Table of Contents

Contract No.: 53-3198-5-51
MPR Reference No.: 7666-307

FSPOS DATA ACCESS SYSTEM

SYSTEM ADMINISTRATOR'S MANUAL

Version 2.0

May 15, 1989

Authors:

David Edson
Richard Lieberman
Esther Miller
Linda Wray

Prepared for:

Food and Nutrition Service
U.S. Department of Agriculture
3101 Park Center Drive
Alexandria, Virginia 22302

Prepared by:

Mathematica Policy Research, Inc.
600 Maryland Avenue, S.W.
Suite 550
Washington, D.C. 20024-2512

TABLE OF CONTENTS

I. INTRODUCTION

This document provides the information required to maintain and enhance the Food Stamp Program Operations Study Data Access System. The individual responsible for these activities, the System Administrator, should have a good working knowledge of dBase III Plus. The System Administrator must be familiar with the dBase programming language and must understand the concepts of data files, file structures and index files. In addition, the System Administrator must be familiar with MS DOS and microcomputers. Additional experience with communications software is necessary for individuals wishing to transfer data from the mainframe to the Access System. Individuals requiring additional training or assistance in any of these items are urged to consult the software manuals or any of the dBase or DOS-related publications available at bookstores.

The System Administrator should also be familiar with the content and operation of the Access System as described in the User's Guide.

This manual begins with an overview of the Access System. The basic structure of the system and the concept of the Data Dictionary are discussed. A brief description of the programs, data files, and index files follows.

The next chapter provides instructions for adding variables to existing files, adding new files in either new or existing subject areas, modifying keywords and variable descriptions, and

II. SYSTEM OVERVIEW

This chapter presents an overview of the Food and Nutrition Service Food Stamp Program Operations Data Access System. The chapter begins with a discussion of a concept central to the design and structure of the system -- the Data Dictionary. A graphical overview of the programs which comprise the system and their relationship to one another is then presented. The chapter concludes with brief descriptions of the programs, data files, and index files used by the system.

A. The Cross-Reference and Data Dictionary Files

The heart of the Access System is a pair of data files -- the Cross Reference File and the Data Dictionary. The Cross Reference file is a dBase file containing information about FSPOS data elements referenced in the FSPOS published reports. The file is intended to serve as an aid to researchers using the published FSPOS material. The Cross Reference also identifies data elements which are included in the Data Access system.

The Data Dictionary is derived from the Cross Reference file. It also is a dBase data file and contains information on the name, definition, location, and characteristics of each data element or variable which may be accessed through the Data Access system. This information is used by the various dBase programs in the system to retrieve and display the data itself.

An example of the use of the Data Dictionary occurs in the Data Display Module. When the user supplies a variable ID number or variable name, the data display programs refer to the Data Dictionary to obtain the variable description and subject area. This information, along with the ID number and name, is displayed on the user's screen. The system also uses the Data Dictionary to retrieve the name of the data file containing the variable selected by the user. This information is then used to retrieve the data from the appropriate data file for display to the user. The structure of the Data Dictionary file is

CHART II-1

Structure of the Data Dictionary File

Field	Field Name	Type	Width	Decimals	Description
1	FIELD_NAME	Character	10		Name of variable
2	FIELD_TYPE	Character	1		Type of variable (character or numeric)
3	FIELD_LEN	Numeric	3	0	Width of variable
4	FIELD_DEC	Numeric	3	0	Number of implied decimals in variable
5	DESCRIP	Character	40		Short description
6	KEYWORD	Character	80		Long description
7	MODULE	Numeric	2	0	Module number (for use in preparing standard reports)
8	SUBJECT	Character	3		Abbreviation for subject area
9	FILENAME	Character	8		Name of .dbf file containing variable
10	IDNUM	Numeric	5	0	Unique ID number assigned to variable
11	CONSTRUCT	Logical	1		Logical variable denoting whether variable is constructed (derived from other variables)
12	YESNO	Logical	1		Logical variable denoting whether variable may take only Yes, No, or missing values

Note: In dBase III Plus, variables or fields may be character, numeric, memo, date or logical type fields. Refer to the dBase manual for a complete discussion of field types.

shown in Chart II-1. Instructions for adding records to the Data Dictionary are included in Chapter III.

B. System Map

Chart II-1 is a system map displaying the relationship between the programs which comprise the Access System. The map is organized in columns, with lines linking programs which are used by other programs. Programs use other programs listed to their immediate right. For instance, the program FNSMENU0 in the second column from the left uses the programs FNSSCRN and FNSCENTR.

C. Description of Programs

The Access System contains approximately fifty separate dBase programs containing over 5,000 lines of dBase instructions which read, manipulate, and display the Program Operations Study data. A brief description of the functions performed by each of the programs is contained in Table II-2. Complete listings of the programs are included in Chapter IV.

D. Description of Data Files

Data from the Food Stamp Program Operations Study are contained in approximately thirty .dbf files. Approximately twenty five additional data files are used by the system for intermediate storage, data display, and maintenance purposes. A brief description of the data files are contained in Tables II-3 and II-4. Listings of the file structures are contained in Chapter V.

