS_PROC.PRG)
*:
*:      Documented 01/28/92 at 09:00                    FoxDoc  version 2.10
*:*****************************************************************************
*Program: NPS_BROW.PRG (delete/view module)
PRIVATE pancol,panmax,panlast,recnumtop,recnumlast,skiprecs,;
   home,endkey,uparrow,downarrow,leftarrow,rightarrow,;
   ROW,rowtop,rowbottom,rowprompt,keystrokes,pagepaint,;
   isedited
* ---Exit if DIRECTOR.DBF selected
IF dbfarea = "4"
   ?? CHR(7)
   WAIT "DIRECTOR can only be edited" WINDOW TIMEOUT 2
   RETURN
ENDIF
* ---Initialize constants.
home = CHR(1)
endkey = CHR(6)
uparrow = CHR(5)
downarrow = CHR(24)
leftarrow = CHR(19)
rightarrow = CHR(4)
keystrokes = "FGM"+uparrow+downarrow+home+leftarrow+;
   rightarrow+endkey+pgdn+pgup+delrecord+returnkey
rowtop = 4
rowbottom = 21
rowprompt = rowbottom + 2
skiprecs = rowbottom - rowtop + 1
GOTO TOP
* ---Initialize local variables.
ROW = rowtop
recnum = RECNO()
recnumtop = recnum
pagepaint = .T.
isedited = .F.
pancol = 1
panlast = 1
DO CASE
CASE dbfarea = "1"
   panmax = 3
CASE dbfarea = "2"
   panmax = 2
CASE dbfarea = "3"
   panmax = 1
ENDCASE
* ---Perform BROWSE.
SET COLOR TO W/N,N/W
CLEAR
@  0, 0 SAY "File:" COLOR W+/N
@  1, 0 SAY "<F1> for help" COLOR W+/N
@  1, 1 SAY "F1" COLOR GR+/N
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
@ rowtop-3,30 SAY "DELETE/VIEW RECORDS" COLOR W+/N
* ---Loop until (Return) is pressed
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Dele"
   IF pagepaint
      recnum = RECNO()
      GOTO recnumtop
      DO paint
   ENDIF
   SET COLOR TO W+/N,N/W
   SET CURSOR OFF
   @ rowprompt-1,0 SAY promptbar
   @ rowprompt,0 SAY ;
      "DELETE/VIEW: (F)ind  (G)oto  (M)ode  <Arrows>  <PgDn>  "+;
      "<PgUp>  <Del>  <Return>  " COLOR W+/BG
   @ ROW,0 SAY CHR(16)
   DO getkey WITH choice,keystrokes
   * ---Reposition record pointer.
   DO WHILE choice $ uparrow+downarrow
      @ ROW,0 SAY " "
      IF choice = uparrow
         SKIP -1
         DO CASE
         CASE BOF()
            GOTO TOP
         CASE ROW > rowtop
            ROW = ROW - 1
         OTHERWISE
            SKIP
         ENDCASE
      ELSE
         SKIP
         DO CASE
         CASE EOF()
            GOTO BOTTOM
         CASE ROW < rowbottom
            ROW = ROW + 1
         OTHERWISE
            SKIP -1
         ENDCASE
      ENDIF
      @ ROW,0 SAY CHR(16)   && record pointer symbol
      DO getkey WITH choice,keystrokes
   ENDDO
   * ---Prompt line selections.
   DO CASE
   CASE choice = returnkey
      IF rowbottom = 39 OR rowbottom = 46
         CLEAR
         @  0,60 SAY DATE() COLOR N/W
         @  0,68 SAY " " COLOR N/W
         SET DISPLAY TO EGA25
      ENDIF
      *********************
      *CLOSE DATABASES          These lines excluded for speed on 80286
      *SET VIEW TO RELATE1
      *SELECT &dbfarea
      *********************
      EXIT
   CASE choice = "F"
      * ---Find a record.
      DO nps_ndxs WITH rowprompt,choice
      IF choice > "0"
         recnumlast = RECNO()
         DO nps_seek WITH rowprompt
         IF EOF()
            DO sayline WITH rowprompt,"No find."
            WAIT
            @ rowprompt,0 CLEAR
            GOTO recnumlast
         ELSE
            recnumtop = RECNO()
            pagepaint = .T.
         ENDIF
      ENDIF
      @ rowprompt,0 CLEAR
   CASE choice = "G"
      DO dogoto WITH rowprompt,recnum,lastrec
      IF recnum > 0
         recnumtop = RECNO()
         pagepaint = .T.
      ENDIF
   CASE choice = "M"
      * ---Change screen display mode
      DO viewbrow
      pagepaint = .T.
   CASE choice = delrecord
      * ---Delete the record.
      IF DELETED()
         RECALL
      ELSE
         DO CASE
         CASE dbfarea = "1"
            DELETE
         CASE dbfarea = "2"
            muic = activity.uic
            SELECT A
            SET ORDER TO 5
            SEEK muic
            IF NOT FOUND()
               SET ORDER TO &ndxorder
               SELECT B
               DELETE
            ENDIF
            IF FOUND()
               ?? CHR(7)
               WAIT "Members exist in UIC.  Can not delete." ;
                  WINDOW TIMEOUT 2
               SET ORDER TO &ndxorder
               SELECT B
            ENDIF
         CASE dbfarea = "3"
            mcurr_num = curricul.curr_num
            SELECT A
            SET ORDER TO 7
            SEEK mcurr_num
            IF NOT FOUND()
               SET ORDER TO &ndxorder
               SELECT C
               DELETE
            ENDIF
            IF FOUND()
               ?? CHR(7)
               WAIT "Members exist in CURRICULUM.  Can not delete." ;
                  WINDOW TIMEOUT 2
               SET ORDER TO &ndxorder
               SELECT C
            ENDIF
         ENDCASE
      ENDIF
      panlast = 0
      pagepaint = .T.
   CASE choice = pgdn
      IF .NOT. EOF()
         GOTO recnumtop
         SKIP skiprecs
         IF EOF()
            GOTO BOTTOM
         ENDIF
         recnumtop = RECNO()
         pagepaint = .T.
      ENDIF
   CASE choice = pgup
      IF .NOT. BOF()
         GOTO recnumtop
         SKIP -skiprecs
         IF BOF()
            GOTO TOP
         ENDIF
         recnumtop = RECNO()
         pagepaint = .T.
      ENDIF
   CASE choice = home
      pagepaint = (pancol <> 1)
      pancol = 1
   CASE choice = leftarrow
      IF pancol > 1
         pancol = pancol - 1
         pagepaint = .T.
      ENDIF
   CASE choice = rightarrow
      IF pancol < panmax
         pancol = pancol + 1
         pagepaint = .T.
      ENDIF
   CASE choice = endkey
      pagepaint = (pancol <> panmax)
      pancol = panmax
   ENDCASE
ENDDO
GOTO TOP
RETURN
*!*****************************************************************************
*!
*!       Procedure: ADISPLAY
*!
*!       Called by: PAINT                   (procedure in NPS_BROW.PRG)
*!
*!*****************************************************************************
PROCEDURE adisplay
* ---Members.dbf browse display
PARAMETER ROW,listrecs
IF listrecs > 1
   * ---Display heading when listing the entire page.
   SET COLOR TO W+/N,N/W
   DO CASE
   CASE pancol = 1
      @ rowtop-2,0 SAY SPACE(48)+"RANK/"+SPACE(7)
      @ rowtop-1,0 SAY "Record#  SSN---------  LAST NAME--------------  "+;
         "FIRST NAME-----  MI  RATE  ==>"
   CASE pancol = 2
      @ rowtop-2,0 SAY SPACE(60)+"CURRI-"+" SMC/"+SPACE(7)
      @ rowtop-1,0 SAY "Record#  SSN---------  BRANCH  LAST T2-  CLASS  "+;
```

```
         "PANO  UIC--  CULUM  CODE  **) "
   CASE pancol = 3
      @ rowtop-2,0 SAY SPACE(80)
      @ rowtop-1,0 SAY "Record#  SSN--------  RECALL_1  RECALL_2  "+;
         "RECALL_3  RECALL_4                    "
   ENDCASE
   * ---Clear the window area.
   SET COLOR TO W/N,N/W
   @ rowtop,0 CLEAR TO rowbottom,79
ENDIF
* ---Display the records.
SET HEADING OFF
SET COLOR TO W/N,N/W
@ ROW-1,0 SAY ""
DO CASE
CASE pancol = 1
   LIST NEXT listrecs ssn,"",last_name,"",first_name,"",mi," ",rank_rate
CASE pancol = 2
   LIST NEXT listrecs ssn,"",branch,"  ",last_t2,"",class,"    ",;
      pano," ",uic,"",curr_num,"  ",smc
CASE pancol = 3
   LIST NEXT listrecs ssn,"",recall_1,"",recall_2,"",recall_3,"",recall_4
ENDCASE
SET HEADING ON
RETURN
*|*****************************************************************
*|
*|      Procedure: BDISPLAY
*|
*|      Called by: PAINT                 (procedure in NPS_BROW.PRG)
*|
*|*****************************************************************
PROCEDURE bdisplay
* ---Activity.dbf browse display
PARAMETER ROW,listrecs
IF listrecs > 1
   * ---Display heading when listing the entire page.
   SET COLOR TO W+/N,N/W
   @ rowtop-1,0
   @ rowtop-1,0 SAY "Record#  "
   DO CASE
   CASE pancol = 1
      ?? "UIC--  ACRONYM----  ACTIVITY NAME-----------------------------"+;
         "------ **)"
   CASE pancol = 2
      ?? "UIC--  POC (Point of Contact)"
   ENDCASE
   * ---Clear the window area.
   SET COLOR TO W/N,N/W
   @ rowtop,0 CLEAR TO rowbottom,79
ENDIF
* ---Display the records.
SET HEADING OFF
SET COLOR TO W/N,N/W
@ ROW-1,0 SAY ""
DO CASE
CASE pancol = 1
   LIST NEXT listrecs uic,"",acronym,"",act_name
CASE pancol = 2
   LIST NEXT listrecs uic,"",poc
ENDCASE
SET HEADING ON
RETURN
*|*****************************************************************
*|
*|      Procedure: CDISPLAY
*|
*|      Called by: PAINT                 (procedure in NPS_BROW.PRG)
*|
*|*****************************************************************
PROCEDURE cdisplay
* ---Curricul.dbf browse display
PARAMETER ROW,listrecs
IF listrecs > 1
   * ---Display heading when listing the entire page.
   SET COLOR TO W+/N,N/W
   @ rowtop-1,0
   @ rowtop-1,0 SAY "Record#  "
   ?? "CURR #  CURRICULUM NAME------------------------------  "+;
      "DEPT #  PHONE #"
   * ---Clear the window area.
   SET COLOR TO W/N,N/W
   @ rowtop,0 CLEAR TO rowbottom,79
ENDIF
* ---Display the records.
SET HEADING OFF
SET COLOR TO W/N,N/W
@ ROW-1,0 SAY ""
LIST NEXT listrecs curr_num,"   ",curr_name,"",dept_code,"    ",phone_no
SET HEADING ON
RETURN
*|*****************************************************************
*|
*:      Procedure: PAINT
*|
*|      Called by: NPS_BROW.PRG
*|
*|         Calls: ADISPLAY           (procedure in NPS_BROW.PRG)
*|              : BDISPLAY           (procedure in NPS_BROW.PRG)
*|              : CDISPLAY           (procedure in NPS_BROW.PRG)
*|
*|*****************************************************************
PROCEDURE paint
@  0, 0 SAY "File:" COLOR W+/N
@  1, 0 SAY "<F1> for help" COLOR W+/N
@  1, 1 SAY "F1" COLOR GR+/N
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
@ rowtop-3,30 SAY "DELETE/VIEW RECORDS" COLOR W+/N
SET COLOR TO W+/N*,N/W
@ 0,6 SAY dbfname
DO CASE
CASE dbfarea = "1"
   DO adisplay WITH (rowtop),skiprecs
CASE dbfarea = "2"
   DO bdisplay WITH (rowtop),skiprecs
CASE dbfarea = "3"
   DO cdisplay WITH (rowtop),skiprecs
ENDCASE
GOTO recnum
IF pancol = panlast
   * ---Reposition record pointer when repainting current page.
   ROW = rowtop
ENDIF
panlast = pancol
pagepaint = .F.
RETURN
*: EOF: NPS_BROW.PRG
```

```
*:**********************************************************************
*:
*: Procedure file: C:\PTARS\NPS_RECA.PRG
*:
*:  Procs & Fncts: RECALLS_MENU
*:               : GORECALLS
*:               : REC_LISTS
*:               : NPS_CREC
*:               : RECPRN
*:               : RECDISPLAY
*:               : EDIT
*:               : VAL_REC1
*:               : VAL_REC2
*:               : VAL_REC3
*:               : VAL_REC4
*:               : WARN_WINDOW
*:
*:         Set by: NPSDC.PRG
*:
*:          Calls: HLP                (procedure in NPS_PROC.PRG)
*:               : RECALLS_MENU       (procedure in NPS_RECA.PRG)
*:               : GETKEY             (procedure in NPS_PROC.PRG)
*:               : PRNCHK             (procedure in NPS_PROC.PRG)
*:               : DBF_BU             (procedure in NPS_PROC.PRG)
*:               : PRNPROB            (procedure in NPS_PROC.PRG)
*:               : PRNSTOP            (procedure in NPS_PROC.PRG)
*:               : GORECALLS          (procedure in NPS_RECA.PRG)
*:               : REC_LISTS          (procedure in NPS_RECA.PRG)
*:               : NPS_CREC           (procedure in NPS_RECA.PRG)
*:
*:      Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:**********************************************************************
*Program: NPS_RECA.PRG (view/edit recall dates module)
PRIVATE rectype,counter,CNT,ROW,origf,buf
STORE " " TO rectype
ROW = 8
origf = "MEMBERS.DBF"
buf = "MEM_BU.DBF"
SET DELETED ON
* ---Identify report forms to project manager
EXTERNAL REPORT recall1
EXTERNAL REPORT recall2
EXTERNAL REPORT recall3
EXTERNAL REPORT recall4
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Reca"
   DO recalls_menu
   DO getkey WITH choice, "0123EPRV"
   @ 10,42 SAY choice
   DO CASE
   CASE choice = "0" .OR. choice = "E"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      ON ERROR                  &&reset
      ON ESCAPE           &&reset
      SET FILTER TO
      SET DELETED OFF
      SET VIEW TO relate1
      SELECT &dbfarea
      EXIT
   CASE choice = "1" .OR. choice = "P"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      IF NOT LASTKEY() = 27      &&If not escape key pressed
         DO prnchk                               &&Printer checklist
         IF prnstat = "Y"
            CLEAR
            @ 7,0 SAY "Backing up current database to hard disk before "+;
               "printing recalls ..."
            DO dbf_bu WITH ROW,origf,buf
            @ 24,0 SAY "WORKING ..." COLOR W/N*
            SET VIEW TO relate2
            SET DELETED ON
            * ---Escape routine
            CLEAR
            SET CURSOR OFF
            SET ESCAPE ON
            STORE .T. TO mprintcnt,mprint
            ON ERROR DO prnprob            &&Present error message if printe
*) r problem
            ON ESCAPE DO prnstop                     &&Stop printing if escape
*)  key pressed
            DO WHILE mprint .AND. mprintcnt
               * ---Do recall4 at beginning of 14th month since last_t2
               SET FILTER TO ((last_t2 < GOMONTH(DATE(),-13)) ;
                  .AND. (NOT EMPTY(recall_3)) .AND. ;
                  (recall_3 < GOMONTH(DATE(), -1)+5) ;
                  .AND. (EMPTY(recall_4)))
               CNT = "4"
               DO gorecalls
               * ---Do recall3 at beginning of 13th month since last_t2
               SET FILTER TO ((last_t2 < GOMONTH(DATE(),-12)) ;
                  .AND. (members.recall_2 < GOMONTH(DATE(), -1)+5) ;
                  .AND. (NOT EMPTY(recall_2)) .AND. (EMPTY(recall_3)))
               CNT = "3"
               DO gorecalls
               * ---Do recall2 at beginning of 12th month since last_t2
               SET FILTER TO ((last_t2 < GOMONTH(DATE(),-11)) ;
                  .AND. (members.recall_1 < GOMONTH(DATE(), -1)+5) ;
                  .AND. (NOT EMPTY(recall_1)) .AND. (EMPTY(recall_2)))
               CNT = "2"
               DO gorecalls
               * ---Do recall1 at beginning of 11th month since last_t2
               SET FILTER TO ((last_t2 < GOMONTH(DATE(),-10)) ;
                  .AND. (EMPTY(recall_1)))
               CNT = "1"
               DO gorecalls
               mprintcnt = .F.
            ENDDO
            ON ESCAPE               &&reset
            SET CURSOR ON
         ENDIF
      ENDIF
      ON KEY = 315                  &&reset
      SET FILTER TO
      SET TEXTMERGE TO
   CASE choice = "2" .OR. choice = "R"
      SET CURSOR OFF
      SET COLOR TO W+/N
      DO rec_lists
   CASE choice = "3" .OR. choice = "V"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      DO nps_crec
   ENDCASE
ENDDO
*|**********************************************************************
*|
*|       Procedure: RECALLS_MENU
*|
*|       Called by: NPS_RECA.PRG
*|
*|**********************************************************************
PROCEDURE recalls_menu
PRIVATE COL
CLEAR
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
* ---Center the menu heading.
menuhdg = "P T A R S    R E C A L L S    M E N U"
COL = (80 - LEN(menuhdg)) / 2
@  1,0 TO 10,79 DOUBLE COLOR G+/N
@  3, 0 SAY "                                       " COLOR G+/N
@  3,40 SAY "                                       " COLOR G+/N
@  2,COL SAY menuhdg COLOR GR+/N
SET COLOR TO W+/N,N/W
@  5, 3 SAY "    <F1> for help"
@  5, 8 SAY "F1" COLOR GR+/N
@  6, 3 SAY "<Alt+F1> for functions"
@  6, 4 SAY "Alt+F1" COLOR GR+/N
COL = 34
@  5,COL SAY "0. Exit to main menu"
@  5,COL + 3 SAY "E" COLOR GR+/N
@  6,COL SAY "1. Print recalls"
@  6,COL + 3 SAY "P" COLOR GR+/N
@  7,COL SAY "2. pRint most recent recall list"
@  7,COL + 4 SAY "R" COLOR GR+/N
@  8,COL SAY "3. View/edit recall dates"
@  8,COL + 3 SAY "V" COLOR GR+/N
@ 10,33 SAY " select : : "
SET CURSOR ON
@ 10,42 SAY ""
RETURN
*|**********************************************************************
*|
*|       Procedure: REC_LISTS
*|
*|       Called by: NPS_RECA.PRG
*|
*|           Calls: DEACTPOP           (procedure in NPS_PROC.PRG)
*|                : PRNCHK             (procedure in NPS_PROC.PRG)
*|                : RECPRN             (procedure in NPS_RECA.PRG)
*|                : PRNPROB            (procedure in NPS_PROC.PRG)
*|
*|**********************************************************************
PROCEDURE rec_lists
IF NOT LASTKEY() = 27    &&If not escape key pressed
   DEFINE POPUP rec_lists FROM 4,48 TO 10,79 TITLE "Select" ;
      MESSAGE "Scroll  or press highlighted letter to select list "+;
      "to print" COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
   DEFINE BAR 1 OF rec_lists PROMPT " Recall \<1 "
   DEFINE BAR 2 OF rec_lists PROMPT " Recall \<2 "
   DEFINE BAR 3 OF rec_lists PROMPT " Recall \<3 "
   DEFINE BAR 4 OF rec_lists PROMPT " Recall \<4 "
   DEFINE BAR 5 OF rec_lists PROMPT " \<All       "
   ON SELECTION POPUP rec_lists DO deactpop
   mprompt = ""
   @ 10,33 SAY "              " COLOR G+/N
   ACTIVATE POPUP rec_lists
   STORE .T. TO mprint
   DO CASE
   CASE mprompt = " Recall 1 "
      rectype = "recall1.lst"
      DO prnchk                               &&Printer checklist
      DO recprn WITH rectype
      EJECT
   CASE mprompt = " Recall 2 "
      rectype = "recall2.lst"
      DO prnchk                               &&Printer checklist
      DO recprn WITH rectype
      EJECT
   CASE mprompt = " Recall 3 "
      rectype = "recall3.lst"
      DO prnchk                               &&Printer checklist
      DO recprn WITH rectype
      EJECT
   CASE mprompt = " Recall 4 "
      rectype = "recall4.lst"
      DO prnchk                               &&Printer checklist
      DO recprn WITH rectype
      EJECT
   CASE mprompt = " All       "
      counter = 0
      DO prnchk                                         &&Printer checklist
      ON ERROR DO prnprob                     &&Present error message if printe
*) r problem
      DO WHILE (counter < 4) .AND. mprint
         counter = counter + 1
         rectype = "recall" + LTRIM(STR(counter)) + ".lst"
         DO recprn WITH rectype
      ENDDO
      CLEAR
      EJECT
   ENDCASE
   RELEASE POPUP rec_lists
ENDIF
RETURN
*|**********************************************************************
*|
*|       Procedure: RECPRN
*|
*|       Called by: REC_LISTS          (procedure in NPS_RECA.PRG)
*|
*|           Calls: PRNPROB            (procedure in NPS_PROC.PRG)
*|                : PRNSTOP            (procedure in NPS_PROC.PRG)
*|
*|     Other Files: &RECTYPE
*|
*|**********************************************************************
PROCEDURE recprn
PARAMETER rectype
IF NOT LASTKEY() = 27    &&If not escape key pressed
   IF prnstat = "Y"
      * ---Escape routine
      CLEAR
      SET ESCAPE ON
      STORE .T. TO mprint,mprintcnt
      ON ERROR DO prnprob                     &&Present error message if printe
*) r problem
      ON ESCAPE DO prnstop                    &&Stop printing if escape key pre
*) ssed
      DO WHILE mprintcnt .AND. mprint
         * ---Print recall list
         CLEAR
         IF mprint
            @ 0,24 SAY "Printing " + rectype + " listing ..." COLOR W+/N
            @ 1,24 SAY "Press <Esc> to abort" COLOR W+/N
         ENDIF
         TYPE &rectype TO PRINTER
         mprintcnt = .F.
      ENDDO
      ON ESCAPE          &&reset
   ENDIF
ENDIF
ON KEY = 315             &&reset
SET ESCAPE OFF
RETURN
*|*********************************************************************
*|
*|      Procedure: GORECALLS
*|
*|      Called by: NPS_RECA.PRG
*|
*|           Calls: SETPRN               (procedure in NPS_PROC.PRG)
*|
*|            Uses: &BAKFILE.DBF
*|
*|    Report Forms: RECALL&CNT.FRX
*|
*|*********************************************************************
PROCEDURE gorecalls
IF aprint
   lstfile = "recall"+CNT+".lst"
   bakfile = "recall"+CNT+".bak"
   GO TOP
   DO setprn
   ERASE &bakfile
   IF FILE( '&lstfile' )
      RENAME &lstfile TO &bakfile
   ENDIF
   SET TEXTMERGE TO &lstfile
   SET TEXTMERGE ON
   \ File:  RECALL<<cnt>>.LST
   \ Listing of most recent Recall <<cnt>> letters.  Created on <<DATE()>>
*) at <<TIME()>>.
   \ SSN         Last Name                First Name       MI  Last T2
   \ ----------- ------------------------ ---------------- --  --------
   SET TEXTMERGE OFF
   CLEAR
   IF aprint
      @ 0,24 SAY "Printing recall "+CNT+" letters ..." COLOR W+/N
      @ 1,24 SAY "Press (Esc) to abort" COLOR W+/N
   ENDIF
   DO WHILE NOT EOF() .AND. aprint
      REPORT FORM recall&cnt NEXT 1 TO PRINTER NOCONSOLE
      REPLACE recall_&cnt WITH DATE()
      SET TEXTMERGE ON
      \ <<ssn>>  <<last_name>>  <<first_name>>  <<mi>>   <<last_t2>>
      SET TEXTMERGE OFF
      SKIP
   ENDDO
ENDIF
RETURN
*|*********************************************************************
*|
*|      Procedure: NPS_CREC
*|
*|      Called by: NPS_RECA.PRG
*|
*|           Calls: NPS_AREA             (procedure in NPS_PROC.PRG)
*|                : HLP                  (procedure in NPS_PROC.PRG)
*|                : RECDISPLAY           (procedure in NPS_RECA.PRG)
*|                : GETKEY               (procedure in NPS_PROC.PRG)
*|                : VIEWBROW             (procedure in NPS_PROC.PRG)
*|                : WARN_WINDOW          (procedure in NPS_RECA.PRG)
*|                : EDIT                 (procedure in NPS_RECA.PRG)
*|                : NPS_NDXS             (procedure in NPS_PROC.PRG)
*|                : NPS_SEEK             (procedure in NPS_PROC.PRG)
*|                : SAYLINE              (procedure in NPS_PROC.PRG)
*|                : DOGOTO               (procedure in NPS_PROC.PRG)
*|
*|*********************************************************************
PROCEDURE nps_crec
* ---Edit recall dates
PRIVATE recnumtop,recnumlast,skiprecs, ;
   home,endkey,uparrow,downarrow,leftarrow,rightarrow, ;
   ROW,rowtop,rowbottom,rowprompt,keystrokes,pagepaint, ;
   isedited,warn_cnt
* ---Use members.dbf
SELECT 1
SET ORDER TO 1
dbfarea = "1"
DO nps_area
* ---Initialize constants.
pgdn = CHR(3)
pgup = CHR(18)
returnkey = CHR(13)
uparrow = CHR(5)
downarrow = CHR(24)
keystrokes = "EFGM"+uparrow+downarrow+pgdn+pgup+returnkey
rowtop = 4
rowbottom = 21
rowprompt = rowbottom + 2
skiprecs = rowbottom - rowtop + 1
GOTO TOP
* ---Initialize local variables.
ROW = rowtop
recnum = RECNO()
recnumtop = recnum
pagepaint = .T.
STORE .F. TO isedited,warn_cnt
* ---Perform VIEW of recall fields.
CLEAR
* ---The following loop is really a "REPEAT/UNTIL <cond>".
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Bror"
   IF pagepaint
      recnum = RECNO()
      GOTO recnumtop
      DO recdisplay WITH (rowtop),skiprecs
      GOTO recnum
      IF isedited = .F.
         ROW = rowtop
      ENDIF
      isedited = .F.
      pagepaint = .F.
   ENDIF
   SET COLOR TO W+/N,N/W
   SET CURSOR OFF
   @ rowprompt-1,0 SAY promptbar
   @ rowprompt,0 SAY ;
      "VIEW RECALLS: (E)dit  (F)ind  (G)oto  (M)ode  <Arrows>  <PgDn> "+;
      " <PgUp>  <Return>" COLOR W+/BG
   @ ROW,0 SAY CHR(16)    && Set record pointer
   choice = "*"
   DO getkey WITH choice,keystrokes
   * ---Reposition record pointer.
   DO WHILE choice $ uparrow+downarrow
      @ ROW,0 SAY " "
      IF choice = uparrow
         SKIP -1
         DO CASE
         CASE BOF()
            GOTO TOP
         CASE ROW > rowtop
            ROW = ROW - 1
         OTHERWISE
            SKIP
         ENDCASE
      ELSE
         SKIP
         DO CASE
         CASE EOF()
            GOTO BOTTOM
         CASE ROW < rowbottom
            ROW = ROW + 1
         OTHERWISE
            SKIP -1
         ENDCASE
      ENDIF
      @ ROW,0 SAY CHR(16)
      DO getkey WITH choice,keystrokes
   ENDDO
   * ---Prompt line selections.
   DO CASE
   CASE choice = returnkey
      IF rowbottom = 39 .OR. rowbottom = 46
         CLEAR
         @  0,60 SAY DATE() COLOR N/W
         @  0,68 SAY " " COLOR N/W
         SET DISPLAY TO EGA25
      ENDIF
      EXIT
   CASE choice = "M"
      IF isedited = .F.
         DO viewbrow
         pagepaint = .T.
      ELSE
         ?? CHR(7)
         WAIT "Mode change not available during editing." WINDOW TIMEOUT 1
      ENDIF
   CASE choice = "E"
      recnumlast = RECNO()
      IF ( ! warn_cnt )
         DO warn_window WITH warn_cnt
      ENDIF
      SAVE SCREEN
      DO EDIT WITH isedited
      IF recnumlast = RECNO() .AND. .NOT. isedited
         RESTORE SCREEN
      ELSE
         * ---Redisplay the screen.
         SET COLOR TO W/N,N/W
         CLEAR
         GO recnumlast
         * ---Do not reposition record pointer.
         isedited = .T.
         pagepaint = .T.
      ENDIF
   CASE choice = "F"
      * ---Find a record.
      DO nps_ndxs WITH rowprompt,choice
      IF choice > "0"
         recnumlast = RECNO()
         DO nps_seek WITH rowprompt
         IF EOF()
            DO sayline WITH rowprompt,"No find."
            WAIT
            @ rowprompt,0 CLEAR
            GOTO recnumlast
         ELSE
            recnumtop = RECNO()
            pagepaint = .T.
         ENDIF
      ENDIF
      @ rowprompt,0 CLEAR
   CASE choice = "G"
      DO dogoto WITH rowprompt,recnum,lastrec
      IF recnum > 0
         recnumtop = RECNO()
         pagepaint = .T.
      ENDIF
   CASE choice = pgdn
      IF .NOT. EOF()
         GOTO recnumtop
         SKIP skiprecs
         IF EOF()
            GOTO BOTTOM
         ENDIF
         recnumtop = RECNO()
         pagepaint = .T.
      ENDIF
   CASE choice = pgup
      IF .NOT. BOF()
         GOTO recnumtop
         SKIP -skiprecs
         IF BOF()
            GOTO TOP
         ENDIF
         recnumtop = RECNO()
         pagepaint = .T.
      ENDIF
   ENDCASE
ENDDO
GOTO TOP
RETURN
*|*********************************************************************
*|
*|      Procedure: RECDISPLAY
*|
*|      Called by: NPS_CREC            (procedure in NPS_RECA.PRG)
*|
*|*********************************************************************
PROCEDURE recdisplay
PARAMETER ROW,listrecs
IF listrecs > 1
   * ---Display heading when listing the entire page.
   SET COLOR TO W+/N,N/W
   @  1, 0 SAY "(F1) for help" COLOR W+/N
   @  1, 1 SAY "F1" COLOR GR+/N
   @  0,60 SAY DATE() COLOR N/W
   @   ,68 SAY " " COLOR N/W
   @ rowtop-3,31 SAY "VIEW RECALL DATES"
   @ rowtop-1,0 SAY "Record#  "
   ?? "SSN--------  LAST NAME, FI     RECALL_1  RECALL_2  "+;
      "RECALL_3  RECALL_4"
   * ---Clear the window area.
   SET COLOR TO W/N,N/W
   @ rowtop,0 CLEAR TO rowbottom,79
ENDIF
* ---Display the records.
SET HEADING OFF
SET COLOR TO W/N,N/W
@ ROW-1,0 SAY ""
LIST NEXT listrecs ssn,"",PADR(SUBSTR(ALLTRIM(last_name)+", "+;
   SUBSTR(first_name,1,1),1,15),15)," ",recall_1,"",recall_2,"",;
   recall_3,"",recall_4,""
SET HEADING ON
RETURN
*|************************************************************
*|
*|      Procedure: EDIT
*|
*|      Called by:  NPS_CREC           (procedure in NPS_RECA.PRG)
*|
*|          Calls: HLP                 (procedure in NPS_PROC.PRG)
*|               : VAL_REC1            (procedure in NPS_RECA.PRG)
*|               : VAL_REC2            (procedure in NPS_RECA.PRG)
*|               : VAL_REC3            (procedure in NPS_RECA.PRG)
*|               : VAL_REC4            (procedure in NPS_RECA.PRG)
*| .
*|************************************************************
PROCEDURE EDIT
PARAMETER isedited
PRIVATE ROW,temp
IF NOT LASTKEY() = 27
   ON KEY LABEL f1 DO hlp WITH "Edit"
   isedited = .T.
   ROW = ROW()
   @ rowtop-3,31 SAY "EDIT RECALL DATES"
   temp = PADR(SUBSTR(ALLTRIM(last_name)+", "+;
      SUBSTR(first_name,1,1),1,15),15)
   @ ROW,0 SAY "EDIT  "+CHR(16)+"  "+ssn+"  "+temp+SPACE(43) COLOR W+/W
   @ ROW,40 SAY recall_1 COLOR N/W
   @ ROW,50 SAY recall_1 COLOR N/W
   @ ROW,60 SAY recall_1 COLOR N/W
   @ ROW,70 SAY recall_1 COLOR N/W
   * ---Edit the record.
   * ---Initialize memvars with field contents.
   STORE recall_1   TO mrecall_1
   STORE recall_2   TO mrecall_2
   STORE recall_3   TO mrecall_3
   STORE recall_4   TO mrecall_4
   @ rowprompt,0 SAY "EDIT RECALLS:  Press (Esc) to abort"+SPACE(45) ;
      COLOR W+/BG
   * ---Get recall date data
   SET CURSOR ON
   @  ROW,40 GET mrecall_1   PICTURE "D" VALID(val_rec1(mrecall_1)) ;
      ERROR "Invalid or inconsistent date."
   @  ROW,50 GET mrecall_2   PICTURE "D" VALID(val_rec2(mrecall_2)) ;
      ERROR "Invalid or inconsistent date."
   @  ROW,60 GET mrecall_3   PICTURE "D" VALID(val_rec3(mrecall_3)) ;
      ERROR "Invalid or inconsistent date."
   @  ROW,70 GET mrecall_4   PICTURE "D" VALID(val_rec4(mrecall_4)) ;
      ERROR "Invalid or inconsistent date."
   READ
   * ---Replace data with new
   IF .NOT. EOF()
      * ---Replace only if there is an available record.
      ON ERROR RETURN .F.
      IF LASTKEY() = 27
         WAIT "Record edit aborted." WINDOW TIMEOUT 1
         isedited = .F.
         EXIT
      ENDIF
      REPLACE;
         recall_1   WITH mrecall_1,;
         recall_2   WITH mrecall_2,;
         recall_3   WITH mrecall_3,;
         recall_4   WITH mrecall_4
   ENDIF
ENDIF
RETURN
*|************************************************************
*|
*|      Procedure: VAL_REC1
*|
*|      Called by: EDIT                (procedure in NPS_RECA.PRG)
*|
*|************************************************************
FUNCTION val_rec1
PARAMETERS recok
DO CASE
CASE EMPTY(mrecall_1) .AND. (( ! EMPTY(mrecall_2)) .OR. ;
      ( ! EMPTY(mrecall_3)) .OR. ( ! EMPTY(mrecall_4)))
   RETURN .F.
CASE ( ! EMPTY(mrecall_1)) .AND. ((mrecall_1 > DATE()) .OR. ;
      (mrecall_1 <= CTOD("01/01/89")) .OR. ((mrecall_1 >= mrecall_2) ;
      .AND. ( ! EMPTY(mrecall_2))))
   RETURN .F.
OTHERWISE
   RETURN .T.
ENDCASE
*|************************************************************
*|
*|      Procedure: VAL_REC2
*|
*|      Called by: EDIT                (procedure in NPS_RECA.PRG)
*|
*|************************************************************
FUNCTION val_rec2
PARAMETERS recok
DO CASE
CASE EMPTY(mrecall_2) .AND. (( ! EMPTY(mrecall_3)) .OR. ;
      ( ! EMPTY(mrecall_4)))
   RETURN .F.
CASE ( ! EMPTY(mrecall_2)) .AND. ((mrecall_2 > DATE()) .OR. ;
      (mrecall_2 <= mrecall_1) .OR. EMPTY(mrecall_1) .OR. ;
      ((mrecall_2 >= mrecall_3) .AND. ( ! EMPTY(mrecall_3))))
   RETURN .F.
OTHERWISE
   RETURN .T.
ENDCASE
*|************************************************************
*|
*|      Procedure: VAL_REC3
*|
*|      Called by: EDIT                (procedure in NPS_RECA.PRG)
*|
*|************************************************************
FUNCTION val_rec3
PARAMETERS recok
DO CASE
CASE EMPTY(mrecall_3) .AND. ( ! EMPTY(mrecall_4))
   RETURN .F.
CASE ( ! EMPTY(mrecall_3)) .AND. ((mrecall_3 > DATE()) .OR. ;
      (mrecall_3 <= mrecall_2) .OR. EMPTY(mrecall_1) .OR. EMPTY(mrecall_2)
*>  .OR. ;
      ((mrecall_3 >= mrecall_4) .AND. ( ! EMPTY(mrecall_4))))
   RETURN .F.
OTHERWISE
   RETURN .T.
ENDCASE
*|************************************************************
*|
*|      Procedure: VAL_REC4
*|
*|      Called by: EDIT                (procedure in NPS_RECA.PRG)
*|
*|************************************************************
FUNCTION val_rec4
PARAMETERS recok
IF (! EMPTY(mrecall_4)) .AND. ((mrecall_4 > DATE()) .OR. ;
      (mrecall_4 <= mrecall_3) .OR. EMPTY(mrecall_1) .OR. EMPTY(mrecall_2)
*>  ;
      .OR. EMPTY(mrecall_3))
   RETURN .F.
ELSE
   RETURN .T.
ENDIF
*|************************************************************
*|
*|      Procedure: WARN_WINDOW
*|
*|      Called by: NPS_CREC            (procedure in NPS_RECA.PRG)
*|
*|************************************************************
PROCEDURE warn_window
PARAMETERS warn_cnt
warn_cnt = .T.
DEFINE WINDOW warn_window FROM 4,15 TO 12,64 DOUBLE NOFLOAT NOGROW ;
   IN SCREEN COLOR W+/R,,GR+/R,W+/R
?? CHR(7)
ACTIVATE WINDOW warn_window
@ 0,18 SAY " WARNING " COLOR GR+/R*
@ 2,8 SAY "EDIT RECALLS WITH EXTREME CAUTION" COLOR GR+/R
@ 3,3 SAY "Results of editing affect recalls printing" COLOR GR+/R
@ 4,5 SAY [Review "Edit Recalls" help for details] COLOR GR+/R
@ 6,10 SAY "Press any key to continue ..."
WAIT ""
DEACTIVATE WINDOW warn_window
RELEASE WINDOW warn_window
RETURN
*: EOF: NPS_RECA.PRG
```