CHART II-1

RELATIONSHIP BETWEEN DATA ACCESS SYSTEM PROGRAMS

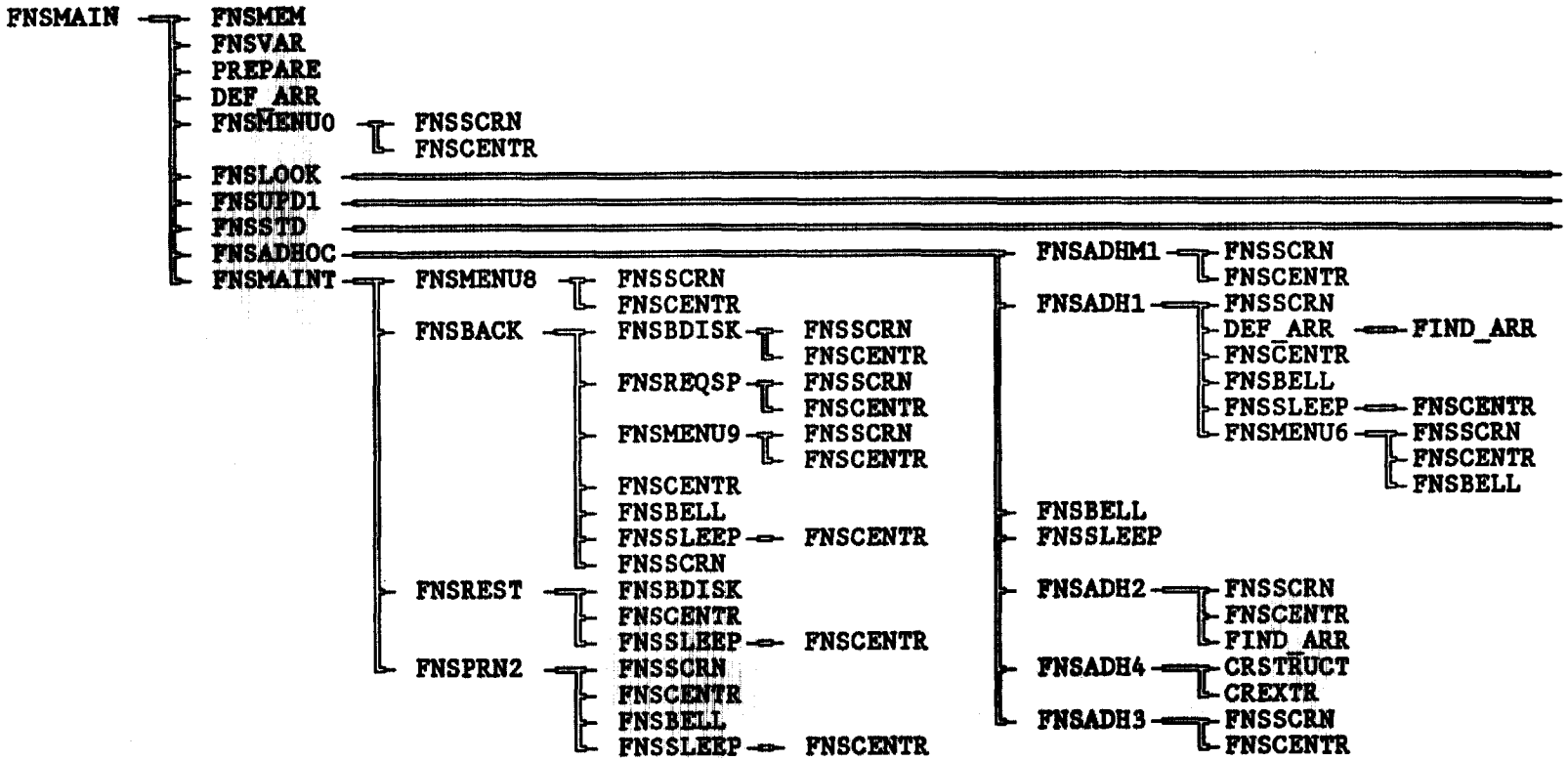

CHART II-1

RELATIONSHIP BETWEEN DATA ACCESS SYSTEM PROGRAMS
(continued)

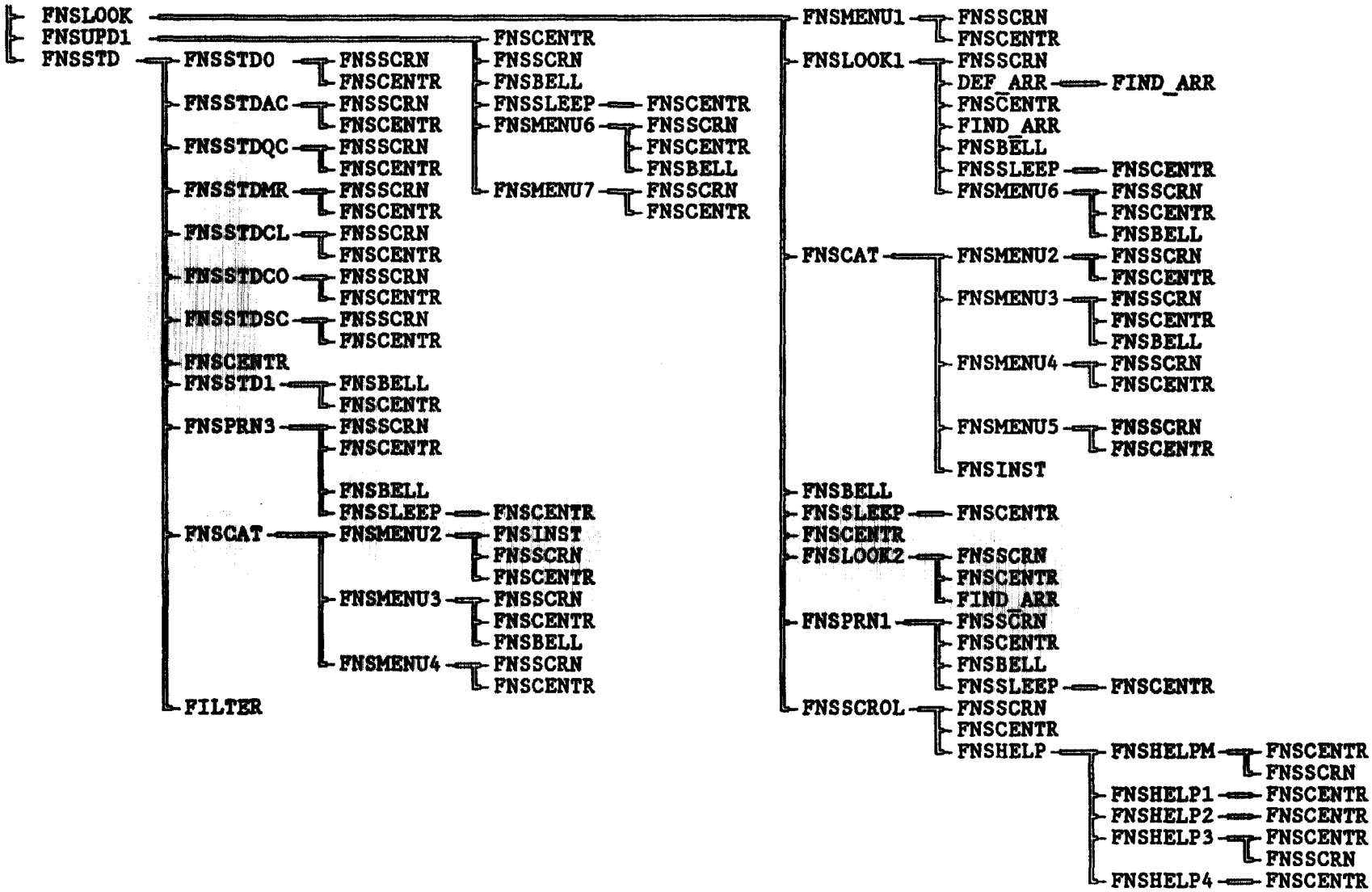

TABLE II-2

Program Descriptions

Program	Description
CREXTR	Creates Ad Hoc extract files
CRSTRUCT	Creates "Structure Extended Files" prior to creating ad hoc extracts (see Note at end of table)
FILTER	Restricts Standard Report processing to selected jurisdiction
FNSADH1	Flags variables to be included in Ad Hoc Reports
FNSADH2	Creates variable information file for Ad Hoc Reports
FNSADH3	RR/Ad Hoc message screen
FNSADH4	Creates Ad Hoc report extract files
FNSADHM1	Main menu for Ad Hoc report module
FNSADHOC	Control program for Ad Hoc reporting module
FNSBACK	Control program for data backup
FNSBDISK	Displays menu to select backup drive
FNSCAT	Program to process display category selection
FNSINST	Initializes variables containing state selection data
FNSLOOK	Control program for Display Module
FNSLOOK1	Program which flags variables for display
FNSLOOK2	Creates display file for Data Display module
FNSMAIN	Main Control Program
FNSMAINT	Control program for Maintenance Module
FNSMEM	Creates "public memory" variables