```
*:******************************************************************************
*:
*: Procedure file: C:\PTARS\NPS_REPO.PRG
*:
*:  Procs & Fncts: REPORT_MENU
*:               : PRNRPT
*:               : LISTPOP
*:               : CLASSPOP
*:               : PANPOP
*:
*:         Set by: NPSDC.PRG
*:
*:          Calls: HLP                 (procedure in NPS_PROC.PRG)
*:               : REPORT_MENU         (procedure in NPS_REPO.PRG)
*:               : GETKEY              (procedure in NPS_PROC.PRG)
*:               : PRNRPT              (procedure in NPS_REPO.PRG)
*:               : LISTPOP             (procedure in NPS_REPO.PRG)
*:               : CLASSPOP            (procedure in NPS_REPO.PRG)
*:               : PANPOP              (procedure in NPS_REPO.PRG)
*:
*:   Report Forms: CLAS_CTS.FRX
*:
*:      Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:******************************************************************************
*Program: NPS_REPO.PRG (reports module)
PRIVATE mcount1,mcount2,mcount3,mcount4,mtotcount,;
   mcntpano1,mcntpano2,mcntpano3,mtotpano, ;
   mpct_cnt1,mpct_cnt2,mpct_cnt3,mpct_cnt4,mpct_or, ;
   mpct_pano1,mpct_pano2,mpct_pano3
STORE 0 TO mcount1,mcount2,mcount3,mcount4,mtotcount,;
   mcntpano1,mcntpano2,mcntpano3,mtotpano, ;
   mpct_cnt1,mpct_cnt2,mpct_cnt3,mpct_cnt4,mpct_or, ;
   mpct_pano1,mpct_pano2,mpct_pano3
STORE " " TO listtype
* ---Identify report forms to project manager
EXTERNAL REPORT activity
EXTERNAL REPORT all_nam
EXTERNAL REPORT all_ssn.
EXTERNAL REPORT clas_cts
EXTERNAL REPORT clas_nam
EXTERNAL REPORT clas_ssn
EXTERNAL REPORT curricul
EXTERNAL REPORT pan_nam
EXTERNAL REPORT pan_ssn
EXTERNAL REPORT uic_nam
EXTERNAL REPORT uic_ssn
* ---Set up report environment
SET VIEW TO relate2
SET DELETED ON
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Repo"
   DO report_menu
   DO getkey WITH choice, "01234567EOMCUPAR"
   @ 14,42 SAY choice
   DO CASE
   CASE choice = "0" .OR. choice = "E"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      SET FILTER TO
      SET DELETED OFF
      SET VIEW TO relate1
      SELECT &dbfarea
      EXIT
   CASE choice = "1" .OR. choice = "O"
      SET CURSOR OFF
      SET COLOR TO W+/N
      * --- Operational readiness report
      SELECT A
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      SET ORDER TO 1
      GO TOP
      COUNT FOR class = 1 TO mcount1
      COUNT FOR class = 2 TO mcount2
      COUNT FOR class = 3 TO mcount3
      COUNT FOR class = 4 TO mcount4
      mtotcount = mcount1+mcount2+mcount3+mcount4
      COUNT FOR pano="GRN" TO mcntpano1
      COUNT FOR pano="YLW" TO mcntpano2
      COUNT FOR pano="RED" TO mcntpano3
      mtotpano=mcntpano1+mcntpano2+mcntpano3
      mpct_cnt1 = (mcount1/mtotcount) * 100
      mpct_cnt2 = (mcount2/mtotcount) * 100
      mpct_cnt3 = (mcount3/mtotcount) * 100
      mpct_cnt4 = (mcount4/mtotcount) * 100
      mpct_or = (mcount1 + mcount2)/mtotcount * 100   &&Operational Readin
*) ess
      mpct_pano1 = (mcntpano1/mtotpano) * 100
      mpct_pano2 = (mcntpano2/mtotpano) * 100
      mpct_pano3 = (mcntpano3/mtotpano) * 100
      CLEAR
      REPORT FORM clas_cts
      @ 22,30 SAY "Print this report? (y/n)"
      SET CURSOR ON
      DO getkey WITH choice, "YN"+returnkey
      IF choice = "Y"
         rptname = "clas_cts"
         ON KEY = 315 WAIT "Help not available" WINDOW TIMEOUT 1
         DO prnrpt WITH rptname
      ENDIF
   CASE choice = "2" .OR. choice = "M"
      SET CURSOR OFF
      SET COLOR TO W+/N
      * ---All members
      ON KEY = 315 WAIT "Help not available" WINDOW TIMEOUT 1
      SELECT A
      listtype = "all"
      DO listpop
      DO prnrpt WITH rptname
   CASE choice = "3" .OR. choice = "C"
      SET CURSOR OFF
      SET COLOR TO W+/N
      * ---Members by class
      ON KEY = 315 WAIT "Help not available" WINDOW TIMEOUT 1
      SELECT A
      listtype = "cla"
      DO listpop
      DO classpop
      DO prnrpt WITH rptname
   CASE choice = "4" .OR. choice = "U"
      SET CURSOR OFF
      SET COLOR TO W+/N
      * ---Members by UIC
      ON KEY = 315 WAIT "Help not available" WINDOW TIMEOUT 1
      SELECT A
      listtype = "uic"
      DO listpop
      DO prnrpt WITH rptname
   CASE choice = "5" .OR. choice = "P"
      SET CURSOR OFF
      SET COLOR TO W+/N
      *.---All Activities
      ON KEY = 315 WAIT "Help not available" WINDOW TIMEOUT 1
      SELECT A
      listtype = "pan"
      DO listpop
      DO panpop
      DO prnrpt WITH rptname
   CASE choice = "6" .OR. choice = "A"
      SET CURSOR OFF
      SET COLOR TO W+/N
      * ---All Activities
      ON KEY = 315 WAIT "Help not available" WINDOW TIMEOUT 1
      SELECT B
      rptname = "activity"
      DO prnrpt WITH rptname
   CASE choice = "7" .OR. choice = "R"
      SET CURSOR OFF
      SET COLOR TO W+/N
      * ---All Curriculums
      ON KEY = 315 WAIT "Help not available" WINDOW TIMEOUT 1
      SELECT C
      rptname = "curricul"
      DO prnrpt WITH rptname
   ENDCASE
ENDDO
RETURN
*!******************************************************************************
*!
*!      Procedure: REPORT_MENU
*!
*!      Called by: NPS_REPO.PRG
*!
*!******************************************************************************
PROCEDURE report_menu
PRIVATE COL,menuhdg
CLEAR
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
* ---Center the menu heading.
menuhdg = "P T A R S    R E P O R T S    M E N U"
COL = (80 - LEN(menuhdg)) / 2
@  1,0 TO 14,79 DOUBLE COLOR G+/N
@  3, 0 SAY "                                                   " COLOR G+/N
@  3,40 SAY "                                                   " COLOR G+/N
@  2,COL SAY menuhdg COLOR GR+/N
SET COLOR TO W+/N,N/W
@  5, 3 SAY "    <F1> for help"
@  5, 8 SAY "F1" COLOR GR+/N
@  6, 3 SAY "<Alt+F1> for functions"
@  6, 4 SAY "Alt+F1" COLOR GR+/N
COL = 34
@  5,COL SAY "0. Exit to main menu"
@  5,COL + 3 SAY "E" COLOR GR+/N
@  6,COL SAY "1. Operational readiness"
@  6,COL + 3 SAY "O" COLOR GR+/N
@  7,COL SAY "2. Members (all)"
@  7,COL + 3 SAY "M" COLOR GR+/N
@  8,COL SAY "3. members by Class"
@  8,COL + 14 SAY "C" COLOR GR+/N
@  9,COL SAY "4. members by UIC (all)"
@  9,COL + 14 SAY "U" COLOR GR+/N
@ 10,COL SAY "5. members by Pano status"
@ 10,COL + 14 SAY "P" COLOR GR+/N
@ 11,COL SAY "6. Activities"
@ 11,COL + 3 SAY "A" COLOR GR+/N
@ 12,COL SAY "7. cuRriculums"
@ 12,COL + 5 SAY "R" COLOR GR+/N
@ 14,33 SAY " select : : "
SET CURSOR ON
@ 14,42 SAY ""
RETURN
*!******************************************************************************
*!
*!      Procedure: LISTPOP
*!
*!      Called by: NPS_REPO.PRG
*!
*!          Calls: DEACTPOP            (procedure in NPS_PROC.PRG)
*!
*!******************************************************************************
PROCEDURE listpop
DEFINE POPUP listpop FROM 7,59 TO 10,79 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select listing type";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF listpop PROMPT "List in \<SSN order  "
DEFINE BAR 2 OF listpop PROMPT "List in \<Alpha order"
ON SELECTION POPUP listpop DO deactpop
mprompt = ""
@ 14,33 SAY "              " COLOR G+/N
ACTIVATE POPUP listpop
DO CASE
CASE mprompt = "List in SSN order  "
   DO CASE
   CASE listtype = "all"
      SET ORDER TO 1
      rptname = "all_ssn"
   CASE listtype = "cla"
      SET ORDER TO 3
      rptname = "clas_ssn"
   CASE listtype = "uic"
      SET ORDER TO 5
      rptname = "uic_ssn"
   CASE listtype = "pan"
      SET ORDER TO 8
      rptname = "pan_ssn"
   ENDCASE
CASE mprompt = "List in Alpha order"
   DO CASE
   CASE listtype = "all"
      SET ORDER TO 2
      rptname = "all_nam"
   CASE listtype = "cla"
      SET ORDER TO 4
      rptname = "clas_nam"
   CASE listtype = "uic"
      SET ORDER TO 6
      rptname = "uic_nam"
   CASE listtype = "pan"
      SET ORDER TO 9
      rptname = "pan_nam"
   ENDCASE
ENDCASE
RELEASE POPUP listpop
RETURN
*!******************************************************************************
*!
*!      Procedure: CLASSPOP
*!
*!      Called by: NPS_REPO.PRG
*!
*!          Calls: DEACTPOP            (procedure in NPS_PROC.PRG)
*!
*!******************************************************************************
```

```
PROCEDURE classpop
IF NOT LASTKEY() = 27    &&If not escape key pressed
   DEFINE POPUP classpop FROM 4,55 TO 12,67 TITLE "Select" ;
      MESSAGE "Scroll  or press highlighted letter to select Class "+;
      "to print" COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
   DEFINE BAR 1 OF classpop PROMPT "Class \<1    "
   DEFINE BAR 2 OF classpop PROMPT "Class \<2    "
   DEFINE BAR 3 OF classpop PROMPT "Class \<3    "
   DEFINE BAR 4 OF classpop PROMPT "Class \<4    "
   DEFINE BAR 5 OF classpop PROMPT "\<Class 1 & 2"
   DEFINE BAR 6 OF classpop PROMPT "C\<lass 3 & 4"
   DEFINE BAR 7 OF classpop PROMPT "\<All Classes"
   ON SELECTION POPUP classpop DO deactpop
   mprompt = ""
   @ 14,33 SAY "              " COLOR G+/N
   ACTIVATE POPUP classpop
   DO CASE
   CASE mprompt = "Class 1    "
      SET FILTER TO class = 1
   CASE mprompt = "Class 2    "
      SET FILTER TO class = 2
   CASE mprompt = "Class 3    "
      SET FILTER TO class = 3
   CASE mprompt = "Class 4    "
      SET F'LTER TO class = 4
   CASE mprompt = "Class 1 & 2"
      SET FILTER TO ((class = 1) .OR. (class = 2))
   CASE mprompt = "Class 3 & 4"
      SET FILTER TO ((class = 3) .OR. (class = 4))
   CASE mprompt = "All Classes"
      SET FILTER TO
   ENDCASE
   RELEASE POPUP classpop
ENDIF
RETURN
*|====================================================================
*|
*|      Procedure: PANPOP
*|
*|      Called by: NPS_REPO.PRG
*|
*|          Calls: DEACTPOP           (procedure in NPS_PROC.PRG)
*|
*|====================================================================
PROCEDURE panpop
IF NOT LASTKEY() = 27    &&If not escape key pressed
   DEFINE POPUP panpop FROM 8,61 TO 13,68 ;
      MESSAGE "Scroll  or press highlighted letter to select Pano "+;
      "Status to print" COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
   DEFINE BAR 1 OF panpop PROMPT "\<Green "
   DEFINE BAR 2 OF panpop PROMPT "\<Red   "
   DEFINE BAR 3 OF panpop PROMPT "\<Yellow"
   DEFINE BAR 4 OF panpop PROMPT "\<All   "
   ON SELECTION POPUP panpop DO deactpop
   mprompt = ""
   @ 14,33 SAY "              " COLOR G+/N
   ACTIVATE POPUP panpop
   DO CASE
   CASE mprompt = "Green "
      SET FILTER TO pano = "GRN"
   CASE mprompt = "Red   "
      SET FILTER TO pano = "RED"
   CASE mprompt = "Yellow"
      SET FILTER TO pano = "YLW"
   CASE mprompt = "All   "
      SET FILTER TO
   ENDCASE
   RELEASE POPUP panpop
ENDIF
RETURN
*|====================================================================
*|
*|      Procedure: PRNRPT
*|
*|      Called by: NPS_REPO.PRG
*|
*|          Calls: PRNCHK             (procedure in NPS_PROC.PRG)
*|               , PRNPROB            (procedure in NPS_PROC.PRG)
*|               , PRNSTOP            (procedure in NPS_PROC.PRG)
*|               , SETPRN             (procedure in NPS_PROC.PRG)
*/
*|   Report Forms: &RPTNAME
*|
*|====================================================================
PROCEDURE prnrpt
PARAMETERS rptname
IF NOT LASTKEY() = 27    &&If not escape key pressed
   DO prnchk                            &&Printer checklist
   IF prnstat = "Y"
      * ---Escape routine
      SET CURSOR OFF
      CLEAR
      SET ESCAPE ON
      STORE .T. TO mprint,mprintcnt
      ON ERROR DO prnprob                 &&Present error message if printe
=> r problem
      ON ESCAPE DO prnstop                &&Stop printing if escape key pre
=> ssed
      DO WHILE mprintcnt .AND. mprint
         @ 10,24 SAY "Sending report to printer ..." COLOR W+/NM
         @ 11,24 SAY "Press (Esc) to abort" COLOR W+/NM
         DO setprn
         REPORT FORM &rptname TO PRINTER NOCONSOLE
         mprintcnt = .F.
      ENDDO
      ON ERROR                  &&reset
      ON ESCAPE          &&reset
      SET CURSOR ON
   ENDIF
ENDIF
ON KEY = 315          &&reset
RETURN
*: EOF: NPS_REPO.PRG
```

```
*:*********************************************************************
*:
*: Procedure file: C:\PTARS\NPS_UTIL.PRG
*:
*:  Procs & Fncts: UTILITY_MENU
*:               : NPS_CPAS
*:               : PACK_POP
*:               : DATE_TIME
*:               : PRNDEFAULT
*:               : WARNING
*:
*:         Set by: NPSDC.PRG
*:
*:          Calls: HLP               (procedure in NPS_PROC.PRG)
*:               : UTILITY_MENU      (procedure in NPS_UTIL.PRG)
*:               : GETKEY            (procedure in NPS_PROC.PRG)
*:               : NPS_BU.PRG
*:               : NPS_CPAS          (procedure in NPS_UTIL.PRG)
*:               : PACK_POP          (procedure in NPS_UTIL.PRG)
*:               : DATE_TIME         (procedure in NPS_UTIL.PRG)
*:               : PRNDEFAULT        (procedure in NPS_UTIL.PRG)
*:
*:      Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:*********************************************************************
* Program: NPS_UTIL.PRG (utilities module)
SET COLOR TO W+/N,N/W
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Util"
   * ---Do Utility Menu
   DO utility_menu
   ON ERROR RETURN .F.
   DO getkey WITH choice, "0123456EBCPAHF"
   @ 12,42 SAY choice
   DO CASE
   CASE choice = "0" .OR. choice = "E"
      SET CURSOR OFF
      SET COLOR TO W+/N
      EXIT
   CASE .NOT. (choice $ "01EB") .AND. ((no_dbf = .T.) .OR. (lastrec = 0))
      ?? CHR(7)
      WAIT "MISSING DATABASE(S):  Use Backup Utilities to restore "+;
         "database(s)." WINDOW TIMEOUT 3
   CASE choice = "1" .OR. choice = "B"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---Start backup utility
      DO nps_bu
   CASE choice = "2" .OR. choice = "C"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---Start change password utility
      DO nps_cpas
      SELECT &dbfarea
   CASE choice = "3" .OR. choice = "P"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 24,0 SAY "WORKING ..." COLOR W/N*
      * ---Start pack program
      warn = "Menu"
      DO pack_pop
   CASE choice = "4" .OR. choice = "H"
      SET CURSOR OFF
      SET COLOR TO W+/N
      DO date_time
   CASE choice = "5" .OR. choice = "F"
      SET CURSOR OFF
      SET COLOR TO W+/N
      DO prndefault
   ENDCASE
ENDDO
RETURN
*!*********************************************************************
*!
*!      Procedure: UTILITY_MENU
*!
*!      Called by: NPS_UTIL.PRG
*!
*!*********************************************************************
PROCEDURE utility_menu
PRIVATE COL,hdg
CLEAR
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
* ---Center the intro heading
hdg = "P T A R S    U T I L I T I E S    M E N U"
COL = (80 - LEN(hdg)) / 2
@ 1,0 TO 12,79 DOUBLE COLOR G+/N
@ 3,0 SAY "                                        " COLOR G+/N
@ 3,40 SAY "                                        " COLOR G+/N
@ 2,COL SAY hdg COLOR GR+/N
SET COLOR TO W+/N,N/W
COL = 34
@  5, 3 SAY "    (F1) for help"
@  5, 8 SAY "F1" COLOR GR+/N
@  6, 3 SAY "(Alt+F1) for functions"
@  6, 4 SAY "Alt+F1" COLOR GR+/N
@  5,COL SAY "0. Exit to main menu"
@  5,COL + 3 SAY "E" COLOR GR+/N
@  6,COL SAY "1. Backup utilities ..."
@  6,COL + 3 SAY "B" COLOR GR+/N
@  7,COL SAY "2. Change password"
@  7,COL + 3 SAY "C" COLOR GR+/N
@  8,COL SAY "3. Pack database(s)"
@  8,COL + 3 SAY "P" COLOR GR+/N
@  9,COL SAY "4. cHange date/time"
@  9,COL + 4 SAY "H" COLOR GR+/N
@ 10,COL SAY "5. deFault printer"
@ 10,COL + 5 SAY "F" COLOR GR+/N
@ 12,33 SAY " select : : "
SET CURSOR ON
@ 12,42 SAY ""
RETURN
*!*********************************************************************
*!
*!      Procedure: PRNDEFAULT
*!
*!      Called by: NPS_UTIL.PRG
*!
*!          Calls: DEACTPOP          (procedure in NPS_PROC.PRG)
*!
*!*********************************************************************
PROCEDURE prndefault
* ---Set default dot matrix printer
DEFINE POPUP prndefault FROM 9,54 TO 12,72 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select a default "+;
   "printer" COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF prndefault PROMPT "\<Epson E/F/J/RX/LQ"
DEFINE BAR 2 OF prndefault PROMPT "\<IBM Proprinter"
ON SELECTION POPUP prndefault DO deactpop
mprompt = ""
@ 12,33 SAY "            " COLOR G+/N
ACTIVATE POPUP prndefault
SELECT E
GO TOP
DO CASE
CASE mprompt = "Epson E/F/J/RX/LQ"
   REPLACE PRINTER WITH mprompt
CASE mprompt = "IBM Proprinter"
   REPLACE PRINTER WITH mprompt
ENDCASE
RELEASE POPUP prndefault
SET CURSOR OFF
IF NOT LASTKEY() = 27
   @ 16,0 SAY mprompt+" is default printer." COLOR W+/N
   WAIT "" TIMEOUT 1
ENDIF
SET CURSOR ON
SELECT &dbfarea
RETURN
*!*********************************************************************
*!
*!      Procedure: DATE_TIME
*!
*!      Called by: NPS_UTIL.PRG
*!
*!          Calls: DEACTPOP          (procedure in NPS_PROC.PRG)
*!
*!*********************************************************************
PROCEDURE date_time
* ---Select and run DOS date or time utility
DEFINE POPUP date_time FROM 8,55 TO 11,67 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select date or time";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF date_time PROMPT "\<Date change"
DEFINE BAR 2 OF date_time PROMPT "\<Time change"
ON SELECTION POPUP date_time DO deactpop
mprompt = ""
@ 12,33 SAY "            " COLOR G+/N
ACTIVATE POPUP date_time
SET COLOR TO W+/N,N/W
@ 16,0 CLEAR TO 24,79
DO CASE
CASE mprompt = "Date change"
   RUN DATE
CASE mprompt = "Time change"
   RUN TIME
ENDCASE
RELEASE POPUP date_time
RETURN
*!*********************************************************************
*!
*!      Procedure: PACK_POP
*!
*!      Called by: NPS_UTIL.PRG
*!
*!          Calls: DEACTPOP          (procedure in NPS_PROC.PRG)
*!               : WARNING           (procedure in NPS_UTIL.PRG)
*!               : NPS_PACK.PRG
*!               : NPS_OPEN.PRG
*!
*!*********************************************************************
PROCEDURE pack_pop
PRIVATE origfil, origndx
DEFINE POPUP pack_pop FROM 6,55 TO 11,66 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select file(s)";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF pack_pop PROMPT "\<All       "
DEFINE BAR 2 OF pack_pop PROMPT "\<Members   "
DEFINE BAR 3 OF pack_pop PROMPT "ac\<Tivity  "
DEFINE BAR 4 OF pack_pop PROMPT "\<Curriculum"
ON SELECTION POPUP pack_pop DO deactpop
mprompt = ""
@ 12,33 SAY "            " COLOR G+/N
SET CURSOR OFF
SET COLOR TO W+/N,N/W
@ 13,0 CLEAR TO 24,79
ACTIVATE POPUP pack_pop
DO CASE
CASE mprompt = "All       "
   CLOSE DATABASES
   origfil = "MEMBERS.DBF"
   origndx = "MEMBERS.CDX"
   DO warning WITH origfil,origndx
   IF choice <> "Y"
      RETURN
   ENDIF
   origfil = "ACTIVITY.DBF"
   origndx = "ACTIVITY.CDX"
   DO nps_pack WITH origfil,origndx
   origfil = "CURRICUL.DBF"
   origndx = "CURRICUL.CDX"
   DO nps_pack WITH origfil,origndx
   * ---Close the datafile and index.
   CLOSE DATABASES
   DO nps_open
   SELECT &dbfarea
CASE mprompt = "Members   "
   CLOSE DATABASES
   origfil = "MEMBERS.DBF"
   origndx = "MEMBERS.CDX"
   DO warning WITH origfil,origndx
   * ---Close the datafile and index.
   CLOSE DATABASES
   DO nps_open
   SELECT &dbfarea
CASE mprompt = "acTivity  "
   CLOSE DATABASES
   origfil = "ACTIVITY.DBF"
   origndx = "ACTIVITY.CDX"
   DO warning WITH origfil,origndx
   * ---Close the datafile and index.
   CLOSE DATABASES
   DO nps_open
   SELECT &dbfarea
CASE mprompt = "Curriculum"
   CLOSE DATABASES
   origfil = "CURRICUL.DBF"
   origndx = "CURRICUL.CDX"
   DO warning WITH origfil,origndx
   * ---Close the datafile and index.
   CLOSE DATABASES
   DO nps_open
   SELECT &dbfarea
ENDCASE
RELEASE POPUP pack_pop
RETURN
*!*********************************************************************
*!
*!      Procedure: WARNING
*!
*!      Called by: PACK_POP          (procedure in NPS_UTIL.PRG)
*!
```