TABLE II-2
(continued)

Program Descriptions

Program	Description
FNSMENU0	Displays Main System Menu
FNSMENU1	Display Main Menu for Data Display Module
FNSMENU2	Display menu for Display Category Selection
FNSMENU3	State selection menu
FNSMENU4	FNS Region selection menu
FNSMENU5	Census Region selection menu (NOTE: This program is included with the system but is not used in Version 2.0)
FNSMENU6	Displays menu for Subject Area selection
FNSMENU7	Menu to select desired system in states with multiple systems
FNSMENU8	Main menu for Maintenance Module
FNSMENU9	Menu for type of backup device
FNSPRN1	Prints Data Display file
FNSPRN2	Prints Data Update History Log
FNSPRN3	Prints Standard Report display file
FNSREQSP	Computes space required for storing backup files
FNSREST	Control program for restoring data from backup files
FNSSCROL	Displays file containing results of Display Module
FNSSTD	Control program for Standard Report Module
FNSSTD0	Menu for selecting Standard Report subject area

TABLE II-2
(continued)

Program Descriptions

Program	Description
FNSSTD1	Creates Standard Report file
FNSSTDAC	Menu to select Automated Certification module for standard report
FNSSTDCL	Menu to select Claims (Census) module for standard report
FNSSTDCO	Menu to select Computer Matching module for standard report
FNSSTDMR	Menu to select Monthly Reporting module for standard report
FNSSTDQC	Menu to select Quality Control module for standard report
FNSSTDSC	Menu to select Claims (Survey) module for standard report
FNSUPD1	Program for modifying data values
FNSUTIL	Procedure file of commonly used utilities, including: FNSSCRN - Generates screens FNSBELL - Rings bell FNSSLEEP - Halts program execution until user responds FNSCENTR - Centers text within box FNSDEBUG - Activates debug code PREPARE - Prepares for array definition DEF_ARR - Defines arrays TO_ARRAY - Stores data to arrays FIND_ARR - Finds absolute address of array STO_SCRN - Stores screen to array RST_SCRN - Restores screen from array
FNSVAR	Program to initialize public variables
HELP1	Program to create help screen
TRLIB	Tom Rettig Library Load/Control Program

NOTE: All files have .prg extension. Refer to the dBase III Plus documentation for a discussion of Structure Extended files and memory variables.

TABLE II-3

Description of Files Containing FSPOS Data

Description	dBase data file
Automated Certification Systems data	
	ACS1
	ACS2
	ACS3
	ACS4
	ACS5
Claims (Census) data	
	CLAIMS1
	CLAIMS2
	CLAIMS3
	CLAIMS4
	CLAIMS5
	CLAIMS6
	CLAIMS7
	CLAIMS8
	CLAIMS9
Computer Matching data	
	COM
Monthly Reporting data	
	MR1
	MR2
	MR3
	MR4
	MR5
Quality Control data	
	QC1
	QC2
	QC3
Claims (Survey) data	
	SURVEY1
	SURVEY2
	SURVEY3
	SURVEY4
	SURVEY5
	SURVEY6
	SURVEY7

TABLE II-4

Description of Other System Data Files

Data File	Description
Data Files Used by Ad Hoc Reporting Module	
ADHOC	Main data file for use with Relational Report Writer
DATACS	File containing Automated Certification data from most recent Ad Hoc request
DATCLA	File containing Claims (Census) data from most recent Ad Hoc request
DATCOM	File containing Computer Matching data from most recent Ad Hoc request
DATMR	File containing Monthly Reporting data from most recent Ad Hoc request
DATQC	File containing Quality Control data from most recent Ad Hoc request
DATSUR	File containing Claims (Survey) data from most recent Ad Hoc request
EXTACS	Extended structure file used to create DATACS.dbf
EXTCLA	Extended structure file used to create DATCLA.dbf
EXTCOM	Extended structure file used to create DATCOM.dbf
EXTMR	Extended structure file used to create DATMR.dbf
EXTQC	Extended structure file used to create DATQC.dbf
EXTSUR	Extended structure file used to create DATSUR.dbf
STRUCT	File which acts as a "shell" for extended structure files
VARINFO	Data file containing information on variables to be included in Ad Hoc extract files
Data Files Used by Maintenance Module	
BACKLIST	Data file containing list of .dbf files to be processed during backup and restore
FNSLOG	Data file containing history of data updates

TABLE II-4
(continued)

Description of Other System Data Files

Data File	Description
Data Files used by Data Display Module	
DISPLAY	Intermediate data file used to store results of data display requests
DISPLAY2	Data file used to store results of data display requests
Data Files used by Standard Report Module	
RESULT2	Data file containing results of Standard Report request
RPTDEF	Intermediate file used to create standard report
SHELLRE2	Intermediate file used to create standard report
STDSTRUC	Intermediate file used to create standard report
STRUCT2	Intermediate file used to create standard report
Other Data Files	
DATADICT	Data Dictionary file
FNSSTATE	Data file containing descriptive information on states

NOTE: All files have .dbf extension

E. Description of Index Files

Table II-5 describes the index files used by the system. (Index files are special files used by dBase III Plus to speed retrieval of specific data from a dBase data file. Refer to the dBase III Plus documentation for further information on index files.) VARINFO is re-indexed each time an Ad Hoc request is processed. The Ad Hoc module also erases data and index files created from earlier sessions and creates new files as hoc subject area index files if required by the Ad Hoc request. Therefore, some of the index files may not be present at a given moment.

TABLE II-5

Index Files and Index Keys

Index File	Associated Data File	Index Key
DATADICT	DATADICT	SUBJECT-FIELD_NAME
IDNUM	DATADICT	IDNUM
NAME	DATADICT	FIELD_NAME
ADHOC	ADHOC	STATE
VARINFO	VARINFO	SUBJECT-FIELD_NAME
DATACS	DATACS	STATE
DATCLA	DATCLA	STATE
DATCOM	DATCOM	STATE
DATMR	DATMR	STATE
DATQC	DATQC	STATE
DATSUR	DATSUR	STATE

NOTE: All index files have a .ndx extension.

III. SYSTEM ADMINISTRATOR FUNCTIONS

The use of a Data Dictionary and the modular design of the Access System makes most system maintenance functions relatively simple. More substantial modifications require changes to the program source code and thus require more programming expertise than the common functions described below.

Adding data for subject areas currently included in the Access System should be performed by modifying existing .dbf files. Data for new subject areas should be added to the Access System by adding new .dbf files.

A. Adding a Variable to an Existing Data File

Adding variables to existing data files will require the System Administrator to perform the following:

1. Determine the data file to be modified

   The standard report Module requires that all variables obtained from an instrument module are contained within the same data file. Therefore, data for the new variable must be added to the same data file which contains data on other variable from the same module.

   a. Refer to the census/survey instrument to determine the module number for the new variable.
   b. Enter dBase and go to the dot prompt.
   c. Enter the following commands:

```
USE DATADICT
DISPLAY ALL FIELDS SUBJECT,MODULE,FILENAME FOR
      SUBJECT = 'sub' .AND. MODULE = 'mod'
```

   where

   sub = Three character abbreviation for subject area
   - ACS - Automated Certification Systems
   - CLA - Claims (Census)
   - QC - Quality Control
   - MR - Monthly Reporting
   - COM - Computer Matching
   - SCL - Claims (Survey)

   mod = module number.

   If the variable is a constructed variable, the module number is the value of the instrument's last module plus one.

2. Modify the Cross-Reference, Data Dictionary, and Report Definition Files

At the dBase dot prompt, append a record to XREF.DBF. The following information should be entered into the fields for the new record:

FIELD_NAME:	Enter the name of the new variable (max. 10 characters). This name should be unique (not in use by another variable in the Access System)
FIELD_TYPE:	Enter C for character, N for numeric. The Access System will not accept logical variables
FIELD_LEN:	Enter the maximum width, or number of characters, required by the new variable. The maximum width excepted by the Access System is 45 characters.
FIELD_DEC:	Enter the number of implied decimals contained in the new variable.
DESCRIP:	Enter a short description of the variable (max. 40 characters). This description will be used by the Data Display Module.
KEYWORD:	Enter a longer (max. 80 characters) description of the new variable.
MODULE:	Enter the number of the census or survey instrument module which contains the new variable. If the variable is a constructed variable, the module number is the value of the instrument's last module plus one.
SUBJECT:	Enter the three character abbreviation for the subject area of the new variable. The abbreviations are as follows: ACS - Automated Certification Systems CLA - Claims (Census) QC - Quality Control MR - Monthly Reporting COM - Computer Matching SCL - Claims (Survey)
FILENAME:	Enter the name of the .dbf file which will contain the new variable.
IDNUM:	Enter a unique ID number for the new variable. Make sure that the number is not used for other variables in DATADICT.
CONSTRUCT:	Enter a logical "True" for constructed variables, "False" otherwise.
YESNO:	Enter a logical "True" for variables whose answers should be "Yes" or "No".
EXTRACT:	Enter a logical "True" for variables which will be accessed by the Data Access System.
REPORTNUM:	Enter the number of the Data Access System Standard Report which will include the new variable. This number corresponds to the report numbers contained on the Standard Report selection screen for each subject area.

3. Run MAKEDICT.prg

Recreate the Data Dictionary, Report Definition and index files by running the program MAKEDICT from the dot prompt.

4. Modify the structure of the data file

Use the desired .dbf file. Use the "MODIFY STRUCTURE" command to add the new variable to the .dbf file. Make sure the information about the new variable matches that described in the Data Dictionary.

5. Modify the data for each record

Use the "EDIT" command to add values for the new variable to each record in the .dbf file. You may also write a "DO" program to add the new data.\

B. Adding Files Containing Data for New Subject Areas

Data for subject areas not included in the current Access System should be added by inserting new data files into the system:

1. Install the New Data Files

Place the New .dbf Files in the same subdirectory as the current .dbf files. The following variables must be included in each installed .dbf file:

ROW_NUM:	A numeric variable representing the record number of the record.
STATE:	A two digit numeric variable containing the code for the record's state. Refer to the contents of FNSSTATE for the code values.
ID_NUM:	A unique system identifier number for states with multiple systems. This variable need not be present for subject areas which have only one system per state.

2. Modify the Cross-Reference, Data Dictionary and Report Definition Files

Append records to XREF.dbf and create new Data Dictionary and Report Definition files using the procedure described above. A new three character SUBJECT abbreviation should be used.

3. Modify Access System Programs

Several programs within the Access System should be modified to reflect the change in number of subject areas. The programs affected are listed in Table III-1. Comments showing the specific locations which should be modified are included in the source code.

Program Modifications Required
By the Addition of New Subject Areas

Program File	Reason for Modification
FNSLOOK1	This program uses specific file names to display information for systems in multi-system states as part of the data display process.
FNSUPD1	This program uses specific file names to display information for systems in multi-system states as part of the update process.
FNSMENU6	This program displays a menu of subject areas in the event of duplicate variable names.
FNSMENU7	This program displays the names of systems for systems in multi-system states.
FNSADH1	This program uses specific file names to display information for systems in multi-system states as part of the Ad Hoc report process.
FNSADH2	This program uses specific values passed from FNSADH1 to create VARINFO.dbf.
FNSADH4	This program requires specific subject area information in order to create the Ad Hoc data files.
FNSSTD	This is the control program for the Standard Report Module. It requires specific values for subject area.
FNSSTD0	This program displays a menu of available subject areas.

If standard reports for the new subject area are to be provided, a new program similar to FNSSTAC.prg must be created. This program will be used by FNSSTD.prg.

4. Backup/Restore

The names of new files to be processed by the Backup and Restore modules must be added to BACKLIST.dbf.

C. Modifying Keywords and Variable Descriptions

The "EDIT" command may be used to modify the keyword and variable description information in XREF.dbf and DATADICT.dbf.

D. Loading Data from Mainframe SAS into the Access System

The complete set of data collected for the Food Stamp Program Operations Study was provided to the Food and Nutrition Service in a format suitable for use with Statistical Analysis System (SAS) software in an IBM mainframe environment. SAS is a collection of software programs and programing languages suitable for the analysis and manipulation of survey data. Converting SAS data residing on the mainframe into data usable by the Access System requires transferring information about the data and the data itself from the mainframe to a microcomputer. The data must then be converted into dBase format. Familiarity with SAS, a telecommunications package and JCL is required to perform the conversion.

1. Create a Extended Structure Text File on the Mainframe

Listing III-1 is an example SAS program which creates a text file which will contain information on the content of the data file. Inputs for the SAS program may be obtained by using PROC CONTENTS and PROC MEANS, saving the output as text files, and editing the files to conform to the format requirements shown in Listing III-1.

2. Create a Comma Delimited Data File on the Mainframe

Listing III-2 shows the an example SAS program which creates a comma separated value text file containing the data to be converted.