```
*|          Calls: GETKEY            (procedure in NPS_PROC.PRG)
*|               : NPS_PACK.PRG
*|
*|*************************************************************************
PROCEDURE warning
PARAMETERS origfi,orignd
@ 12,33 SAY "              " COLOR G+/N
@ 13,0 CLEAR TO 24,79
@ 16, 3 TO 19,76 COLOR W+/RB
@ 17, 4 SAY "  Packing the database permanently removes records "+;
   "marked for deletion  " COLOR W+/RB
@ 17,27 SAY "permanently" COLOR R+/RB
@ 18, 4 SAY "  and improves the performance of the database.  It "+;
   "is not undoable.    " COLOR W+/RB
SET CURSOR ON
@ 21,18 SAY "Do you want to pack the database now? (y/n)" COLOR W+/N
DO getkey WITH choice,"YN"+returnkey
IF choice <> "Y"
   RETURN
ENDIF
DO nps_pack WITH origfi,orignd
RETURN
*|*************************************************************************
*|
*|       Procedure: NPS_CPAS
*|
*|       Called by: NPS_UTIL.PRG
*|
*|          Calls: HLP               (procedure in NPS_PROC.PRG)
*|               : PASSTRAN          (procedure in NPS_PROC.PRG)
*|
*|*************************************************************************
PROCEDURE nps_cpas
* --- Change password
PRIVATE mnewpass,mverify,colget,colsay,mmax,C
* ---Using NPS_MISC.DBF
SELECT E
CLEAR
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
* --- Center the form heading
menuhdg = "P A S S W O R D   C H A N G E   U T I L I T Y"
COL = (80 - LEN(menuhdg)) / 2
@  1,0 TO 11,79 DOUBLE COLOR G+/N
@  3, 0 SAY "                                              " COLOR G+/N
@  3,40 SAY "                                              " COLOR G+/N
@  2,COL SAY menuhdg  COLOR GR+/N
SET COLOR TO W+/N,N/W
mnewpass = SPACE(6)
mverify = SPACE(6)
ON KEY LABEL f1 DO hlp WITH "Pass"
@  5, 3 SAY "    (F1) for help"
@  5, 8 SAY "F1" COLOR GR+/N
@  6, 3 SAY "<Alt+F1> for functions"
@  6, 4 SAY "Alt+F1" COLOR GR+/N
@  7, 3 SAY "   (Esc) to abort"
@  7, 7 SAY "Esc" COLOR GR+/N
@  5,34 SAY "Enter new password:"
@  5,55 SAY "      " COLOR N/W
colget = 64
colsay = 54
mnewpass = ""
SET CURSOR ON
FOR i = 1 TO 6
   mmax = SPACE(1)
   @ 5,colget + i GET mmax ;
      VALID (UPPER(mmax) $ "ABCDEFGHIJKLMNOPQRSTUVWXYZ") ;
      ERROR ("Only alpha characters allowed.  Continue entry ...") ;
      COLOR  ,I
   READ
   IF LASTKEY() = 27
      SET CURSOR OFF
      WAIT "Password change aborted." WINDOW TIMEOUT 1
      SET CURSOR ON
      EXIT
   ENDIF
   @ 5,colsay + i SAY "."
   mnewpass = mnewpass+mmax
   CLEAR GETS
ENDFOR
IF mnewpass = "      " .OR. (LEN(ALLTRIM(mnewpass)) < 6)
   RETURN
ELSE
   ?? CHR(7)
   @  7,34 SAY "Verify new password:"
   @  7,55 SAY "      " COLOR N/W
   mverify = ""
   FOR i = 1 TO 6
      mmax = SPACE(1)
      @ 7,colget + i GET mmax COLOR  ,I
      READ
      IF LASTKEY() = 27
         SET CURSOR OFF
         WAIT "Password change aborted." WINDOW TIMEOUT 1
         SET CURSOR ON
         CLEAR
         EXIT
      ENDIF
      @ 7,colsay + i SAY "."
      mverify = mverify+mmax
      CLEAR GETS
   ENDFOR
   IF LASTKEY() = 27
      EXIT
   ENDIF
   SET CURSOR OFF
   IF mnewpass = mverify
      REPLACE password WITH passtran(mnewpass,6,.T.)
      @  9,34 SAY "Verification successful."
      WAIT "" TIMEOUT 1
   ELSE
      ?? CHR(7)
      WAIT "Verification failure.  Original password remains in effect." ;
         WINDOW TIMEOUT 1
   ENDIF
ENDIF
SET CURSOR ON
RETURN
*: EOF: NPS_UTIL.PRG
```

```
*:*****************************************************************************
*:
*: Procedure file: C:\PTARS\NPS_BU.PRG
*:
*:  Procs & Fncts: BU_MENU
*:               : BU_SELECT
*:               : LIST_SELECT
*:               : FMT
*:               : RESTOR
*:               : BU
*:               : BU_ALL
*:               : BU_SINGLE
*:               : FLOPPY_SIZE
*:               : LIST_FILES
*:               : REST_SELECT
*:               : REST_POP
*:               : DO_REST_HARD
*:               : DO_REST_FLOP
*:
*:         Set by: PROMPTBU          (procedure in NPSDC.PRG)
*:               : NPS_UTIL.PRG
*:
*:          Calls: HLP               (procedure in NPS_PROC.PRG)
*:               : BU_MENU           (procedure in NPS_BU.PRG)
*:               : GETKEY            (procedure in NPS_PROC.PRG)
*:               : BU_SELECT         (procedure in NPS_BU.PRG)
*:               : LIST_SELECT       (procedure in NPS_BU.PRG)
*:               : FMT               (procedure in NPS_BU.PRG)
*:               : RESTOR            (procedure in NPS_BU.PRG)
*:
*:     * Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:*****************************************************************************
* Program: NPS_BU.PRG (backup/restore module)
PRIVATE bufile
DO WHILE .T.
   ON KEY LABEL f1 DO hlp WITH "Buti"
   * ---Do Backup/Restore Menu
   DO bu_menu
   ON ERROR RETURN .F.
   DO getkey WITH choice, "01234EQBLFR"
   @ 11,42 SAY choice
   DO CASE
   CASE choice = "0" .OR. choice = "E" .OR. choice = "Q"
      SET CURSOR OFF
      SET COLOR TO W+/N
      RETURN
   CASE .NOT. (choice $ "0234EQLFR") .AND. ((no_dbf = .T.) .OR. (lastrec =
 0))
      ?? CHR(7)
      WAIT "MISSING DATABASE(S):  Backup not available.  Restore "+;
         "database(s)." WINDOW TIMEOUT 3
   CASE choice = "1" .OR. choice = "B"
      SET CURSOR OFF
      SET COLOR TO W+/N
      DO bu_select
      SELECT &dbfarea
   CASE choice = "2" .OR. choice = "L"
      SET CURSOR OFF
      SET COLOR TO W+/N
      DO list_select      ,
   CASE choice = "3" .OR. choice = "F"
      SET CURSOR OFF
      SET COLOR TO W+/N
      DO fmt
   CASE choice = "4" .OR. choice = "R"
      SET CURSOR OFF
      SET COLOR TO W+/N
      DO RESTOR
   ENDCASE
ENDDO
ON ERROR
RETURN
*!*****************************************************************************
*!
*!       Procedure: BU_MENU
*!
*!       Called by: NPS_BU.PRG
*!
*!*****************************************************************************
PROCEDURE bu_menu
PRIVATE COL,hdg
CLEAR
* ---Center the intro heading
hdg = "P T A R S    B A C K U P    M E N U"
COL = (80 - LEN(hdg)) / 2
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
SET COLOR TO W+/N,N/W
@ 1,0 TO 11,79 DOUBLE COLOR G+/N
@ 3,0 SAY "                                        " COLOR G+/N
@ 3,40 SAY "                                        " COLOR G+/N
@ 2,COL SAY hdg  COLOR GR+/N
COL = 34
@  5, 3 SAY "    <F1> for help"
@  5, 8 SAY "F1" COLOR GR+/N
@  6, 3 SAY "<Alt+F1> for functions"
@  6, 4 SAY "Alt+F1" COLOR GR+/N
* ---Check if quitting the program
IF bu = .F.
   @  5,COL SAY "0. Exit to utilities menu"
   @  5,COL + 3 SAY "E" COLOR GR+/N
ELSE
   @  5,COL SAY "0. Quit"
   @  5,COL + 3 SAY "Q" COLOR GR+/N
ENDIF
@  6;COL SAY "1. Backup database file(s)"
@  6,COL + 3 SAY "B" COLOR GR+/N
@  7,COL SAY "2. List files"
@  7,COL + 3 SAY "L" COLOR GR+/N
@  8,COL SAY "3. Format floppy disk (A:)"
@  8,COL + 3 SAY "F" COLOR GR+/N
@  9,COL SAY "4. Restore backup(s)"
@  9,COL + 3 SAY "R" COLOR GR+/N
@ 11,33 SAY " select : : "
@ 11,42 SAY ""
SET CURSOR ON
RETURN
*!*****************************************************************************
*!
*!       Procedure: BU_SELECT
*!
*!       Called by: NPS_BU.PRG
*!
*!           Calls: DEACTPOP           (procedure in NPS_PROC.PRG)
*!                : BU                 (procedure in NPS_BU.PRG)
*!
*!*****************************************************************************
PROCEDURE bu_select
PRIVATE bu_disk
DEFINE POPUP bu_select FROM 5,60 TO 8,72 TITLE "Select";
   MESSAGE "Scroll  or press highlighted letter to select "+;
   "destination disk" COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF bu_select PROMPT "\<Hard disk  "
DEFINE BAR 2 OF bu_select PROMPT "\<Floppy disk"
ON SELECTION POPUP bu_select DO deactpop
mprompt = ""
@ 11,33 SAY "              " COLOR G+/N
ACTIVATE POPUP bu_select
SET CURSOR OFF
DO CASE
CASE mprompt = "Hard disk  "
   bu_disk = "H"
   DO bu
CASE mprompt = "Floppy disk"
   bu_disk = "F"
   DO bu
ENDCASE
RELEASE POPUP bu_select
SET CURSOR ON
RETURN
*!*****************************************************************************
*!
*!       Procedure: BU
*!
*!       Called by: BU_SELECT          (procedure in NPS_BU.PRG)
*!
*!           Calls: DEACTPOP           (procedure in NPS_PROC.PRG)
*!                : DBF_BU             (procedure in NPS_PROC.PRG)
*!                : GETKEY             (procedure in NPS_PROC.PRG)
*!                : FLOPPY_SIZE        (procedure in NPS_BU.PRG)
*!                : BU_ALL             (procedure in NPS_BU.PRG)
*!                : BU_SINGLE          (procedure in NPS_BU.PRG)
*!
*!            Uses: MEMBERS.DBF
*!                : ACTIVITY.DBF
*!                : CURRICUL.DBF
*!                : DIRECTOR.DBF
*!
*!       CDX files: MEMBERS.CDX
*!                : ACTIVITY.CDX
*!                : CURRICUL.CDX
*!
*!*****************************************************************************
PROCEDURE bu
PRIVATE no_size,ALL,ROW,origf,buf
ROW = 15
DEFINE POPUP bu FROM 3,60 TO 9,71 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select file(s)";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF bu PROMPT "\<All        "
DEFINE BAR 2 OF bu PROMPT "\<Members    "
DEFINE BAR 3 OF bu PROMPT "ac\<Tivity   "
DEFINE BAR 4 OF bu PROMPT "\<Curriculum"
DEFINE BAR 5 OF bu PROMPT "\<Director   "
ON SELECTION POPUP bu DO deactpop
mprompt = ""
@ 11,33 SAY "              " COLOR G+/N
ACTIVATE POPUP bu
SET CURSOR OFF
@ 12,0 CLEAR TO 24,79
DO CASE
CASE mprompt = "All        "
   DO CASE
   CASE bu_disk = "H"
      SELECT 1
      USE members
      SET ORDER TO 1
      origf = "MEMBERS.DBF"
      buf = "MEM_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
      SELECT 2
      SET ORDER TO 1
      origf = "ACTIVITY.DBF"
      buf = "ACT_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
      SELECT 3
      SET ORDER TO 1
      origf = "CURRICUL.DBF"
      buf = "CUR_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
      SELECT 4
      origf = "DIRECTOR.DBF"
      buf = "DIR_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
   CASE bu_disk = "F"
      @ 15,0 CLEAR 24,79
      @ 15,0 SAY "Insert a formatted backup floppy into the A: drive."
      WAIT
      no_size = .F.
      ALL = .T.
      SET CURSOR ON
      IF FILE( "A:\MEMBERS.DBF" ) .OR. FILE( "A:\ACTIVITY.DBF" );
            .OR. FILE( "A:\CURRICUL.DBF" ) .OR. FILE( "A:\DIRECTOR.DBF" )
         SET CURSOR ON
         @ 15,0 CLEAR 24,79
         ?? CHR(7)
         @ 15,7 SAY "Previous backup file(s) exist on floppy disk.  "+;
            "Replace them? (y/n)"
         DO getkey WITH choice, "YN"+returnkey
         IF choice = "Y"
            SET CURSOR OFF
            SET COLOR TO W+/N
            @ 15,0 CLEAR TO 24,79
            @ 15,0 SAY "Erasing previous backup(s) ..."
            ERASE "a:\members.dbf"
            ERASE "a:\activity.dbf"
            ERASE "a:\curricul.dbf"
            ERASE "a:\director.dbf"
            WAIT "" TIMEOUT 1
            DO floppy_size WITH no_size,ALL
            IF ( ! no_size )
               DO bu_all
            ELSE
               ?? CHR(7)
               WAIT "Insufficient disk space on floppy.  "+;
                  "Aborting backup ..." WINDOW TIMEOUT 2
            ENDIF
         ENDIF
      ELSE
         SET CURSOR OFF
         SET COLOR TO W+/N
         DO floppy_size WITH no_size
         IF ( ! no_size )
            DO bu_all
         ELSE
            ?? CHR(7)
            WAIT "Insufficient disk space on floppy.  "+;
               "Aborting backup ..." WINDOW TIMEOUT 2
         ENDIF
      ENDIF
   ENDCASE
CASE mprompt = "Members    "
   SELECT 1
   USE members
   SET ORDER TO 1
   origf = "MEMBERS.DBF"
   DO CASE
   CASE bu_disk = "H"
      buf = "MEM_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
   CASE bu_disk = "F"
      buf = "A:\MEMBERS.DBF"
      DO bu_single WITH origf,buf
   ENDCASE
CASE aprompt = "acTivity  "
   SELECT 2
   SET ORDER TO 1
   origf = "ACTIVITY.DBF"
   DO CASE
   CASE bu_disk = "H"
      buf = "ACT_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
   CASE bu_disk = "F"
    ' buf = "A:\ACTIVITY.DBF"
      DO bu_single WITH origf,buf
   ENDCASE
CASE aprompt = "Curriculum"
   SELECT 3
   SET ORDER TO 1
   origf = "CURRICUL.DBF"
   DO CASE
   CASE bu_disk = "H"
      buf = "CUR_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
   CASE bu_disk = "F"
      buf = "A:\CURRICUL.DBF"
      DO bu_single WITH origf,buf
   ENDCASE
CASE aprompt = "Director  "
   SELECT 4
   origf = "DIRECTOR.DBF"
   DO CASE
   CASE bu_disk = "H"
      buf = "DIR_BU.DBF"
      DO dbf_bu WITH ROW,origf,buf
   CASE bu_disk = "F"
      buf = "A:\DIRECTOR.DBF"
      DO bu_single WITH origf,buf
   ENDCASE
ENDCASE
RELEASE POPUP bu
SET CURSOR ON
RETURN
*|==========================================================================
*|
*|      Procedure: BU_ALL
*|
*|      Called by: BU                    (procedure in NPS_BU.PRG)
*|
*|          Calls: DBF_BU                (procedure in NPS_PROC.PRG)
*|
*|           Uses: MEMBERS.DBF
*|
*|      CDX files: MEMBERS.CDX
*|
*|==========================================================================
PROCEDURE bu_all
PRIVATE ROW,origf,buf,flop
flop = .T.
ROW = 15
SET CURSOR OFF
SET COLOR TO W+/N
SELECT 1
USE members
SET ORDER TO 1
origf = "MEMBERS.DBF"
buf = "A:\MEMBERS.DBF"
DO dbf_bu WITH ROW,origf,buf,flop
SELECT 2
SET ORDER TO 1
origf = "ACTIVITY.DBF"
buf = "A:\ACTIVITY.DBF"
DO dbf_bu WITH ROW,origf,buf,flop
SELECT 3
SET ORDER TO 1
origf = "CURRICUL.DBF"
buf = "A:\CURRICUL.DBF"
DO dbf_bu WITH ROW,origf,buf,flop
SELECT 4
origf = "DIRECTOR.DBF"
buf = "A:\DIRECTOR.DBF"
DO dbf_bu WITH ROW,origf,buf,flop
SET CURSOR ON
RETURN
*|==========================================================================
*|
*|      Procedure: BU_SINGLE
*|
*|      Called by: BU                    (procedure in NPS_BU.PRG)
*|
*|          Calls: GETKEY                (procedure in NPS_PROC.PRG)
*|               : FLOPPY_SIZE           (procedure in NPS_BU.PRG)
*|               : DBF_BU                (procedure in NPS_PROC.PRG)
*|
*|           Uses: &BU.DBF
*|
*|==========================================================================
PROCEDURE bu_single
PARAMETERS orig,bu
PRIVATE no_size,filesize,dspace,ROW,flop
no_size = .F.
flop = .T.
ROW = 15
* ---Backup to floppy
@ 15,0 CLEAR 24,79
SET CURSOR OFF
@ 15,0 SAY "Insert a formatted backup floppy disk into the A: drive."
WAIT
SET CURSOR ON
IF FILE( bu )
   @ 15,0 CLEAR 24,79
   ?? CHR(7)
   @ 15,7 SAY bu+" exists from previous backup.  Replace it? (y/n)"
   DO getkey WITH choice, "YN"+returnkey
   IF choice = "Y"
      SET CURSOR OFF
      SET COLOR TO W+/N
      @ 15,0 CLEAR TO 24,79
      @ 15,0 SAY "Erasing previous backup ..."
      ERASE &bu
      WAIT "" TIMEOUT 1
      DO floppy_size WITH no_size
      IF ( ! no_size )
         DO dbf_bu WITH ROW,orig,bu,flop
      ELSE
         ?? CHR(7)
         WAIT [Insufficient disk space for "all" on floppy.  ]+;
            [Aborting backup ...] WINDOW TIMEOUT 2
      ENDIF
   ENDIF
   RETURN
ELSE
   SET CURSOR OFF
   SET COLOR TO W+/N
   DO floppy_size WITH no_size
   IF ( ! no_size )
      DO dbf_bu WITH ROW,orig,bu,flop
   ELSE
      ?? CHR(7)
      WAIT "Insufficient disk space on floppy disk. "+;
         "Aborting backup ..." WINDOW TIMEOUT 2
   ENDIF
ENDIF
RETURN
*|==========================================================================
*|
*|      Procedure: FLOPPY_SIZE
*|                                             )
*|      Called by: BU                    (procedure in NPS_BU.PRG)
*|               : BU_SINGLE             (procedure in NPS_BU.PRG)
*|
*|==========================================================================
PROCEDURE floppy_size
PARAMETER no_size,filesize,dspace,ALL
PRIVATE drive,file_size
* ---Check if sufficient diskspace available for backup to floppy
drive = SYS(5)
SET DEFAULT TO A:
file_size =  HEADER() + (RECSIZE() * RECCOUNT() + 1)
IF ( ALL )                  &&Add'l space req'd for activity,curricul,direct
*) or
   file_size = file_size + 10240
ENDIF
dspace = DISKSPACE()
IF dspace < file_size
   no_size = .T.
   SET DEFAULT TO &drive
   RETURN
ENDIF
SET DEFAULT TO &drive
RETURN
*|==========================================================================
*|
*|      Procedure: LIST_SELECT
*|
*|      Called by: NPS_BU.PRG
*|
*|          Calls: DEACTPOP              (procedure in NPS_PROC.PRG)
*|               : LIST_FILES            (procedure in NPS_BU.PRG)
*|
*|==========================================================================
PROCEDURE list_select
PRIVATE list_disk
DEFINE POPUP list_select FROM 6,49 TO 9,61 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select file(s)";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF list_select PROMPT "\<Hard disk  "
DEFINE BAR 2 OF list_select PROMPT "\<Floppy disk"
ON SELECTION POPUP list_select DO deactpop
aprompt = ""
@ 11,33 SAY "             " COLOR G+/N
ACTIVATE POPUP list_select
SET CURSOR OFF
DO CASE
CASE aprompt = "Hard disk  "
   list_disk = "H"
   DO list_files
CASE aprompt = "Floppy disk"
   list_disk = "F"
   DO list_files
ENDCASE
RELEASE POPUP list_select
SET CURSOR ON
RETURN
*|==========================================================================
*|
*|      Procedure: LIST_FILES
*|
*|      Called by: LIST_SELECT           (procedure in NPS_BU.PRG)
*|
*|          Calls: DEACTPOP              (procedure in NPS_PROC.PRG)
*|
*|==========================================================================
PROCEDURE list_files
DEFINE POPUP list_files FROM 6,49 TO 9,64 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select file(s)";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF list_files PROMPT "\<Database files"
DEFINE BAR 2 OF list_files PROMPT "\<All files     "
ON SELECTION POPUP list_files DO deactpop
aprompt = ""
@ 11,33 SAY "             " COLOR G+/N
ACTIVATE POPUP list_files
SET CURSOR OFF
DO CASE
CASE aprompt = "Database files"
   DO CASE
   CASE list_disk = "H"
      CLEAR
      @  0,60 SAY DATE() COLOR N/W
      @  0,68 SAY " " COLOR N/W
      DIRECTORY
      WAIT
   CASE list_disk = "F"
      @ 15,0 CLEAR TO 23,79
      @ 15,0 SAY "Insert floppy disk into the A: drive."
      WAIT
      CLEAR
      @  0,60 SAY DATE() COLOR N/W
      @  0,68 SAY " " COLOR N/W
      DIRECTORY ON A:
      WAIT
   ENDCASE
CASE aprompt = "All files     "
   DO CASE
   CASE list_disk = "H"
      CLEAR
      @  0,60 SAY DATE() COLOR N/W
      @  0,68 SAY " " COLOR N/W
      DIRECTORY LIKE *.*
      WAIT
   CASE list_disk = "F"
      @ 15,0 CLEAR TO 23,79
      @ 15,0 SAY "Insert floppy disk into the A: drive."
      WAIT
      CLEAR
      @  0,60 SAY DATE() COLOR N/W
      @  0,68 SAY " " COLOR N/W
```

```
         DIRECTORY ON A: LIKE *.*
         WAIT
      ENDCASE
ENDCASE
RELEASE POPUP list_files
SET CURSOR ON
RETURN
*|**********************************************************************
*|
*|        Procedure: FMT
*|
*|        Called by: NPS_BU.PRG
*|
*|            Calls: DEACTPOP            (procedure in NPS_PROC.PRG)
*|
*|**********************************************************************
PROCEDURE fmt
DEFINE POPUP fmt FROM 6,62 TO 10,76 TITLE "Select" ;
   MESSAGE "Scroll  to select [drive type] --> [diskette type]";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R
DEFINE BAR 1 OF fmt PROMPT "360K --> 360K"
DEFINE BAR 2 OF fmt PROMPT "1.2M --> 360K"
DEFINE BAR 3 OF fmt PROMPT "1.2M --> 1.2M"
ON SELECTION POPUP fmt DO deactpop
mprompt = ""
@ 11,33 SAY "              " COLOR G+/N
ACTIVATE POPUP fmt
DO CASE
CASE mprompt = "360K --> 360K"
   @ 15,0 CLEAR TO 24,79
   @ 15,0 SAY "Insert floppy disk into the A: drive."
   WAIT
   CLEAR
   RUN FORMAT A:
CASE mprompt = "1.2M --> 360K"
   @ 15,0 CLEAR TO 24,79
   @ 15,0 SAY "Insert floppy disk into the A: drive."
   WAIT
   CLEAR
   RUN FORMAT A: /4
CASE mprompt = "1.2M --> 1.2M"
   @ 15,0 CLEAR TO 24,79
   @ 15,0 SAY "Insert floppy disk into the A: drive."
   WAIT
   CLEAR
   RUN FORMAT A:
ENDCASE
RELEASE POPUP fmt
RETURN
*|**********************************************************************
*|
*|        Procedure: RESTOR
*|
*|        Called by: NPS_BU.PRG
*|
*|            Calls: GETKEY              (procedure in NPS_PROC.PRG)
*|                 , REST_SELECT         (procedure in NPS_BU.PRG)
*|
*|**********************************************************************
PROCEDURE RESTOR
* ---Restores backup database file(s)
choice = "n"
@ 11,33 SAY "              " COLOR G+/N
@ 12,0 CLEAR TO 24,79
?? CHR(7)
@ 14, 2 TO 20,77 COLOR W+/RB
@ 15, 3 SAY "  Restoring backup(s) replaces current database file data.  "
=> +;
   "Be sure that  " COLOR W+/RB
@ 15,25 SAY "replaces" COLOR R+/RB
@ 16, 3 SAY "  the backup(s) that you are restoring is/are the backup(s) "
=> +;
   "that you want " COLOR W+/RB
@ 17, 3 SAY [  or should use.  If unsure, use the menu option "List ]+;
   [files" to view the ] COLOR W+/RB
@ 18, 3 SAY "  database file creation dates before restoring.  Review "+;
   "the Help avail-  " COLOR W+/RB    "
@ 19, 3 SAY "  able for restoring and backing up by pressing <F1>.
=>             " COLOR W+/RB
@ 19,52 SAY "F1" COLOR GR+/RB
SET CURSOR ON
@ 22,12 SAY "Are you sure you want to restore backup(s) now ? (y/n)" ;
   COLOR W+/N
DO getkey WITH choice, "YN"
@ 12,0 CLEAR TO 24,79
IF choice = "Y"
   DO rest_select
ENDIF
RETURN
*|**********************************************************************
*|
*|        Procedure: REST_SELECT
*|
*|        Called by: RESTOR              (procedure in NPS_BU.PRG)
*|
*|            Calls: DEACTPOP            (procedure in NPS_PROC.PRG)
*|                 , REST_POP            (procedure in NPS_BU.PRG)
*|
*|**********************************************************************
PROCEDURE rest_select
PRIVATE rest_disk
DEFINE POPUP rest_select FROM 8,56 TO 11,68 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select source disk";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF rest_select PROMPT "\<Hard disk  "
DEFINE BAR 2 OF rest_select PROMPT "\<Floppy disk"
ON SELECTION POPUP rest_select DO deactpop
mprompt = ""
@ 11,33 SAY "              " COLOR G+/N
@ 12,0 CLEAR TO 24,79
ACTIVATE POPUP rest_select
SET CURSOR OFF
DO CASE
CASE mprompt = "Hard disk  "
   rest_disk = "H"
   DO rest_pop
CASE mprompt = "Floppy disk"
   rest_disk = "F"
   DO rest_pop
ENDCASE
RELEASE POPUP rest_select
SET CURSOR ON
RETURN
*|**********************************************************************
*|
*|        Procedure: REST_POP
*|
*|        Called by: REST_SELECT         (procedure in NPS_BU.PRG)
*|
*|            Calls: DEACTPOP            (procedure in NPS_PROC.PRG)
*|                 , DO_REST_HARD        (procedure in NPS_BU.PRG)
*|                 , DO_REST_FLOP        (procedure in NPS_BU.PRG)
*|                 , NPS_OPEN.PRG
*|
*|**********************************************************************
PROCEDURE rest_pop
PRIVATE origf, origndx, buf
DEFINE POPUP rest_pop FROM 6,56 TO 12,67 TITLE "Select" ;
   MESSAGE "Scroll  or press highlighted letter to select file(s)";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF rest_pop PROMPT "\<All        "
DEFINE BAR 2 OF rest_pop PROMPT "\<Members    "
DEFINE BAR 3 OF rest_pop PROMPT "ac\<Tivity   "
DEFINE BAR 4 OF rest_pop PROMPT "\<Curriculum"
DEFINE BAR 5 OF rest_pop PROMPT "\<Director   "
ON SELECTION POPUP rest_pop DO deactpop
mprompt = ""
@ 11,33 SAY "              " COLOR G+/N
SET CURSOR OFF
SET COLOR TO W+/N,N/W
@ 12,0 CLEAR TO 23,79
ACTIVATE POPUP rest_pop
DO CASE
CASE mprompt = "All        "
   DO CASE
   CASE rest_disk = "H"
      origf = "MEMBERS.DBF"
      orign = "MEMBERS.CDX"
      buf = "MEM_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
      origf = "ACTIVITY.DBF"
      orign = "ACTIVITY.CDX"
      buf = "ACT_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
      origf = "CURRICUL.DBF"
      orign = "CURRICUL.CDX"
      buf = "CUR_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
      origf = "DIRECTOR.DBF"
      orign = "DIRECTOR.CDX"     && dummy name (director.cdx doesn't exist
=> )
      buf = "DIR_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
   CASE rest_disk = "F"
      origf = "MEMBERS.DBF"
      orign = "MEMBERS.CDX"
      buf = "A:\MEMBERS.DBF"
      @ 15,0 CLEAR 23,79
      @ 15,0 SAY [Insert the "restore from" floppy disk into the A: drive.
=> ]
      WAIT
      DO do_rest_flop WITH origf,orign,buf
      origf = "ACTIVITY.DBF"
      orign = "ACTIVITY.CDX"
      buf = "A:\ACTIVITY.DBF"
      DO do_rest_flop WITH origf,orign,buf
      origf = "CURRICUL.DBF"
      orign = "CURRICUL.CDX"
      buf = "A:\CURRICUL.DBF"
      DO do_rest_flop WITH origf,orign,buf
      origf = "DIRECTOR.DBF"
      orign = "DIRECTOR.CDX"     && dummy name (director.cdx doesn't exist
=> )
      buf = "A:\DIRECTOR.DBF"
      DO do_rest_flop WITH origf,orign,buf
   ENDCASE
CASE mprompt = "Members    "
   origf = "MEMBERS.DBF"
   orign = "MEMBERS.CDX"
   DO CASE
   CASE rest_disk = "H"
      buf = "MEM_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
   CASE rest_disk = "F"
      buf = "A:\MEMBERS.DBF"
      @ 15,0 CLEAR 23,79
      @ 15,0 SAY [Insert the "restore from" floppy disk into the A: drive.
=> ]
      WAIT
      DO do_rest_flop WITH origf,orign,buf
   ENDCASE
CASE mprompt = "acTivity   "
   origf = "ACTIVITY.DBF"
   orign = "ACTIVITY.CDX"
   DO CASE
   CASE rest_disk = "H"
      buf = "ACT_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
   CASE rest_disk = "F"
      buf = "A:\ACTIVITY.DBF"
      @ 15,0 CLEAR 23,79
      @ 15,0 SAY [Insert the "restore from" floppy disk into the A: drive.
=> ]
      WAIT
      DO do_rest_flop WITH origf,orign,buf
   ENDCASE
CASE mprompt = "Curriculum"
   origf = "CURRICUL.DBF"
   orign = "CURRICUL.CDX"
   DO CASE
   CASE rest_disk = "H"
      buf = "CUR_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
   CASE rest_disk = "F"
      buf = "A:\CURRICUL.DBF"
      @ 15,0 CLEAR 23,79
      @ 15,0 SAY [Insert the "restore from" floppy disk into the A: drive.
=> ]
      WAIT
      DO do_rest_flop WITH origf,orign,buf
   ENDCASE
CASE mprompt = "Director   "
   origf = "DIRECTOR.DBF"
   orign = "DIRECTOR.CDX"        && dummy name (director.cdx doesn't exist
=> )
   DO CASE
   CASE rest_disk = "H"
      buf = "DIR_BU.DBF"
      DO do_rest_hard WITH origf,orign,buf
   CASE rest_disk = "F"
      buf = "A:\DIRECTOR.DBF"
      @ 15,0 CLEAR 23,79
      @ 15,0 SAY [Insert the "restore from" floppy disk into the A: drive.
=> ]
      WAIT
      DO do_rest_flop WITH origf,orign,buf
   ENDCASE
ENDCASE
RELEASE POPUP rest_pop
IF mbu = .F.
   DO nps_open
   SELECT &dbfarea
ENDIF
RETURN
```

```
*|*****************************************************************
*|
*|       Procedure: DO_REST_HARD
*|
*|       Called by: REST_POP              (procedure in NPS_BU.PRG)
*|
*|            Uses: &TEMPFILE.DBF
*|                : &BUFILE
*|                : &ORIGFILE.DBF
*|                : &ORIGNDX.DBF
*|
*|*****************************************************************
PROCEDURE do_rest_hard
PARAMETERS origfile, origndx, bufile
PRIVATE rcd_cnt,isfileok,tempfile
mbu = .F.
* ---Delete old temporary file if it exists
tempfile = "temp.dbf"
IF FILE( tempfile )
   ERASE &tempfile
ENDIF
CLOSE DATABASES
IF FILE( bufile )
   * ---Count number of records in backup file
   USE &bufile
   GOTO BOTTOM
   rcd_cnt = RECNO()
   @ 15,0 CLEAR TO 23,79
   @ 15,0 SAY [Restoring "] + origfile + [" from hard disk backup "] + ;
      bufile + [" ...]
   @ 16,0 SAY "Continuing ..."
   SET TALK ON
   COPY TO &tempfile
   SET TALK OFF
   WAIT "" TIMEOUT 1
   USE
   * ---Check if restoration is OK
   isfileok = .F.
   IF FILE( tempfile )
      USE &tempfile
      GOTO BOTTOM
      * ---Record count from original and backup must match
      isfileok = (rcd_cnt = RECNO())
      USE
   ENDIF
   IF ( ! isfileok)
      * ---Temp file was not created or has incorrect record count
      ?? CHR(7)
      WAIT "Restoration failed." WINDOW TIMEOUT 2
      IF FILE( tempfile )
         ERASE &tempfile
      ENDIF
      RETURN
   ENDIF
   ERASE &origfile
   RENAME &tempfile TO &origfile
   ERASE &origndx
ELSE
   @ 15,0 CLEAR TO 23,79
   @ 15,0 SAY [Hard disk backup "] + bufile + [" does not exist.]
   WAIT "" WINDOW TIMEOUT 2
ENDIF
RETURN
*|*****************************************************************
*|
*|       Procedure: DO_REST_FLOP
*|
*|       Called by: REST_POP              (procedure in NPS_BU.PRG)
*|
*|            Uses: &TEMPFILE.DBF
*|                : &BUFILE
*|                : &ORIGFILE.DBF
*|                : &ORIGNDX.DBF
*|
*|*****************************************************************
PROCEDURE do_rest_flop
PARAMETERS origfile, origndx, bufile
PRIVATE rcd_cnt,isfileok,tempfile
mbu = .F.
* ---Delete old temporary file if it exists
tempfile = "temp.dbf"
IF FILE( tempfile )
   ERASE &tempfile
ENDIF
CLOSE DATABASES
IF FILE( bufile )
   * ---Count number of records in backup file
   USE &bufile
   GOTO BOTTOM
   rcd_cnt = RECNO()
   @ 15,0 CLEAR TO 23,79
   @ 15,0 SAY [Restoring "] + origfile + [" from floppy disk backup "] + ;
      bufile + [" ...]
   @ 16,0 SAY "Continuing ..."
   SET TALK ON
   COPY TO &tempfile
   SET TALK OFF
   WAIT "" TIMEOUT 1
   USE
   * ---Check if restoration is OK
   isfileok = .F.
   IF FILE( tempfile )
      USE &tempfile
      GOTO BOTTOM
      * ---Record count from original and backup must match
      isfileok = (rcd_cnt = RECNO())
      USE
   ENDIF
   IF ( ! isfileok)
      * ---Temp file was not created or has incorrect record count
      ?? CHR(7)
      WAIT "Restoration failed." WINDOW TIMEOUT 2
      IF FILE( tempfile )
         ERASE &tempfile
      ENDIF
      RETURN
   ENDIF
   ERASE &origfile
   RENAME &tempfile TO &origfile
   ERASE &origndx
ELSE
   @ 15,0 CLEAR TO 23,79
   @ 15,0 SAY [Floppy disk backup "] + bufile + [" does not exist.]
   WAIT "" WINDOW TIMEOUT 2
ENDIF
RETURN
*: EOF: NPS_BU.PRG
```