LISTING III-1

Creating a Comma Separated Extended Structure File

```
*************************************************;
** READS CONTENTS AND MEANS FILES TO CREATE   **;
** DATA DICTIONARY (INPUT FILES SHOULD BE IN  **;
** CARD FORMAT)                               **;
*************************************************;
DATA CONTENTS;
 INFILE CONTIN;
 INPUT
      VARPOS  1-8
      @10 VARNAME $CHAR8.
      @19 VARTYPE $CHAR1.
      @23 VARLABEL $CHAR40.;
 DECIMAL = 0;
PROC SORT DATA=CONTENTS;
   BY VARNAME;
DATA MEANS;
  INFILE MEANIN;
  INPUT
        @1 VARNAME $CHAR8.
           MINVALUE  $
           MAXVALUE  $;
  VARWIDTH = MAX(LENGTH(MINVALUE),LENGTH(MAXVALUE));

PROC SORT DATA=MEANS;
   BY VARNAME;

DATA MERGED;
  MERGE
       CONTENTS
        MEANS;
  BY VARNAME;

PROC SORT DATA=MERGED;
  BY VARPOS;

DATA _NULL_;
 SET MERGED;
  FILE PRINT;
  IF VARWIDTH EQ . THEN VARWIDTH = 2;
 PUT
    @2 VARNAME ',' VARTYPE ',' VARWIDTH ',' DECIMAL ',' VARLABEL;
```

LISTING III-2

Creating a Comma Separated Data File

```
DATA _NULL_;
 SET DATAIN.MEMBER;
 M = -1;
 RETAIN ROW_NUM 0;
 ROW_NUM + 1;
 FILE DATAOUT;
 PUT
  STATE      +M ','
  ID_NUM     +M ','
  ROW_NUM    +M ','
  VAR1       +M ','
  VAR2       +M ','
  VAR3       +M ','
  .
  .
  .
  VARN   '*';
```

NOTES:

DATAIN is the DDNAME for the input SAS library
MEMBER is the member name
DATAOUT is the DDNAME for the output text file
STATE, ID_NUM, ROW_NUM, VAR1...VARN are variable names

3. Download the Two Files to the Microcomputer

The text files created in Steps 1 and 2 must be transferred to the microcomputer. Communications programs providing for error correction are recommended for transferring large files. KERMIT is one such communications program. The files should be reviewed for spurious characters, blank lines, or other errors before proceeding.

4. Create an Extended Structure dBase File

The .dbf file STRUCT, supplied with the Access System, may be used to create an extended structure dBase file. STRUCT should not contain any records. Existing records may be removed using the following commands:

```
USE STRUCT
DELETE ALL FOR FIELD_NAME <> 'Z'
PACK
```

With STRUCT still in the active area, enter the following commands:

```
APPEND FROM extfn TYPE DELIMITED WITH ,
COPY TO fnnew
```

where:

extfn is the extended structure text file which was downloaded from the mainframe
fnnew is the name of the new extended structure dBase file

5. Create a dBase Data File

Create a new data file in dBase format using the following commands:

```
CREATE newdata FROM fnnew
APPEND FROM textfn TYPE DELIMITED WITH ,
```

where:

newdata is the name of the new dBase data file,
fnnew in the extended structure dBase file created above, textfn is the text data file downloaded from the mainframe

APPENDIX A:

SOURCE LISTINGS AND CROSS-REFERENCE TABLES

```
 1 **********************************
 2 *  Program: CREXTR.PRG           *
 3 *    Edson 10/88                 *
 4 *&Creates Ad Hoc Extract Files
 5 **********************************
 6
 7 ******** DEFINE PARAMETER
 8 PARAMETERS FILEAREA
 9 COUNTER = 0
10 ******** CREATE EXTRACT FILE
11 CREATE DAT&FILEAREA FROM EXT&FILEAREA
12 CLOSE DATABASES
13 SELECT A
14 USE DAT&FILEAREA
15 SELECT C
16 USE VARINFO
17 DO WHILE .NOT. EOF()
18    IF SUBJECT = '&FILEAREA'
19       COUNTER = COUNTER + 1
20       IF COUNTER = 1
21          TEMPFILE = C->FILENAME
22          SELECT A
23          APPEND FROM &TEMPFILE
24          INDEX ON STATE  TO  DAT&FILEAREA
25       ELSE
26          SELECT B
27          TEMPFILE = C->FILENAME
28          USE &TEMPFILE
29          SELECT A
30          TEMPNAME = C->FIELD_NAME
31          UPDATE ON ROWNUM FROM B REPLACE &TEMPNAME WITH B->&TEMPNAME;
31            RANDOM
32       ENDIF
33    ENDIF
34    SELECT C
35    SKIP
36 ENDDO
37 CLOSE DATABASES
```

VARIABLES CROSS-REF FOR CREXTR.PRG

Variable	Line number in file
A:	13 22 29
B:	26 31
C:	15 21 27 30 34
COUNTER:	9 19 20
DAT:	11 14 24
EXT:	11
FIELD_NAME:	30
FILEAREA:	8 11 14 24
FILENAME:	21 27
REPLACE:	31
ROWNUM:	31
STATE:	24
SUBJECT:	18
TEMPFILE:	21 23 27 28
TEMPNAME:	30 31
VARINFO:	16

```
 1 ******************************
 2 * CRSTRUCT.PRG - Edson 9/27/88
 3 *&Creates Structure Extended Files Prior to Creating Ad Hoc Extracts
 4 ******************************
 5 SET PROCEDURE TO FNSUTIL
 6
 7 ********************* DEFINE PARAMETERS
 8 PARAMETERS FILEAREA
 9 ********************* SET UP EXTRACT STRUCTURE FILE
10 USE STRUCT
11 ERASE STR&FILEAREA.DBF
12 COPY TO EXT&FILEAREA
13 ********************* SELECT EXTRACT STRUCTURE FILE
14 SELECT A
15 USE EXT&FILEAREA
16 APPEND BLANK
17 ********************* SELECT VARIABLE INFORMATION FILE
18 SELECT B
19 USE VARINFO
20 GO TOP
21 ********************* PLACE INFORMATION INTO STRUCTURE FILE
22 DO WHILE .NOT. EOF()
23    IF B->SUBJECT = "&FILEAREA"
24       REPLACE A->FIELD_NAME WITH B->FIELD_NAME, A->FIELD_TYPE WITH B;
24        ->FIELD_TYPE, A->FIELD_LEN WITH B->FIELD_LEN,   A->FIELD_DEC;
24         WITH B->FIELD_DEC
25       SELECT A
26       APPEND BLANK
27       SELECT B
28    ENDIF
29    SKIP
30 ENDDO
31 ********************* DELETE EMPTY RECORDS
32 SELECT A
33 DELETE ALL FOR FIELD_NAME = ""
34 PACK
35 CLOSE DATABASES
```

VARIABLES CROSS-REF FOR CRSTRUCT.PRG

Variable	Line number in file			
A:	14	24	25	32
B:	18	23	24	27
EXT:	12	15		
FIELD_DEC:	24			
FIELD_LEN:	24			
FIELD_NAME:	24	33		
FIELD_TYPE:	24			
FILEAREA:	8	12	15	
FILEAREA.D:	11			
FNSUTIL:	5			
STR:	11			
SUBJECT:	23			
VARINFO:	19			