```
*:*****************************************************************************
*:
*: Procedure file: C:\PTARS\NPS_PACK.PRG
*:
*:  Procs & Fncts: PACK
*:               : DOPACK
*:
*:         Set by: PACK_CHECK          (procedure in NPSDC.PRG)
*:               : PACK_POP            (procedure in NPS_UTIL.PRG)
*:               : WARNING             (procedure in NPS_UTIL.PRG)
*:
*:          Calls: GETKEY              (procedure in NPS_PROC.PRG)
*:               : PACK                (procedure in NPS_PACK.PRG)
*:
*:           Uses: &PACK_FIL
*:
*:      Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:*****************************************************************************
*PROGRAM: NPS_PACK (pack/sort database module)
PARAMETERS pack_fil,pack_ndx
PRIVATE tempf,isfileok,req_space,kbytes
tempf = "TEMP.DBF"
SET COLOR TO W+/N,N/W
IF warn = "Quit"
   @ 14,33 SAY "              " COLOR G+/N
   @ 15,0 CLEAR TO 24,79
   ?? CHR(7)
   @ 16, 2 TO 21,77 COLOR W+/RB
   @ 17, 3 SAY "  More than 10% of the database consists of records "+;
      "marked for deletion.  " COLOR W+/RB
   @ 18, 3 SAY "  The performance of the database will improve if these "+
*) ;
      "records are re-   " COLOR W+/RB
   @ 19, 3 SAY "  moved.  Be aware that packing the database permanently "
*) +;
      "removes records  " COLOR W+/RB
   @ 19,48 SAY "permanently" COLOR R+/RB
   @ 20, 3 SAY "  marked for deletion.  It is not undoable.
*)               " COLOR W+/RB
   SET CURSOR ON
   @ 23,18 SAY "Do you want to pack the database now? (y/n)" COLOR W+/N
   DO getkey WITH choice,"YN"+returnkey
   IF choice <> "Y"
      RETURN
   ENDIF
   origf = "MEMBERS.DBF"
   orign = "MEMBERS.CDX"
ENDIF
SET CURSOR OFF
SET COLOR TO W+/N
@ 24,0 SAY "WORKING ..." COLOR W/N*
CLOSE DATABASES
isfileok = .T.
* ---Check for sufficient disk space to pack and sort databases
IF pack_fil = "MEMBERS.DBF"
   USE &pack_fil
   req_space = 3*(HEADER() + ((RECCOUNT()+1)*RECSIZE()))
   IF DISKSPACE() < req_space
      CLEAR
      ?? CHR(7)
      kbytes = "(minimum "+SUBSTR(LTRIM(STR(req_space)),1,3)+"K req'd)."
      WAIT "Insufficient disk space "+kbytes+"  Database pack aborted." ;
         WINDOW TIMEOUT 3
      * ---Close the datafile and index.
      EXIT
   ENDIF
ENDIF
DO PACK WITH pack_fil,tempf,pack_ndx,isfileok
RETURN
*!*****************************************************************************
*!
*!       Procedure: PACK
*!
*!       Called by: NPS_PACK.PRG
*!
*!           Calls: DOPACK               (procedure in NPS_PACK.PRG)
*!
*!            Uses: &TEMPFILE.DBF*
*!                : &ORIGFILE.DBF
*!
*!*****************************************************************************
PROCEDURE PACK
PARAMETERS origfile,tempfile,origindex,isfileok
PRIVATE delrecs,undelrecs
@ 15,0 CLEAR TO 24,79
* ---Delete old temporary file if it exists.
IF FILE( tempfile )
   ERASE &tempfile
ENDIF
* ---Open original file
USE &origfile ORDER 1
COUNT FOR DELETED() TO delrecs
DO CASE
CASE NOT delrecs = 0
   @ 16,0 SAY [Removing all *DELETED* "] + origfile + [" records...]
   WAIT "Continuing ..." TIMEOUT 1
   * ---PACK file.
   @ 15,0 CLEAR TO 23,79
   @ 16,0 SAY [Copying "] + origfile + ["...]
   DO dopack
OTHERWISE
   @ 15,0 CLEAR TO 23,79
   @ 16,0 SAY ["] + origfile + [" already packed]
   WAIT "Continuing ..." TIMEOUT 1
   * ---SORT file.
   @ 15,0 CLEAR TO 23,79
   @ 16,0 SAY [Sorting "] + origfile + ["...]
   DO dopack
ENDCASE
RETURN
*!*****************************************************************************
*!
*!       Procedure: DOPACK
*!
*!       Called by: PACK                 (procedure in NPS_PACK.PRG)
*!
*!            Uses: &TEMPFILE.DBF
*!                : &ORIGFILE.DBF
*!                : &ORIGINDEX.DBF
*!
*!*****************************************************************************
PROCEDURE dopack
* ---Copy undeleted records in order of index.
* ---The new database will be "SORTed" in index order.
SET DELETED ON
SET TALK ON
COPY TO &tempfile
SET TALK OFF
SET DELETED OFF
* ---Turn off the index and count the number of undeleted records.
COUNT FOR .NOT. DELETED() TO undelrecs
* ---Close original file.
USE
* ---Was COPY TO temporary file successful?  .
isfileok = .F.
IF FILE( tempfile )
   USE &tempfile
   * ---The undeleted records in both files must match.
   IF EOF()
      isfileok = (undelrecs = 0)
   ELSE
      GOTO BOTTOM
      isfileok = (undelrecs = RECNO())
   ENDIF
   USE
ENDIF
IF .NOT. isfileok
   * ---"tempFILE" was not created or has an incorrect record count.
   ?? CHR(7)
   WAIT "The file could NOT be packed." WINDOW TIMEOUT 2
   RETURN
ENDIF
* ---Delete the original file and RENAME temporary to original
ERASE &origfile
RENAME &tempfile TO &origfile
* ---Delete original index and have calling program recreate it.
ERASE &origindex
RETURN
*: EOF: NPS_PACK.PRG
```

```
*:*********************************************************************
*:
*: Procedure file: C:\PTARS\NPS_PROC.PRG
*:
*:  Procs & Fncts: POPCAL
*:               : HLP
*:               : ABOUT
*:               : FUNC_LIST
*:               : BAR1
*:               : BAR2
*:               : BAR3
*:               : BAR4
*:               : VIEWBROW
*:               : DBF_BU
*:               : SAYREC
*:               : GETKEY
*:               : STATLINE
*:               : SAYEOF
*:               : SAYLINE
*:               : GOTOREC
*:               : DOGOTO
*:               : DOLOCATE
*:               : DOCONT
*:               : NPS_AREA
*:               : NPS_FILE
*:               : NPS_NDXS
*:               : NPS_SEEK
*:               : NPS_KEYS
*:               : NPS_FORMA1
*:               : NPS_FORMA2
*:               : NPS_FORMB
*:               : NPS_FORMC
*:               : NPS_SAYS
*:               : NPS_ASAY
*:               : NPS_BSAY
*:               : NPS_CSAY
*:               : NPS_STOR
*:               : NPS_ASTO
*:               : NPS_BSTO
*:               : NPS_CSTO
*:               : NPS_GETS
*:               : NPS_AGET
*:               : NPS_BGET
*:               : NPS_CGET
*:               : NPS_REPL
*:               : NPS_AREP
*:               : NPS_BREP
*:               : NPS_CREP
*:               : DEACTPOP
*:               : VAL_BRAN
*:               : VAL_PANO
*:               : VAL_UIC
*:               : VAL_CURR
*:               : PASSTRAN
*:               : PRNCHK
*:               : PRNSTOP
*:               : PRNPROB
*:               : SETPRN
*:
*:      Documented 01/28/92 at 09:00               FoxDoc  version 2.10
*:*********************************************************************
* PROGRAM: NPS_PROC.PRG (PROCEDURE file)
*!*********************************************************************
*!
*!      Procedure: POPCAL
*!
*!      Called by: ENV_SETUP          (procedure in NPSDC.PRG)
*!               : BAR2               (procedure in NPS_PROC.PRG)
*!
*!*********************************************************************
PROCEDURE popcal                                    &&Calendar'popup utility
PRIVATE i
DEFINE WINDOW cal FROM 1,0 TO 16,21 ;
   PANEL FOOTER "(T) (PgUp/Dn) (Esc)" ;
   TITLE "Calendar" COLOR ,,W/N,GR+/W
ACTIVATE WINDOW cal
ACTIVATE WINDOW calendar IN cal
MOVE WINDOW calendar TO -1,-1          && Center calendar in window
DO WHILE .T.
   i=INKEY(0,"H")                      && Get keystroke
   DO CASE
   CASE i=27                           && Esc (add i=13 for Enter key)
      EXIT
   CASE i=84 .OR. i=116                && 'T' for Today
      _DIARYDATE=DATE()
   CASE i=3                            && Page down
      _DIARYDATE=GOMONTH(_DIARYDATE,1)
   CASE i=18                           && Page up
      _DIARYDATE=GOMONTH(_DIARYDATE,-1)
   CASE i= 30                          && ^Page down
      _DIARYDATE=GOMONTH(_DIARYDATE,12)
   CASE i= 31                          && ^Page Up
      _DIARYDATE=GOMONTH(_DIARYDATE,-12)
   ENDCASE
ENDDO
RELEASE WINDOW cal
RETURN
*!*********************************************************************
*!
*!      Procedure: HLP
*!
*!      Called by: BAR1               (procedure in NPS_PROC.PRG)
*!               : MAIN_MENU          (procedure in NPSDC.PRG)
*!               : NPS_APPE.PRG
*!               : NPS_EDIT.PRG
*!               : NPS_BROW.PRG
*!               : NPS_RECA.PRG
*!               : NPS_REPO.PRG
*!               : NPS_UTIL.PRG
*!               : NPS_BU.PRG
*!               : DIRECTOR           (procedure in NPS_EDIT.PRG)
*!               : NPS_CREC           (procedure in NPS_RECA.PRG)
*!               : EDIT               (procedure in NPS_RECA.PRG)
*!               : NPS_CPAS           (procedure in NPS_UTIL.PRG)
*!
*!    Other Files: NPS_HLP
*!
*!*********************************************************************
PROCEDURE hlp
PARAMETER hlptopic
SET HELP ON
SET HELP TO nps_hlp
DO CASE
CASE hlptopic = "Main"
   SET TOPIC TO main
   HELP
CASE hlptopic = "Appe"
   SET TOPIC TO APPEND
   HELP
CASE hlptopic = "Bror"
   SET TOPIC TO view_recalls
   HELP
CASE hlptopic = "Dire"
   SET TOPIC TO DIRECTOR
   HELP
CASE hlptopic = "Edit"
   SET TOPIC TO EDIT
   HELP
CASE hlptopic = "Edir"
   SET TOPIC TO edit_recalls
   HELP
CASE hlptopic = "Func"
   SET TOPIC TO functions
   HELP
CASE hlptopic = "Repo"
   SET TOPIC TO REPORT
   HELP
CASE hlptopic = "Reca"
   SET TOPIC TO recalls
   HELP
CASE hlptopic = "Dele"
   SET TOPIC TO DELETE
   HELP
CASE hlptopic = "Pass"
   SET TOPIC TO password
   HELP
CASE hlptopic = "Buti"
   SET TOPIC TO backup_utils
   HELP
CASE hlptopic = "Help"
   SET TOPIC TO HELP
   HELP
CASE hlptopic = "Util"
   SET TOPIC TO utilities
   HELP
ENDCASE
SET HELP OFF
RETURN
*!*********************************************************************
*!
*!      Procedure: ABOUT
*!
*!      Called by: ENV_SETUP          (procedure in NPSDC.PRG)
*!               : BAR4               (procedure in NPS_PROC.PRG)
*!
*!          Calls: GETKEY             (procedure in NPS_PROC.PRG)
*!
*!*********************************************************************
PROCEDURE about
PRIVATE oldcur, keystrokes, choice
oldcur = SET ("cursor")
SET CURSOR OFF
DEFINE WINDOW about FROM 2,11 TO 15,68 ;
   DOUBLE NOFLOAT NOGROW IN SCREEN ;
   COLOR W+/RB,,W+/RB,W+/RB
ACTIVATE WINDOW about
keystrokes = "DN"
DO WHILE .T.
   @  1, 5 SAY "NPSDC PATIENT TRACKING AND RECALL SYSTEM (PTARS)"
   @  3, 7 SAY "Created by LCDR Timothy P. Steele, MSC, USN"
   @  4, 9 SAY "Naval Postgraduate School, Monterey, CA"
   @  5,22 SAY "January, 1992"
   @  7,24 SAY "          "
   @  8,24 SAY "          "
   @  9,24 SAY "          "
   @ 10,24 SAY "   1.0    "
   @ 11,14 SAY "<Done>                    <Next>"
   @ 11,15 SAY "D" COLOR GR+/RB
   @ 11,38 SAY "N" COLOR GR+/RB
   choice = "*"
   DO getkey WITH choice, keystrokes
   IF choice = "D"
      EXIT
   ENDIF
   CLEAR
   @ 1,18 SAY "SYSTEM ENVIRONMENT"
   @ 3,14 SAY "DOS version:          "+SUBSTR(OS(),6,4)
   @ 4,14 SAY "Files in CONFIG.SYS: "+SYS(2010)
   @ 5,14 SAY "Current directory:   "+SYS(5)+SYS(2003)
   @ 7,14 SAY "Processor:      "+SYS(17)
   @ 8,14 SAY "Video adapter: "+SYS(2006)
   @ 11,14 SAY "<Done>                    <Next>"
   @ 11,15 SAY "D" COLOR GR+/RB
   @ 11,38 SAY "N" COLOR GR+/RB
   DO getkey WITH choice, keystrokes
   IF choice = "D"
      EXIT
   ENDIF
   CLEAR
   @ 1,14 SAY "SYSTEM ENVIRONMENT (cont'd)"
   @ 3, 9 SAY "EMS memory available: "+SYS(23)+" Kbytes"
   @ 4, 9 SAY "Remaining DOS memory: "+LTRIM(STR(MEMORY()))+" Kbytes"
   @ 6, 9 SAY "Total disk "+SYS(5)+" size:       "+SYS(2020)+" bytes"
   @ 7, 9 SAY "Disk "+SYS(5)+" space remaining: "+LTRIM(STR(DISKSPACE()))+
*) ;
      " bytes"
   @ 11,14 SAY "<Done>                    <Next>"
   @ 11,15 SAY "D" COLOR GR+/RB
   @ 11,38 SAY "N" COLOR GR+/RB
   DO getkey WITH choice, keystrokes
   IF choice = "D"
      EXIT
   ENDIF
   CLEAR
ENDDO
DEACTIVATE WINDOW about
SET CURSOR &oldcur
RELEASE WINDOW about
RETURN
*!*********************************************************************
*!
*!      Procedure: FUNC_LIST
*!
*!      Called by: ENV_SETUP          (procedure in NPSDC.PRG)
*!
*!          Calls: BAR1               (procedure in NPS_PROC.PRG)
*!               : BAR2               (procedure in NPS_PROC.PRG)
*!               : BAR3               (procedure in NPS_PROC.PRG)
*!               : BAR4               (procedure in NPS_PROC.PRG)
*!
*!*********************************************************************
PROCEDURE func_list
DEFINE POPUP funcs FROM 1,0 TO 6,17 TITLE "Functions" ;
   MESSAGE "Scroll  or press highlighted letter to select function" ;
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF funcs PROMPT " \<Help F1       "
DEFINE BAR 2 OF funcs PROMPT " \<Calendar F2   "
DEFINE BAR 3 OF funcs PROMPT " \<Poptris F3    "
DEFINE BAR 4 OF funcs PROMPT " \<About PTARS F4"
ON SELECTION BAR 1 OF funcs DO bar1 IN nps_proc
ON SELECTION BAR 2 OF funcs DO bar2 IN nps_proc
ON SELECTION BAR 3 OF funcs DO bar3 IN nps_proc
ON SELECTION BAR 4 OF funcs DO bar4 IN nps_proc
ACTIVATE POPUP funcs
```

```
RETURN
*|********************************************************************
*|
*|       Procedure: BAR1
*|
*|       Called by: FUNC_LIST            (procedure in NPS_PROC.PRG)
*|
*|           Calls: HLP                  (procedure in NPS_PROC.PRG)
*|   .
*|********************************************************************
PROCEDURE bar1
DO hlp WITH "Func"
DEACTIVATE POPUP funcs
RETURN
*|********************************************************************
*|
*|       Procedure: BAR2
*|
*|       Called by: FUNC_LIST            (procedure in NPS_PROC.PRG)
*|
*|           Calls: POPCAL               (procedure in NPS_PROC.PRG)
*|
*|********************************************************************
PROCEDURE bar2
DO popcal
DEACTIVATE POPUP funcs
RETURN
*|********************************************************************
*|
*|       Procedure: BAR3
*|
*|       Called by: FUNC_LIST            (procedure in NPS_PROC.PRG)
*|
*|           Calls: POPTRIS.PRG
*|
*|********************************************************************
PROCEDURE bar3
DO poptris
DEACTIVATE POPUP funcs
RETURN
*|********************************************************************
*|
*|       Procedure: BAR4
*|
*|       Called by: FUNC_LIST            (procedure in NPS_PROC.PRG)
*|
*|           Calls: ABOUT                (procedure in NPS_PROC.PRG)
*|
*|********************************************************************
PROCEDURE bar4
DO about
DEACTIVATE POPUP funcs
RETURN
*|********************************************************************
*|
*|       Procedure: VIEWBROW
*|
*|       Called by: NPS_BROW.PRG
*|                : NPS_CREC             (procedure in NPS_RECA.PRG)
*|
*|           Calls: DEACTPOP             (procedure in NPS_PROC.PRG)
*|
*|********************************************************************
PROCEDURE viewbrow
* ---Change screen display mode
SET COLOR TO W/N,N/W
* ---Test for EGA or VGA to do View mode
IF ! ( "EGA" $ SYS( 2006 ) .OR. "VGA" $ SYS( 2006 ) )
   ?? CHR(7)
   WAIT "Sorry, (M)ode requires EGA/VGA video adapter." WINDOW TIMEOUT 2
   RETURN ( "" )
ENDIF
DEFINE POPUP mode FROM rowprompt - 7,35 TO rowprompt - 2,41 ;
   MESSAGE "Scroll  or press highlighted letter to select a view mode";
   COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF mode PROMPT "\<EGA25"
DEFINE BAR 2 OF mode PROMPT "EGA\<43"
DEFINE BAR 3 OF mode PROMPT "\<VGA25"
DEFINE BAR 4 OF mode PROMPT "VGA\<50"
ON SELECTION POPUP mode DO deactpop
mprompt = ""
ACTIVATE POPUP mode
DO CASE
CASE mprompt = "EGA25"
   CLEAR
   recnumtop = RECNO()
   SET DISPLAY TO EGA25
   rowbottom = 21
CASE mprompt = "EGA43"
   CLEAR
   recnumtop = RECNO()
   SET DISPLAY TO EGA43
   rowbottom = 39
CASE mprompt = "VGA25"
   IF ! ( "VGA" $ SYS( 2006 ) )
      ?? CHR(7)
      WAIT "Sorry, VGA video adapter required." WINDOW TIMEOUT 2
      RETURN
   ELSE
      CLEAR
      recnumtop = RECNO()
      SET DISPLAY TO VGA25
      rowbottom = 21
   ENDIF
CASE mprompt = "VGA50"
   IF ! ( "VGA" $ SYS( 2006 ) )
      ?? CHR(7)
      WAIT "Sorry, VGA video adapter required." WINDOW TIMEOUT 2
      RETURN
   ELSE
      CLEAR
      recnumtop = RECNO()
      SET DISPLAY TO VGA50
      rowbottom = 46
   ENDIF
ENDCASE
RELEASE POPUP mode
rowprompt = rowbottom + 2
skiprecs = rowbottom - rowtop + 1
RETURN
*|********************************************************************
*|
*|       Procedure: DBF_BU
*|
*|       Called by: QUITBU               (procedure in NPSDC.PRG)
*|                : NPS_RECA.PRG
*|                : BU                   (procedure in NPS_BU.PRG)
*|                : BU_ALL               (procedure in NPS_BU.PRG)
*|                : BU_SINGLE            (procedure in NPS_BU.PRG)
*|
*|            Uses: &TEMPFILE.DBF
*|                : &BUFILE
*|                : &BUFILE.DBF
*|
*|********************************************************************
PROCEDURE dbf_bu
* ---Creates session dbf file backups
PARAMETERS ROW,origfile,bufile,floppy
PRIVATE rcd_cnt,isfileok
tempfile = "temp.dbf"
* ---Count number of records in current database
GOTO BOTTOM
rcd_cnt = RECCOUNT()
GOTO TOP
@  ROW,0 CLEAR TO 23,79
* ---Delete old temporary file if it exists
IF FILE( tempfile )
   ERASE &tempfile
ENDIF
IF FILE( bufile )
   IF ( ! floppy)          &&if not backup to floppy disk
      RENAME &bufile TO &tempfile
   ENDIF
ENDIF
@ ROW,0 SAY [Backing up "] + origfile + [" to "] + bufile + [" ...]
@ ROW+1,0 SAY "Continuing ..."
SET TALK ON
COPY TO &bufile
SET TALK OFF
WAIT "" TIMEOUT 1
* ---Check if backup is OK
isfileok = .F.
IF FILE( bufile )
   SELECT 10
   USE &bufile
   GOTO BOTTOM
   * ---Record count from original and backup must match
   isfileok = (rcd_cnt = RECCOUNT())
   USE
ENDIF
IF ( ! isfileok)
   * ---Backup file was not created or has incorrect record count
   ?? CHR(7)
   WAIT "Backup file creation failed." WINDOW TIMEOUT 2
   IF FILE( bufile )
      ERASE &bufile
   ENDIF
   IF FILE( tempfile )
      IF ( ! floppy )
         RENAME &tempfile TO &bufile
      ENDIF
   ENDIF
ELSE
   ERASE &tempfile
   IF (bufile = "A:\MEMBERS.DBF")
      * --- Write members backup date to file nps_misc.dbf
      SELECT 5
      GO TOP
      REPLACE nps_misc.bu_date WITH DATE()
   ENDIF
ENDIF
RETURN
*|********************************************************************
*|
*|       Procedure: SAYREC
*|
*|       Called by: DOCONT               (procedure in NPS_PROC.PRG)
*|                : NPS_EDIT.PRG
*|
*|           Calls: STATLINE             (procedure in NPS_PROC.PRG)
*|                : NPS_SAYS             (procedure in NPS_PROC.PRG)
*|
*|********************************************************************
PROCEDURE sayrec
* ---"SayRec" is used by the EDIT program and PROCEDURE DoCONT.
DO statline WITH RECNO(),DELETED()
DO nps_says
RETURN
*|********************************************************************
*|
*|       Procedure: GETKEY
*|
*|       Called by: NPSDC.PRG
*|                : ABOUT                (procedure in NPS_PROC.PRG)
*|                : DOGOTO               (procedure in NPS_PROC.PRG)
*|                : DOCONT               (procedure in NPS_PROC.PRG)
*|                : NPS_NDXS             (procedure in NPS_PROC.PRG)
*|                : PRNCHK               (procedure in NPS_PROC.PRG)
*|                : NPS_INTR.PRG
*|                : PROMPTBU             (procedure in NPSDC.PRG)
*|                : NPS_APPE.PRG
*|                : NPS_EDIT.PRG
*|                : NPS_BROW.PRG
*|                : NPS_RECA.PRG
*|                : NPS_REPO.PRG
*|                : NPS_UTIL.PRG
*|                : NPS_BU.PRG
*|                : NPS_PACK.PRG
*|                : NPS_CREC             (procedure in NPS_RECA.PRG)
*|                : WARNING              (procedure in NPS_UTIL.PRG)
*|                : RESTOR               (procedure in NPS_BU.PRG)
*|                : BU                   (procedure in NPS_BU.PRG)
*|                : BU_SINGLE            (procedure in NPS_BU.PRG)
*|
*|********************************************************************
PROCEDURE getkey
PARAMETER choice,keychars
PRIVATE keycode
choice = "*"
DO WHILE .NOT. (choice $ keychars)
   keycode = INKEY()
   IF keycode > 0
      choice = UPPER(CHR(keycode))
   ENDIF
ENDDO
RETURN
*|********************************************************************
*|
*|       Procedure: STATLINE
*|
*|       Called by: SAYREC               (procedure in NPS_PROC.PRG)
*|                : NPS_APPE.PRG
*|
*|********************************************************************
PROCEDURE statline
PARAMETER recnum,isdeleted
SET COLOR TO W+/N,N/W
@ 0, 0 SAY SUBSTR( STR( recnum + 1000000,7 ),2 )
@ 0,29 SAY "(         )"
@ 0,30 SAY SUBSTR( dbfname,1,AT( ".",dbfname )-1 )
IF isdeleted
   @ 0,50 SAY "*DELETED*"
ELSE
   @ 0,50 SAY "         "
```

```
ENDIF
RETURN
*!******************************************************************
*!
*!      Procedure: SAYEOF
*!
*!      Called by: DOLOCATE           (procedure in NPS_PROC.PRG)
*!               : DOCONT             (procedure in NPS_PROC.PRG)
*!               : NPS_EDIT.PRG
*!
*!******************************************************************
PROCEDURE sayeof
PARAMETER ROW,oldrecnum
SET COLOR TO W+/N,N/W
@ ROW,0 CLEAR
?? CHR(7)
IF EOF()
   @ ROW,0 SAY "END-OF-FILE encountered"
ELSE
   @ ROW,0 SAY "BEGINNING-OF-FILE encountered"
ENDIF
WAIT
@ ROW,0 CLEAR
IF oldrecnum > 0
   GOTO oldrecnum
ENDIF
RETURN
*!******************************************************************
*!
*!      Procedure: SAYLINE
*!
*!      Called by: DOLOCATE           (procedure in NPS_PROC.PRG)
*!               : DOCONT             (procedure in NPS_PROC.PRG)
*!               : NPS_EDIT.PRG
*!               : NPS_BROW.PRG
*!               : NPS_CREC           (procedure in NPS_RECA.PRG)
*!
*!******************************************************************
PROCEDURE sayline
PARAMETER ROW,strg
SET COLOR TO W+/N,N/W
@ ROW,0 CLEAR
@ ROW,0 SAY strg
RETURN
*!******************************************************************
*!
*!      Procedure: GOTOREC
*!
*!      Called by: DOGOTO             (procedure in NPS_PROC.PRG)
*!
*!******************************************************************
PROCEDURE gotorec
PARAMETER ROW,recnum,lastrecnum
recnum = 0
SET COLOR TO W+/N,N/W
@ ROW,0 CLEAR
@ ROW+1,17 SAY "( 1 to "
@ ROW+1,24 SAY SUBSTR( STR( lastrecnum + 1000000,7 ),2 ) + " ) + (Return)"
SET CURSOR ON
@ ROW,0 SAY "Enter RECORD number" GET recnum;
   PICTURE "@Z 9999999" RANGE 0,lastrecnum
READ
@ ROW,0 CLEAR
IF recnum > 0
   GOTO recnum
ENDIF
RETURN
*!******************************************************************
*!
*!      Procedure: DOGOTO
*!
*!      Called by: NPS_EDIT.PRG
*!               : NPS_BROW.PRG
*!               : NPS_CREC           (procedure in NPS_RECA.PRG)
*!
*!          Calls: GETKEY             (procedure in NPS_PROC.PRG)
*!               : GOTOREC            (procedure in NPS_PROC.PRG)
*!
*!******************************************************************
PROCEDURE dogoto
PARAMETER ROW,recnum,lastrecnum
recnum = 0
SET COLOR TO W+/N,N/W
SET CURSOR OFF
@ ROW,0 CLEAR
@ ROW,0 SAY "GOTO:  (T)op  (B)ottom  (R)ecord#  (Return)"+SPACE(37) ;
   COLOR W+/BG
DO getkey WITH choice,"TBR"+returnkey
@ ROW,0 CLEAR
DO CASE
CASE choice = returnkey
   recnum = lastrecnum
   RETURN
CASE choice = "T"
   GOTO TOP
   recnum = RECNO()
CASE choice = "B"
   GOTO BOTTOM
   recnum = RECNO()
CASE choice = "R"
   DO gotorec WITH ROW,recnum,lastrecnum
ENDCASE
RETURN
*!******************************************************************
*!
*!      Procedure: DOLOCATE
*!
*!          Calls: SAYLINE            (procedure in NPS_PROC.PRG)
*!               : SAYEOF             (procedure in NPS_PROC.PRG)
*!               : DOCONT             (procedure in NPS_PROC.PRG)
*!
*!******************************************************************
PROCEDURE dolocate
PARAMETER ROW,expr
PRIVATE oldrecnum
oldrecnum = RECNO()
DO sayline WITH ROW,"Locating..."
LOCATE FOR &expr
IF EOF()
   DO sayeof WITH ROW,oldrecnum
ELSE
   @ ROW,0 CLEAR
   @ ROW,0 SAY "LOCATE FOR" GET expr
   CLEAR GETS
   DO docont WITH ROW
ENDIF
RETURN
*!******************************************************************
*!
*!      Procedure: DOCONT
*!
*!      Called by: DOLOCATE           (procedure in NPS_PROC.PRG)
*!
*!          Calls: SAYREC             (procedure in NPS_PROC.PRG)
*!               : SAYLINE            (procedure in NPS_PROC.PRG)
*!               : GETKEY             (procedure in NPS_PROC.PRG)
*!               : SAYEOF             (procedure in NPS_PROC.PRG)
*!
*!******************************************************************
PROCEDURE docont
PARAMETER ROW
PRIVATE oldrecnum
choice = "Y"
DO WHILE choice = "Y" .AND. .NOT. EOF()
   oldrecnum = RECNO()
   DO sayrec
   DO sayline WITH ROW+1,"Continue? (y/n)"
   DO getkey WITH choice,"YN"+returnkey
   @ ROW+1,0 CLEAR
   IF choice = "Y"
      CONTINUE
   ENDIF
ENDDO
IF EOF()
   DO sayeof WITH ROW,oldrecnum
ENDIF
RETURN
*!******************************************************************
*!
*!      Procedure: NPS_AREA
*!
*!      Called by: NPS_FILE           (procedure in NPS_PROC.PRG)
*!               : NPS_OPEN.PRG
*!               : NPS_CREC           (procedure in NPS_RECA.PRG)
*!
*!******************************************************************
PROCEDURE nps_area
SELECT &dbfarea
DO CASE
CASE dbfarea = "1"
   dbfname = "MEMBERS.DBF"
   ndxorder = "1"
CASE dbfarea = "2"
   dbfname = "ACTIVITY.DBF"
   ndxorder = "1"
CASE dbfarea = "3"
   dbfname = "CURRICUL.DBF"
   ndxorder = "1"
CASE dbfarea = "4"
   dbfname = "DIRECTOR.DBF"
   * no index needed
ENDCASE
lastrec = RECCOUNT()
RETURN
*!******************************************************************
*!
*!      Procedure: NPS_FILE
*!
*!      Called by: NPSDC.PRG
*!
*!          Calls: DEACTPOP           (procedure in NPS_PROC.PRG)
*!               : NPS_AREA           (procedure in NPS_PROC.PRG)
*!
*!******************************************************************
PROCEDURE nps_file
DEFINE POPUP dbfile FROM 9,54 TO 14,63 ;
   MESSAGE "Scroll  or press highlighted letter to select a database "+;
   "file" COLOR W+/BG,W+/BG,B/BG,B/BG,BG+/N,W+/R,GR+/BG
DEFINE BAR 1 OF dbfile PROMPT "\<MEMBERS "
DEFINE BAR 2 OF dbfile PROMPT "\<ACTIVITY"
DEFINE BAR 3 OF dbfile PROMPT "\<CURRICUL"
DEFINE BAR 4 OF dbfile PROMPT "\<DIRECTOR"
ON SELECTION POPUP dbfile DO deactpop
mprompt = ""
@ 14,33 SAY "              " COLOR G+/N
ACTIVATE POPUP dbfile
dbfchoice=dbfarea
DO CASE
CASE mprompt = "MEMBERS "
   dbfchoice = "1"
CASE mprompt = "ACTIVITY"
   dbfchoice = "2"
CASE mprompt = "CURRICUL"
   dbfchoice = "3"
CASE mprompt = "DIRECTOR"
   dbfchoice = "4"
ENDCASE
RELEASE POPUP dbfile
dbfarea = dbfchoice
DO nps_area
RETURN
*!******************************************************************
*!
*!      Procedure: NPS_NDXS
*!
*!      Called by: NPS_EDIT.PRG
*!               : NPS_BROW.PRG
*!               : NPS_CREC           (procedure in NPS_RECA.PRG)
*!
*!          Calls: GETKEY             (procedure in NPS_PROC.PRG)
*!
*!******************************************************************
PROCEDURE nps_ndxs
PARAMETER ROW,ndxchoice
DO CASE
CASE dbfarea = "1"
   SET COLOR TO W+/N,N/W
   SET CURSOR OFF
   @ ROW,0 CLEAR
   @ ROW,0 SAY "SEARCH BY:  1-MEMBER_SSN  2-MEMBER_NAME  (Return)"+;
      SPACE(31) COLOR W+/BG
   DO getkey WITH ndxchoice,"12"+returnkey
   SET CURSOR ON
   IF ndxchoice = returnkey
      RETURN
   ENDIF
   ndxorder = ndxchoice
   SET ORDER TO &ndxorder
CASE dbfarea = "2"
   * ---Only one index.
   ndxchoice = "1"
CASE dbfarea = "3"
   * ---Only one index.
   ndxchoice = "1"
ENDCASE
RETURN
*!******************************************************************
*!
*!      Procedure: NPS_SEEK
*!
*!      Called by: NPS_EDIT.PRG
*!               : NPS_BROW.PRG
*!               : NPS_CREC           (procedure in NPS_RECA.PRG)
*!
```

```
*|**************************************************************************
PROCEDURE nps_seek
PARAMETER ROW
PRIVATE expr
IF ndxorder = "0"
   RETURN
ENDIF
SET COLOR TO W+/N,N/W
DO CASE
CASE dbfarea = "1"
   @ ROW,0 CLEAR
   DO CASE
   CASE ndxorder = "1"
      mssn = SPACE(11)
      @ ROW,  0 SAY "Enter Ssn" GET mssn PICTURE "999-99-9999"
      READ
      expr = TRIM( mssn )
      IF "" <> expr
         SEEK expr
      ENDIF
   CASE ndxorder = "2"
      mlast_name = SPACE(23)
      mfirst_nam = SPACE(15)
      @ ROW,  0 SAY "Enter Last_name" GET mlast_name ;
         PICTURE "!AAAAAAAAAAAAAAAAAAAAAA"
      @ ROW+1,0 SAY "     First_name" GET mfirst_nam ;
         PICTURE "!AAAAAAAAAAAAAA"
      READ
      expr = TRIM( MLast_name )+TRIM( MFirst_nam )
      IF "" <> expr
         SEEK expr
      ENDIF
   ENDCASE
CA_E dbfarea = "2"
   @ ROW,0 CLEAR
   muicb = SPACE(5)
   @ ROW,  0 SAY "Enter UIC" GET muicb PICTURE "@!"
   READ
   expr = TRIM( muicb )
   IF "" <> expr      && If Expr not nul
      SEEK expr
   ENDIF
CASE dbfarea = "3"
   @ ROW,0 CLEAR
   mcurr_numc = SPACE(3)
   @ ROW,  0 SAY "Enter Curriculum #" GET mcurr_numc PICTURE "999"
   READ
   expr = TRIM( mcurr_numc )
   IF "" <> expr      && If Expr not nul
      SEEK expr
   ENDIF
ENDCASE
RETURN
*|**************************************************************************
*|
*|       Procedure: NPS_KEYS
*|
*|       Called by: NPS_APPE.PRG
*|                ; NPS_EDIT.PRG
*|
*|**************************************************************************
PROCEDURE nps_keys
PARAMETER expr,isblank,isunique
expr = ""
isblank = .F.
isunique = .F.
DO CASE
CASE dbfarea = "1"
   SET COLOR TO ,N/W
   @  3,16 GET mssn PICTURE "999-99-9999" ;
      VALID(11 = (LEN(ALLTRIM(mssn)))) ERROR "Valid SSN required."
   READ
   isblank = ("" = TRIM( mssn ))
   expr = mssn
   isunique = .T.
CASE dbfarea = "2"
   SET COLOR TO ,N/W
   @ 4,4 GET muicb PICTURE "@!" ;
      VALID(5 = (LEN(ALLTRIM(muicb)))) ERROR "UIC required.  Use 5 "+;
      "characters."
   READ
   isblank = ("" = TRIM( muicb ))
   expr = muicb
   isunique = .T.
CASE dbfarea = "3"
   SET COLOR TO ,N/W
   @ 4,4 GET mcurr_numc PICTURE "999" ;
      VALID(3 = (LEN(ALLTRIM(mcurr_numc)))) ;
      ERROR "3 digit Curriculum Number required."
   READ
   isblank = ("" = TRIM( mcurr_numc ))
   expr = mcurr_numc
   isunique = .T.
ENDCASE
RETURN
*|**************************************************************************
*|
*|       Procedure: NPS_FORMA1
*|
*|       Called by: NPS_FORMA2         (procedure in NPS_PROC.PRG)
*|                ; NPS_APPE.PRG
*|
*|**************************************************************************
PROCEDURE nps_forma1
CLEAR
@ 1,0 TO 14,79 DOUBLE COLOR B+/N
SET COLOR TO W+/N,N/W
@  0, 0 SAY SPACE(80)
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
@  0, 0 SAY "Record:"
@ promptrow ,0 SAY promptbar
@  2, 1 SAY "<F1> for Help"
@  2, 2 SAY "F1" COLOR GR+/N
SET COLOR TO W/N,N/W
@  3, 3 SAY "Member's SSN"
@  5, 3 SAY "Last Name "
@  5,40 SAY "First Name "
@  5,70 SAY "M.I. "
@  7, 3 SAY "Rank/Rate "
@  7,22 SAY "Service Branch "
@  7,45 SAY "Last T2 Exam "
@  7,70 SAY "Class "
@  8,58 SAY "MM/DD/YY"
@  9, 3 SAY "Pano Status"
@ 10, 0 SAY "                                      " COLOR B+/N
@ 10,40 SAY "                                      " COLOR B+/N
@ 12, 6 SAY "UIC "
RETURN
*|**************************************************************************
*|
*|       Procedure: NPS_FORMA2
*|
*|       Called by: NPS_EDIT.PRG
*|
*|           Calls: NPS_FORMA1         (procedure in NPS_PROC.PRG)
*|
*|**************************************************************************
PROCEDURE nps_forma2
DO nps_forma1
@ 1,0 TO 21,79 DOUBLE COLOR B+/N
@ 10, 0 SAY "                                      " COLOR B+/N
@ 10,40 SAY "                                      " COLOR B+/N
@ 14, 0 SAY "                                      " COLOR B+/N
@ 14,40 SAY "                                      " COLOR B+/N
@ 16,16 SAY "Dates of Previous Recall Letters Routed To Member"
@ 18, 8 SAY "Recall 1"
@ 18,27 SAY "Recall 2"
@ 18,46 SAY "Recall 3"
@ 18,65 SAY "Recall 4"
RETURN
*|**************************************************************************
*|
*|       Procedure: NPS_FORMB
*|
*|       Called by: NPS_APPE.PRG
*|                ; NPS_EDIT.PRG
*|
*|**************************************************************************
PROCEDURE nps_formb
CLEAR
@ 1,0 TO 9,79 DOUBLE COLOR B+/N
SET COLOR TO W+/N,N/W
@  0, 0 SAY SPACE(80)
@  0, 0 SAY "Record:"
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
@ promptrow ,0 SAY promptbar
@  2, 1 SAY "<F1> for Help"
@  2, 2 SAY "F1" COLOR GR+/N
SET COLOR TO W/N,N/W
@  3, 4 SAY "UIC"
@  3,15 SAY "Activity Name"
@  6, 4 SAY "Acronym"
@  6,20 SAY "Point of Contact"
RETURN
*|**************************************************************************
*|
*|       Procedure: NPS_FORMC
*|
*|       Called by: NPS_APPE.PRG
*|                ; NPS_EDIT.PRG
*|
*|**************************************************************************
PROCEDURE nps_formc
CLEAR
@ 1,0 TO 9,79 DOUBLE COLOR B+/N
SET COLOR TO W+/N,N/W
@  0, 0 SAY SPACE(80)
@  0, 0 SAY "Record:"
@  0,60 SAY DATE() COLOR N/W
@  0,68 SAY " " COLOR N/W
@ promptrow ,0 SAY promptbar
@  2, 1 SAY "<F1> for Help"
@  2, 2 SAY "F1" COLOR GR+/N
SET COLOR TO W/N,N/W
@  3, 4 SAY "Curriculum #"
@  3,23 SAY "Curriculum Name"
@  6, 4 SAY "Department"
@  7, 4 SAY "Code"
@  6,23 SAY "Phone #"
RETURN
*|**************************************************************************
*|
*|       Procedure: NPS_SAYS
*|
*|       Called by: SAYREC              (procedure in NPS_PROC.PRG)
*|
*|           Calls: NPS_ASAY            (procedure in NPS_PROC.PRG)
*|                ; NPS_BSAY            (procedure in NPS_PROC.PRG)
*|                ; NPS_CSAY            (procedure in NPS_PROC.PRG)
*|
*|**************************************************************************
PROCEDURE nps_says
DO CASE
CASE dbfarea = "1"
   DO nps_asay
CASE dbfarea = "2"
   DO nps_bsay
CASE dbfarea = "3"
   DO nps_csay
ENDCASE
RETURN
*|**************************************************************************
*|
*|       Procedure: NPS_ASAY
*|
*|       Called by: NPS_SAYS            (procedure in NPS_PROC.PRG)
*|
*|**************************************************************************
PROCEDURE nps_asay
* ---Using MEMBERS.DBF
SET COLOR TO ,N/W
@  3,16 GET members.ssn
@  5,13 GET members.last_name
@  5,51 GET members.first_name
@  5,75 GET members.mi
@  7,13 GET members.rank_rate
@  7,37 GET members.branch
@  7,58 GET members.last_t2
@  7,76 GET members.class
@  9,15 GET members.pano
@ 12,10 GET members.uic
IF members.uic = "31405"
   SET COLOR TO W/N
   @ 11,30 SAY "NPS Student"
   @ 12,30 SAY "Curriculum Number "
   @ 12,65 SAY "SMC"
   @ 13, 1 SAY SPACE(78) COLOR ,N/N
   SET COLOR TO ,N/W
   @ 12,48 GET members.curr_num
   @ 12,69 GET members.smc
ELSE
   @ 11,30 SAY SPACE(44) COLOR ,N/N
   @ 12,30 SAY "Department Code "+SPACE(28) COLOR W/N,N/N
   @ 12,46 GET members.smc
   @ 13, 1 SAY SPACE(78) COLOR ,N/N
ENDIF
@ 19, 8 GET members.recall_1   PICTURE "@D" COLOR ,W/N
@ 19,27 GET members.recall_2   PICTURE "@D" COLOR ,W/N
@ 19,46 GET members.recall_3   PICTURE "@D" COLOR ,W/N
@ 19,65 GET members.recall_4   PICTURE "@D" COLOR ,W/N
CLEAR GETS
RETURN
```