Table of Contents

```
 1 * FILTER.PRG                *
 2 * written on 12 Feb 1989    *
 3 * by David Edson            *
 4 *  Adds jurisdiction filter *
 5 *  to Std. Report Module    *
 6 *****************************
 7 *&Determines state abbreviations for display states
 8 SET ESCAPE OFF
 9 SELECT 1
10 USE FNSSTATE
11 DO CASE
12 CASE disp_cat = 1 && DISPLAY SPECIFIC STATES
13    numstate = numtog
14    num = 1
15    DO WHILE num < numstate + 1
16       y = LTRIM(STR(num,2))
17       LOCATE FOR ABBREV = state_&y
18       code = STATECODE
19       * SELECT 2
20       * COUNT FOR STATE = code to numobs&y
21       * IF numobs&y > maxobs
22       *   maxobs = numobs&y
23       * ENDIF
24       * SELECT 1
25       num = num + 1
26    ENDDO num
27 CASE disp_cat = 2 && DISPLAY FNS REGIONS
28    COUNT FOR FNSREGION = disp_choice to numstate
29    LOCATE FOR FNSREGION = disp_choice
30    num = 1
31    DO WHILE num < numstate + 1
32       y = LTRIM(STR(num,2))
33       state_&y = ABBREV
34       code = STATECODE
35       * SELECT 2
36       * COUNT FOR STATE = code to numobs&y
37       * IF numobs&y > maxobs
38       *   maxobs = numobs&y
39       * ENDIF
40       * SELECT 1
41       CONTINUE
42       num = num + 1
43    ENDDO num
44 CASE disp_cat = 3 && DISPLAY ALL STATES (using FNS REGION)
45    COUNT FOR FNSREGION = disp_choice to numstate
46    LOCATE FOR FNSREGION = disp_choice
47    num = 1
48    DO WHILE num < numstate + 1
49       y = LTRIM(STR(num,2))
50       state_&y = ABBREV
51       code = STATECODE
52       * SELECT 2
53       * COUNT FOR STATE = code to numobs&y
54       * IF numobs&y > maxobs
55       *   maxobs = numobs&y
56       * ENDIF
57       * SELECT 1
58       CONTINUE
59       num = num + 1
60    ENDDO num
61 ENDCASE
```

VARIABLES CROSS-REF FOR FILTER.PRG

Variable	Line number in file											
ABBREV:	17	33	50									
CODE:	18	34	51									
DISP_CAT:	12	27	44									
DISP_CHOIC:	28	29	45	46								
FNSREGION:	28	29	45	46								
FNSSTATE:	10											
NUM:	14	15	16	25	30	31	32	42	47	48	49	59
NUMSTATE:	13	15	28	31	45	48						
NUMTOG:	13											
STATECODE:	18	34	51									
STATE_:	17	33	50									
Y:	16	17	32	33	49	50						

```
 1 * FNSADH1.PRG *
 2 * written on 25 April 1988
 3 * modified: Edson 14 June 1988
 4 * by Richard Lieberman
 5 *&Flags Variables to be Included in Ad Hoc Reports
 6 SET PROCEDURE TO FNSUTIL
 7 DO FNSSCRN WITH 'FNSLOOK1'
 8 arr_len = 81
 9 numelt = 100
10 DO DEF_ARR WITH "VAR_LINE","STR",100,arr_len
11
12 STORE SUBSTR(array_id,1,8) TO argument
13 CALL trpass WITH argument
14 CALL trexe  WITH "SEPARATE"
15 segment = tr_retc
16 offset = tr_retn
17
18 USE DATADICT INDEX IDNUM,DATADICT,NAME
19
20 SET COLOR TO &highlight
21 @5,25 SAY "AD HOC REPORTING MODULE"
22 @7,6 SAY "ID NUM"
23 @7,15 SAY "FIELD NAME"
24 @7,37 SAY "FIELD DESCRIPTION"
25 @7,70 SAY "SUBJECT"
26 SET COLOR TO &color
27
28 STORE 1 TO elements
29
30 prow = 9
31 @8,1 CLEAR TO 20,78
32
33 DO WHILE elements < 101
34
35
36    STORE SPACE(10) TO lookvar
37    @23,18 SAY "Enter Variable Name or Variable Number Above"
38    SET COLOR TO &highlight
39    @24,3 SAY "[Ctrl][W]=Exit"
40    SET COLOR TO &color
41
42    @22,2 SAY SPACE(77)
43    @22,35 GET lookvar PICTURE REPLICATE("!",8)
44    READ
45
46    IF READKEY() = 14 .OR. READKEY() = 270
47       STORE SPACE(1) TO answer
48       @22,1 CLEAR TO 23,78
49       DO FNSCENTR WITH "Do you wish to exit and return to the";
49        + " previous menu (Y/N)?",22
50       DO FNSCENTR WITH "Type [Y] to exit; [N] to continue selecting";
50        + " variables",23
51       @24,3 SAY REPLICATE(CHR(205),15)
52       @23,68 GET answer PICTURE "Y"
53       READ
54       IF answer = "Y"
55          RELEASE ALL
56          IF elements = 1     && reset elements in cases where
57             elements = 0     && user did not select any variables
58          ENDIF
59          RETURN
60       ELSE
61          @23,1 SAY SPACE(77)
62          LOOP
```

```
63          ENDIF
64       ENDIF
65
66       IF .NOT. ISALPHA("&lookvar")
67          SET ORDER TO 1
68          SEEK VAL(SUBSTR(lookvar,1,8))
69          IF FOUND()
70             IF elements > 12
71                y = 19
72                x = elements - 2
73                @20,1 SAY SPACE(77)
74                prow = 20
75                DO WHILE y > 8
76                   IF TYPE("offset") = "C"
77                      CALL trpass with offset
78                      CALL trexe with "DEC"
79                      offset = tr_retn
80                   ENDIF
81                   new_offset = offset + (x * arr_len)
82                   CALL trpass WITH segment
83                   CALL trpass WITH new_offset
84                   CALL trexe WITH "PEEKSTR"
85                   @y,5 SAY SUBSTR(tr_retc,1,4)
86                   @y,15 SAY SUBSTR(tr_retc,5,10)
87                   @y,30 SAY SUBSTR(tr_retc,15,40)
88                   subval = '   '
89                   tempsub = SUBSTR(tr_retc,55,2)
90                   DO CASE
91                   CASE tempsub = 'MR'
92                      subval = 'MR '
93                   CASE tempsub = 'CL'
94                      subval = 'CLA'
95                   CASE tempsub = 'CO'
96                      subval = 'COM'
97                   CASE tempsub = 'AC'
98                      subval = 'ACS'
99                   CASE tempsub = 'QC'
100                     subval = 'QC '
101                  CASE tempsub = 'SU'
102                     subval = 'SUR'
103                  ENDCASE
104                  @y,72 SAY subval
105                  x = x - 1
106                  y = y - 1
107            ENDDO
108         ENDIF
109         @prow,5 SAY STR(IDNUM,4)
110         @prow,15 SAY FIELD_NAME
111         @prow,30 SAY DESCRIP
112         @prow,72 SAY TRIM(SUBJECT)
113         line = STR(IDNUM,4)+FIELD_NAME+DESCRIP+FILENAME
114         DO FIND_ARR WITH "VAR_LINE"
115         IF TYPE("offset") = "C"
116            CALL trpass with offset
117            CALL trexe with "DEC"
118            offset = tr_retn
119         ENDIF
120         new_offset = offset + ((elements-1) * arr_len)
121         CALL trpass WITH line
122         CALL trexe WITH "retc"
123         CALL trpass WITH segment
124         CALL trpass WITH new_offset
125         CALL trexe WITH TYPE("new_offset")
126         CALL trpass WITH tr_retc
127         CALL trexe WITH "POKESTR"
128         elements = elements + 1
129         prow = prow + 1
```

```
130       ELSE
131          DO FNSBELL
132          DO FNSCENTR WITH "*** ID NUMBER NOT FOUND ***",22
133          DO FNSSLEEP
134       ENDIF
135    ELSE
136       SET EXACT ON
137       COUNT FOR FIELD_NAME = "&lookvar" TO samevar
138       IF samevar > 1
139          cl = 'TEMP'
140          CALL trpass WITH cl
141          CALL trexe WITH "SAVESCR"
142          DO FNSMENU6
143          DO CASE
144          CASE fns_retn = 1
145             subject = 'ACS'
146          CASE fns_retn = 2
147             subject = 'QC'
148          CASE fns_retn = 3
149             subject = 'MR'
150          CASE fns_retn = 4
151             subject = 'CLA'
152          CASE fns_retn = 5
153             subject = 'COM'
154          CASE fns_retn = 6
155             subject = 'SUR'
156          ENDCASE
157          cl = 'TEMP'
158          CALL trpass WITH cl
159          CALL trexe WITH "RESTSCR"
160          SET ORDER TO 2
161          SEEK "&subject&lookvar"
162       ELSE
163          SET ORDER TO 3
164          SEEK "&lookvar"
165       ENDIF
166       IF FOUND()
167          IF elements > 12
168             y = 19
169             x = elements - 2
170             @20,1 SAY SPACE(77)
171             prow = 20
172             DO WHILE y > 8
173                IF TYPE("offset") = "C"
174                   CALL trpass with offset
175                   CALL trexe with "DEC"
176                   offset = tr_retn
177                ENDIF
178                new_offset = offset + (x * arr_len)
179                CALL trpass WITH segment
180                CALL trpass WITH new_offset
181                CALL trexe WITH "PEEKSTR"
182                @y,5 SAY SUBSTR(tr_retc,1,4)
183                @y,15 SAY SUBSTR(tr_retc,5,10)
184                @y,30 SAY SUBSTR(tr_retc,15,40)
185                subval = '    '
186                tempsub = SUBSTR(tr_retc,55,2)
187                DO CASE
188                CASE tempsub = 'MR'
189                   subval = 'MR  '
190                CASE tempsub = 'CL'
191                   subval = 'CLA'
192                CASE tempsub = 'CO'
193                   subval = 'COM'
194                CASE tempsub = 'AC'
195                   subval = 'ACS'
196                CASE tempsub = 'QC'
```

```
197                   subval = 'QC '
198                CASE tempsub = 'SU'
199                   subval = 'SUR'
200                ENDCASE
201                @y,72 SAY subval
202                x = x - 1
203                y = y - 1
204             ENDDO
205          ENDIF
206          @prow,2 SAY SPACE(25)
207          @prow,5 SAY STR(IDNUM,4)
208          @prow,15 SAY FIELD_NAME
209          @prow,30 SAY DESCRIP
210          @prow,72 SAY TRIM(SUBJECT)
211          line = STR(IDNUM,4)+FIELD_NAME+DESCRIP+FILENAME
212          DO FIND_ARR WITH "VAR_LINE"
213          IF TYPE("offset") = "C"
214             CALL trpass with offset
215             CALL trexe with "DEC"
216             offset = tr_retn
217          ENDIF
218          new_offset = offset + ((elements-1) * arr_len)
219          CALL trpass WITH line
220          CALL trexe WITH "retc"
221          CALL trpass WITH segment
222          CALL trpass WITH new_offset
223          CALL trexe WITH TYPE("new_offset")
224          CALL trpass WITH tr_retc
225          CALL trexe WITH "POKESTR"
226          elements = elements + 1
227          prow = prow + 1
228       ELSE
229          DO FNSBELL
230          DO FNSCENTR WITH "*** VARIABLE NAME NOT FOUND ***",22
231          DO FNSSLEEP
232       ENDIF
233    ENDIF
234
235 ENDDO elements
236 RELEASE ALL
237 RETURN
```

VARIABLES CROSS-REF FOR FNSADH1.PRG

Variable	Line number in file
ANSWER:	47 52 54
ARGUMENT:	12 13
ARR_LEN:	8 10 81 120 178 218
C1:	139 140 157 158
DATADICT:	18
DEF_ARR:	10
DESCRIP:	111 113 209 211
ELEMENTS:	28 33 56 57 70 72 120 128 167 169 218 226
FIELD_NAME:	110 113 137 208 211
FILENAME:	113 211
FIND_ARR:	114 212
FNSBELL:	131 229
FNSCENTR:	49 50 132 230
FNSMENU6:	142
FNSSCRN:	7
FNSSLEEP:	133 231
FNSUTIL:	6
FNS_RETN:	144 146 148 150 152 154
IDNUM:	18
LINE:	113 121 211 219
LOOKVAR:	36 43 66 68
NAME:	18
NEW_OFFSET:	81 83 120 124 125 178 180 218 222 223
NUMELT:	9
OFFSET:	16 76 77 79 81 115 116 118 120 173 174 176 178 213 214 216 218
PROW:	30 74 109 110 111 112 129 171 206 207 208 209 210 227
SAMEVAR:	137 138
SEGMENT:	15 82 123 179 221
SUBJECT:	145 147 149 151 153 155
SUBVAL:	88 92 94 96 98 100 102 104 185 189 191 193 195 197 199 201
TEMPSUB:	89 91 93 95 97 99 101 186 188 190 192 194 196 198
TREXE:	14 78 84 117 122 125 127 141 159 175 181 215 220 223 225
TRPASS:	13 77 82 83 116 121 123 124 126 140 158 174 179 180 214 219 221 222 224
TR_RETC:	15 126 224
TR_RETN:	16 79 118 176 216
X:	72 105 169 202
Y:	71 75 85 86 87 104 106 168 172 182 183 184 201 203

```
 1 * FNSADH2.PRG           *
 2 * written on 18 AUG 1988 *
 3 * by David Edson         *
 4 *&Creates Variable Information File for Ad Hoc Reports
 5 CLOSE DATABASES
 6 SELECT A
 7 USE VARINFO
 8 DELETE ALL
 9 PACK
10
11 DO FNSSCRN WITH 'FNSADH2'
12 DO FNSCENTR WITH 'Working...',22
13 numelt = 1
14 DO WHILE numelt < ELEMENTS
15    DO FIND_ARR WITH "VAR_LINE"
16    new_offset = offset + ((numelt-1) * arr_len)
17    CALL trpass WITH segment
18    CALL trpass WITH new_offset
19    CALL trexe WITH "PEEKSTR"
20    varnum = SUBSTR(tr_retc,1,4)
21    varname = SUBSTR(tr_retc,5,8)
22    vardes = SUBSTR(tr_retc,13,40)
23    varfile = SUBSTR(tr_retc,55,8)
24    tempsub = SUBSTR(tr_retc,55,2)
25    DO CASE
26    CASE tempsub = 'MR'
27       subval = 'MR '
28    CASE tempsub = 'CL'
29       subval = 'CLA'
30    CASE tempsub = 'CO'
31       subval = 'COM'
32    CASE tempsub = 'AC'
33       subval = 'ACS'
34    CASE tempsub = 'QC'
35       subval = 'QC '
36    CASE tempsub = 'SU'
37       subval = 'SUR'
38    ENDCASE
39    APPEND BLANK
40    REPLACE FIELD_NAME WITH varname,SUBJECT WITH subval,FILENAME WITH;
40      varfile
41    numelt = numelt + 1
42    IF numelt > ELEMENTS - 1
43       EXIT
44    ENDIF
45 ENDDO
46 INDEX ON SUBJECT-FIELD_NAME TO VARINFO
47 SELECT B
48 USE DATADICT INDEX DATADICT
49 SELECT A
50 USE VARINFO INDEX VARINFO
51 DO WHILE .NOT. EOF()
52    subval = A->SUBJECT
53    varname = A->FIELD_NAME
54    SELECT B
55    GO TOP
56    SEEK subval-varname
57    SELECT A
58    REPLACE FIELD_TYPE WITH B->FIELD_TYPE,FIELD_LEN WITH B->FIELD_LEN,;
58      FIELD_DEC WITH B->FIELD_DEC
59    SKIP
60 ENDDO
61
62 CLOSE DATABASES
```