```
*|******************************************************************
*|
*|       Procedure: NPS_BSAY
*|
*|       Called by: NPS_SAYS            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_bsay
* ---Using ACTIVITY.DBF
SET COLOR TO ,N/W
@ 4, 4 GET activity.uic
@ 4,15 GET activity.act_name
@ 7, 4 GET activity.acronym
@ 7,20 GET activity.poc
CLEAR GETS
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_CSAY
*|
*|       Called by: NPS_SAYS            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_csay
* ---Using CURRICUL.DBF
SET COLOR TO ,N/W
@ 4, 4 GET curricul.curr_num
@ 4,23 GET curricul.curr_name
@ 7, 9 GET curricul.dept_code
@ 7,23 GET curricul.phone_no
CLEAR GETS
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_STOR
*|
*|       Called by: NPS_APPE.PRG
*|                , NPS_EDIT.PRG
*|
*|           Calls: NPS_ASTO            (procedure in NPS_PROC.PRG)
*|                , NPS_BSTO            (procedure in NPS_PROC.PRG)
*|                , NPS_CSTO            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_stor
DO CASE
CASE dbfarea = "1"
   DO nps_asto
CASE dbfarea = "2"
   DO nps_bsto
CASE dbfarea = "3"
   DO nps_csto
ENDCASE
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_ASTO
*|
*|       Called by: NPS_STOR            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_asto
* ---Using MEMBERS.DBF
* ---Initialize memvars with field contents.
STORE ssn        TO mssn
STORE last_name  TO mlast_name
STORE first_name TO mfirst_nam
STORE mi         TO mmi
STORE rank_rate  TO mrank_rate
STORE branch     TO mbranch
STORE last_t2    TO mlast_t2
STORE class      TO mclass
STORE pano       TO mpano
STORE uic        TO muic
STORE curr_num   TO mcurr_num
STORE smc        TO msmc
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_BSTO
*|
*|       Called by: NPS_STOR            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_bsto
* ---Using ACTIVITY.DBF
* ---Initialize memvars with field contents.
STORE activity.uic                       TO muicb
STORE activity.act_name         TO mact_name
STORE activity.acronym          TO macronym
STORE activity.poc                       TO mpoc
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_CSTO
*|
*|       Called by: NPS_STOR            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_csto
* ---Using CURRICUL.DBF
* ---Initialize memvars with field contents.
STORE curricul.curr_num         TO mcurr_numc
STORE curricul.curr_name        TO mcurr_name
STORE curricul.dept_code        TO mdept_code
STORE curricul.phone_no         TO mphone_no
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_GETS
*|
*|       Called by: NPS_APPE.PRG
*|                , NPS_EDIT.PRG
*|
*|           Calls: NPS_AGET            (procedure in NPS_PROC.PRG)
*|                , NPS_BGET            (procedure in NPS_PROC.PRG)
*|                , NPS_CGET            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_gets
DO CASE
CASE dbfarea = "1"
   DO nps_aget
CASE dbfarea = "2"
   DO nps_bget
CASE dbfarea = "3"
   DO nps_cget
ENDCASE
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_AGET
*|
*|       Called by: NPS_GETS            (procedure in NPS_PROC.PRG)
*|
*|           Calls: VAL_BRAN            (procedure in NPS_PROC.PRG)
*|                , VAL_PANO            (procedure in NPS_PROC.PRG)
*|                , VAL_UIC             (procedure in NPS_PROC.PRG)
*|                , VAL_CURR            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_aget
* ---Using MEMBERS.DBF
SET COLOR TO W/N,N/W
@  5,13 GET mlast_name PICTURE "!AAAAAAAAAAAAAAAAAAA" ;
   VALID(NOT EMPTY(ALLTRIM( mlast_name))) ;
   ERROR "Last name required." FUNCTION "A"
@  5,51 GET mfirst_nam  PICTURE "!AAAAAAAAAAAAA" ;
   VALID (NOT EMPTY(ALLTRIM( mfirst_name))) ;
   ERROR "First name required." FUNCTION "A"
@  5,75 GET mmi         PICTURE "!" FUNCTION "A"
@  7,13 GET mrank_rate  PICTURE "@!" ;
   VALID (NOT EMPTY(ALLTRIM( mrank_rate))) ;
   ERROR "Rank or rate required."
@  7,37 GET mbranch     PICTURE "@!" VALID(val_bran(mbranch)) ;
   ERROR "Service Branch required." FUNCTION "A"
@  7,58 GET mlast_t2   PICTURE "D" VALID((NOT EMPTY(mlast_t2)) .AND. ;
   (mlast_t2 <= DATE()) .AND. (mlast_t2 >= CTOD("01/01/89"))) ;
   ERROR "Valid Last_T2 date required."
@  7,76 GET mclass      PICTURE "9" RANGE 1,4 VALID(NOT EMPTY(mclass)) ;
   ERROR "Valid Class required." FUNCTION "Z"
@  9,15 GET mpano       PICTURE "@!" VALID(val_pano(mpano)) ;
   ERROR "Pano Status required." FUNCTION "A"
@ 12,10 GET muic        PICTURE "@!" VALID(val_uic(muic)) ;
   ERROR "UIC required.  Append new UIC to ACTIVITY.DBF if N/A."
READ
IF LASTKEY() = 27
   RETURN
ENDIF
DO CASE
CASE muic <> "31405"
   @ 12,16 CLEAR TO 13,78
   @ 12,30 SAY "Department Code "+SPACE(20) COLOR W/N,N/N
   @ 13,30 SAY "(Leave blank if NA)" COLOR RB+/N
   @ 12,46 GET msmc PICTURE "NNNN"      &&Is locater code for non-student
+) s
   READ
CASE muic = "31405"
   @ 12,16 CLEAR TO 13,78
   @ 11,30 SAY "NPS Student"
   @ 12,30 SAY "Curriculum Number "
   @ 12,65 SAY "SMC"
   @ 12,69 SAY "    " COLOR N/W
   @ 12,69 SAY smc COLOR N/W
   @ 12,48 GET mcurr_num   PICTURE "999" VALID(val_curr( mcurr_num )) ;
      ERROR "If NPS student, Curriculum Number required."
   READ
   IF LASTKEY() = 27
      RETURN
   ENDIF
   @ 12,69 GET msmc PICTURE "NNNN"
   @ 13,55 SAY "(Leave blank if unknown)" COLOR RB+/N
   READ
ENDCASE
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_BGET
*|
*|       Called by: NPS_GETS            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_bget
* ---Using ACTIVITY.DBF
SET COLOR TO W/N,N/W
@ 4,15 GET mact_name               PICTURE "@!" ;
   VALID (NOT EMPTY(ALLTRIM( mact_name ))) ;
   ERROR "Activity name required."
@ 7, 4 GET macronym                PICTURE "@!"
@ 7,20 GET mpoc ;
   VALID (NOT EMPTY(ALLTRIM( mpoc ))) ;
   ERROR "Point of Contact required."
READ
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_CGET
*|
*|       Called by: NPS_GETS            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_cget
* ---Using CURRICUL.DBF
@ 4,23 GET mcurr_name ;
   VALID (NOT EMPTY(ALLTRIM( mcurr_name ))) ;
   ERROR "Curriculum name required."
@ 7, 9 GET mdept_code   PICTURE "@!"
@ 7,23 GET mphone_no              PICTURE "9999"
READ
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_REPL
*|
*|       Called by: NPS_APPE.PRG
*|                , NPS_EDIT.PRG
*|
*|           Calls: NPS_AREP            (procedure in NPS_PROC.PRG)
*|                , NPS_BREP            (procedure in NPS_PROC.PRG)
*|                , NPS_CREP            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_repl                                 &&Replaces/appends input
+) data to database
DO CASE
CASE dbfarea = "1"
   DO nps_arep
CASE dbfarea = "2"
   DO nps_brep
CASE dbfarea = "3"
   DO nps_crep
ENDCASE
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_AREP
*|
*|       Called by: NPS_REPL            (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_arep
* ---Using MEMBERS.DBF
IF .NOT. EOF()
   * ---Replace only if there is an available record.
   ON ERROR RETURN .F.
   REPLACE;
      ssn        WITH mssn,;
      last_name  WITH mlast_name,;
      first_name WITH mfirst_nam,;
      mi         WITH mmi,;
      rank_rate  WITH mrank_rate,;
      branch     WITH mbranch,;
      last_t2    WITH mlast_t2,;
      class      WITH mclass,;
      pano       WITH mpano,;
      uic        WITH muic,;
      curr_num   WITH mcurr_num,;
      ssc        WITH mssc
ENDIF
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_BREP
*|
*|       Called by: NPS_REPL           (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_brep
* ---Using ACTIVITY.DBF
IF .NOT. EOF()
   * ---Replace only if there is an available record.
   ON ERROR RETURN .F.
   REPLACE ;
      uic                                  WITH muicb,;
      act_name                        WITH mact_name,;
      acronym                         WITH macronym,;
      poc                                  WITH mpoc
ENDIF
RETURN
*|******************************************************************
*|
*|       Procedure: NPS_CREP
*|
*|       Called by: NPS_REPL           (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
PROCEDURE nps_crep
* ---Using CURRICUL.DBF
IF .NOT. EOF()
   * ---Replace only if there is an available record.
   ON ERROR RETURN .F.
   REPLACE ;
      curricul.curr_num                   WITH mcurr_numc,;
      curr_name          WITH mcurr_name,;
      dept_code          WITH mdept_code,;
      phone_no                    WITH mphone_no
ENDIF
RETURN
*|******************************************************************
*|
*|       Procedure: DEACTPOP
*|
*|       Called by: VIEWBROW           (procedure in NPS_PROC.PRG)
*|                , NPS_FILE           (procedure in NPS_PROC.PRG)
*|                , SET_POP            (procedure in NPSDC.PRG)
*|                , REC_LISTS          (procedure in NPS_RECA.PRG)
*|                , LISTPOP            (procedure in NPS_REPO.PRG)
*|                , CLASSPOP           (procedure in NPS_REPO.PRG)
*|                , PANPOP             (procedure in NPS_REPO.PRG)
*|                , PACK_POP           (procedure in NPS_UTIL.PRG)
*|                , DATE_TIME          (procedure in NPS_UTIL.PRG)
*|                , PRNDEFAULT         (procedure in NPS_UTIL.PRG)
*|                , BU_SELECT          (procedure in NPS_BU.PRG)
*|                , LIST_SELECT        (procedure in NPS_BU.PRG)
*|                , FMT                (procedure in NPS_BU.PRG)
*|                , BU                 (procedure in NPS_BU.PRG)
*|                , LIST_FILES         (procedure in NPS_BU.PRG)
*|                , REST_SELECT        (procedure in NPS_BU.PRG)
*|                , REST_POP           (procedure in NPS_BU.PRG)
*|
*|******************************************************************
PROCEDURE deactpop
STORE PROMPT() TO mprompt
DEACTIVATE POPUP
RETURN
*|******************************************************************
*|
*|       Procedure: VAL_BRAN
*|
*|       Called by: NPS_AGET           (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
FUNCTION val_bran
* ---Validates MEMBERS.BRANCH
PARAMETER branid
ON KEY LABEL esc RETURN .F.
DO WHILE NOT (mbranch="USN ".OR.mbranch="USA ".OR. ;
      mbranch="USAF".OR.mbranch="USCG".OR.mbranch="USMC")
   mprompt = ""
   ACTIVATE POPUP branch
   mbranch = mprompt
   @ 7,37 GET mbranch
   CLEAR GETS
   RETURN .T.
ENDDO
ON KEY LABEL esc
*|******************************************************************
*|
*|       Procedure: VAL_PANO
*|
*|       Called by: NPS_AGET           (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
FUNCTION val_pano
* ---Validates MEMBERS.Pano
PARAMETER panocolor
ON KEY LABEL esc RETURN .F.
DO WHILE NOT (mpano="GRN".OR.mpano="RED".OR.mpano="YLW")
   mprompt = ""
   ACTIVATE POPUP pano
   mpano = SUBSTR(mprompt,1,3)
   @ 9,15 GET mpano
   CLEAR GETS
   RETURN .T.
ENDDO
ON KEY LABEL esc
*|******************************************************************
*|
*|       Procedure: VAL_UIC
*|
*|       Called by: NPS_AGET           (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
FUNCTION val_uic
* ---Validates MEMBERS.UIC against ACTIVITY.UIC
PARAMETER uicid
PRIVATE recnum
recnum = RECNO()
SELECT activity
SEEK uicid
IF FOUND()
   SELECT members
   IF NOT APPE = .T.
      GO RECORD recnum
   ENDIF
   RETURN .T.
ENDIF
DO WHILE .T.
   DO WHILE NOT FOUND()
      mprompt = ""
      ACTIVATE POPUP popuic
      SEEK SUBSTR(mprompt,1,5)
   ENDDO
   ON KEY = 315 RETURN .F.
   SELECT members
   IF NOT APPE = .T.
      GO RECORD recnum
   ENDIF
   muic = SUBSTR(mprompt,1,5)
   @ 12,10 GET muic
   CLEAR GETS
   RETURN .T.
ENDDO
ON KEY = 315
*|******************************************************************
*|
*|       Procedure: VAL_CURR
*|
*|       Called by: NPS_AGET           (procedure in NPS_PROC.PRG)
*|
*|******************************************************************
FUNCTION val_curr
* ---Validates MEMBERS.CURR_NUM against CURRICUL.CURR_NUM
PARAMETER currnum
PRIVATE recnum
recnum = RECNO()
SELECT curricul
SEEK currnum
IF FOUND()
   SELECT members
   IF NOT APPE = .T.
      GO RECORD recnum
   ENDIF
   RETURN .T.
ENDIF
DO WHILE .T.
   DO WHILE NOT FOUND()
      mprompt = ""
      ACTIVATE POPUP popcurr
      SEEK SUBSTR(mprompt,1,3)
   ENDDO
   ON KEY = 315 RETURN .F.
   SELECT members
   IF NOT APPE = .T.
      GO RECORD recnum
   ENDIF
   mcurr_num = SUBSTR(mprompt,1,3)
   @ 12,48 GET mcurr_num
   CLEAR GETS
   RETURN .T.
ENDDO
ON KEY = 315
*|******************************************************************
*|
*|       Procedure: PASSTRAN
*|
*|       Called by: NPS_INTR.PRG
*|                , NPS_CPAS           (procedure in NPS_UTIL.PRG)
*|
*|******************************************************************
PROCEDURE passtran
* (password encryption/translation module)
PARAMETERS pass_str,word_lngth,ENCRYPT
PRIVATE wrk_string,wrk_pass,wrk_char,offset
IF ENCRYPT                                                         &
 => & Procedure called to encrypt string
   IF !EMPTY(pass_str)
      wrk_string = PADR(UPPER(pass_str),word_lngth)
      STORE '' TO wrk_pass, wrk_char
      *---Generate a random ascii code for the offset character
      offset = MOD(RAND(-1)*10000,255)+1
      wrk_pass = CHR(offset)                        && Store offset character
 =>  as first
      && character in the translated string
      FOR i=1 TO word_lngth                                 && Parse characte
 => rs from input string
         wrk_char = SUBSTR(wrk_string,i,1)
         *---The character is encrypted as follows:
         *  1: Take its ASCII value
         *  2: Add value of the offset
         *  3: MOD this new number by 255 to make sure it is a legal
         *     ASCII value
         *  4: Convert it back into an ASCII character
         wrk_char = CHR(MOD(ASC(wrk_char)+offset,255)+1)
         wrk_pass = wrk_pass+wrk_char
         *---The ASCII value of this encrypted character is then used as t
 => he
         *---offset for the next character.  In this way, the offset value
         *---changes for each character, making it harder to decrypt.
         offset = ASC(wrk_char)
      ENDFOR
   ELSE                                                            &
 => & If string to be encrypted is blank
      wrk_pass = SPACE(word_lngth+1)     && return blanks as the encrypted
 =>  value.
   ENDIF
ELSE                                                               &
 => & Decryption section...
   IF !EMPTY(pass_str)
      STORE '' TO wrk_pass, wrk_char
      wrk_string = SUBSTR(pass_str,2)
      word_lngth = LEN(pass_str)
      done = .F.
      *---Determine offset value of first character
      offset = ASC(pass_str)
      *---Note first character is skipped because it is only the "offset
      *   character" for the first character of the actual string
      FOR i=2 TO word_lngth
         wrk_char = SUBSTR(pass_str,i,1)
         IF ASC(wrk_char) <= offset
            wrk_pass = wrk_pass + CHR(255+ASC(wrk_char)-offset-1)
         ELSE
            wrk_pass = wrk_pass + CHR(ASC(wrk_char)-offset-1)
         ENDIF
         offset = ASC(wrk_char)
      ENDFOR
   ELSE
      wrk_pass = SPACE(word_lngth)
   ENDIF
ENDIF
RETURN wrk_pass
*|*******************************************************************
*|
*|      Procedure: PRNCHK
*|
*|      Called by: NPS_RECA.PRG
*|               : REC_LISTS           (procedure in NPS_RECA.PRG)
*|               : PRNRPT              (procedure in NPS_REPO.PRG)
*|
*|          Calls: GETKEY              (procedure in NPS_PROC.PRG)
*|
*|*******************************************************************
PROCEDURE prnchk                            && Printer safety checklist
CLEAR
SET COLOR TO W+/N,N/W
@ 7,24 TO 14,55 COLOR W+/RB
@ 8,25 SAY "  Check the printer ...       " COLOR W+/RB
@ 9,25 SAY "    - is the power on?        " COLOR W+/RB
@ 10,25 SAY "    - is it on-line?          " COLOR W+/RB
@ 11,25 SAY "    - is it set to 12-pitch?  " COLOR W+/RB
@ 12,25 SAY "    - is there enough paper?  " COLOR W+/RB
@ 13,25 SAY "    - is the ribbon OK?       " COLOR W+/RB
SET CURSOR ON
@ 16,25 SAY "Continue with print job? (y/n)"
choice = "n"
DO getkey WITH choice, "YN"
prnstat = choice
RETURN
*|*******************************************************************
*|
*|      Procedure: PRNSTOP
*|
*|      Called by: NPS_RECA.PRG
*|               : RECPRN              (procedure in NPS_RECA.PRG)
*|               : PRNRPT              (procedure in NPS_REPO.PRG)
*|
*|*******************************************************************
PROCEDURE prnstop                                  && Stops printing when es
*) cape pressed
@ 10,10 TO 11,79 CLEAR
IF mprint = .T.
   STORE .F. TO mprint
   EJECT
   WAIT "Print job aborted." WINDOW TIMEOUT 2
ENDIF
RETURN
*|*******************************************************************
*|
*|      Procedure: PRNPROB
*|
*|      Called by: NPS_RECA.PRG
*|               : REC_LISTS           (procedure in NPS_RECA.PRG)
*|               : RECPRN              (procedure in NPS_RECA.PRG)
*|               : PRNRPT              (procedure in NPS_REPO.PRG)
*|
*|*******************************************************************
PROCEDURE prnprob
CLEAR
mprint = .F.
IF ERROR() = 125
   ?? CHR(7)
   WAIT "Printer problem.  Check the printer." WINDOW TIMEOUT 1
ENDIF
RETURN
*|*******************************************************************
*|
*|      Procedure: SETPRN
*|
*|      Called by: GORECALLS           (procedure in NPS_RECA.PRG)
*|               : PRNRPT              (procedure in NPS_REPO.PRG)
*|
*|*******************************************************************
PROCEDURE setprn
IF nps_misc.printer = "IBM Proprinter    "
   ??? CHR(27) + CHR(58)
ELSE
   ??? CHR(27) + "M"
ENDIF
RETURN
*: EOF: NPS_PROC.PRG
```

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3. LCDR Timothy P. Steele, MSC, USN 12339 Coleraine Court Reston, Virginia 22091	2
4. Prof. Hemant K. Bhargava, Code AS/BH Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
5. Prof. Dani Zweig, Code AS/ZG Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
6. Director, Branch Dental Clinic Naval Postgraduate School Monterey, California 93943-5100	1
7. Prof. Tung X. Bui Code AS/BD Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1