63 RETURN

VARIABLES CROSS-REF FOR FNSADH2.PRG

Variable	Line number in file
A:	6 49 52 53 57
ARR_LEN:	16
B:	47 54 58
DATADICT:	48
ELEMENTS:	14 42
FIELD_DEC:	58
FIELD_LEN:	58
FIELD_NAME:	40 46 53
FIELD_TYPE:	58
FILENAME:	40
FIND_ARR:	15
FNSCENTR:	12
FNSSCRN:	11
NEW_OFFSET:	16 18
NUMELT:	13 14 16 41 42
OFFSET:	16
SEGMENT:	17
SUBJECT:	40 46 52
SUBVAL:	27 29 31 33 35 37 40 52 56
TEMPSUB:	24 26 28 30 32 34 36
TREXE:	19
TRPASS:	17 18
VARDES:	22
VARFILE:	23 40
VARINFO:	7 46 50
VARNAME:	21 40 53 56
VARNUM:	20

```
 1 ******************************************
 2 * FNSADH3                                *
 3 * written by Edson: 11 Sept 1988         *
 4 *&RR/Ad Hoc Message Screen
 5 * Public Variables Required:             *
 6 *    highlight                           *
 7 *    color                               *
 8 *    fns_retn                            *
 9 * Procedures Required:                   *
10 *    FNSSCRN, FNSCENTER in FNSUTIL       *
11 ******************************************
12 SET PROCEDURE TO FNSUTIL
13
14 DO FNSSCRN WITH 'FNSADH3'                                          &&;
14  MODIFY
15 DO FNSCENTR WITH 'Press any key to continue',22                    &&;
15  MODIFY
16 scrtitle = 'USING RELATIONAL REPORT WRITER'                        &&;
16  MODIFY
17
18 ******************************
19 DO FNSCENTR WITH 'USING RELATIONAL REPORT WRITER FOR AD HOC REPORTS',8
20 DO FNSCENTR WITH '  Memory limitations on your computer prevent you',10
21 DO FNSCENTR WITH 'from using both Relational Report Writer and the ',11
22 DO FNSCENTR WITH 'FSPOS Data Access System at the same time.       ',12
23 DO FNSCENTR WITH '  Press any key to return to DOS, then type:    ',13
24 DO FNSCENTR WITH '                                                ',14
25 DO FNSCENTR WITH '               RR <CR>                          ',15
26 DO FNSCENTR WITH '                                                ',16
27 DO FNSCENTR WITH 'to create an Ad Hoc report.                     ',17
28 DO FNSCENTR WITH '  At the DOS prompt, type "FNS" to return to the ',18
29 DO FNSCENTR WITH 'FSPOS Data Access System.                       ',19
30
31 key = 0                                 && Wait for user key
32 DO WHILE key = 0                        && Wait for user key
33    key = INKEY()                         && Wait for user key
34 ENDDO                                   && Wait for user key
35
36 RETURN
```

VARIABLES CROSS-REF FOR FNSADH3.PRG

Variable	Line number in file
FNSCENTR:	15 19 20 21 22 23 24 25 26 27 28 29
FNSSCRN:	14
FNSUTIL:	12
KEY:	31 33
SCRTITLE:	16

```
 1 **********************************
 2 * Program: FNSADH4.PRG           *
 3 *  Edson 10/88                   *
 4 *&Creates Ad Hoc Report Extract Files
 5 **************************************
 6 @8,1 CLEAR TO 20,78
 7 DO FNSCENTR WITH "System is gathering information about the items",10
 8 DO FNSCENTR WITH "which are to be included in the Ad Hoc extracts.",11
 9 PUBLIC TEMPFILE, TEMPNAME
10 DO CRSTRUCT WITH 'ACS' && CREATE STRUCTURE FOR EXTRACT DATABASE
11 DO CRSTRUCT WITH 'CLA' && CREATE STRUCTURE FOR EXTRACT DATABASE
12 DO CRSTRUCT WITH 'MR'  && CREATE STRUCTURE FOR EXTRACT DATABASE
13 DO CRSTRUCT WITH 'QC'  && CREATE STRUCTURE FOR EXTRACT DATABASE
14 DO CRSTRUCT WITH 'COM' && CREATE STRUCTURE FOR EXTRACT DATABASE
15 DO CRSTRUCT WITH 'SUR' && CREATE STRUCTURE FOR EXTRACT DATABASE
16 @8,1 CLEAR TO 20,78
17 DO FNSCENTR WITH "System is gathering data for the Ad Hoc extracts.",10
18 DO CREXTR WITH 'ACS'   && CREATE AND PLACE DATA IN EXTRACT DATABASE
19 DO CREXTR WITH 'CLA'   && CREATE AND PLACE DATA IN EXTRACT DATABASE
20 DO CREXTR WITH 'MR'    && CREATE AND PLACE DATA IN EXTRACT DATABASE
21 DO CREXTR WITH 'QC'    && CREATE AND PLACE DATA IN EXTRACT DATABASE
22 DO CREXTR WITH 'COM'   && CREATE AND PLACE DATA IN EXTRACT DATABASE
23 DO CREXTR WITH 'SUR'   && CREATE AND PLACE DATA IN EXTRACT DATABASE
24 RETURN
```

VARIABLES CROSS-REF FOR FNSADH4.PRG

Variable	Line number in file					
CREXTR:	18	19	20	21	22	23
CRSTRUCT:	10	11	12	13	14	15
FNSCENTR:	7	8	17			
TEMPFILE:	9					
TEMPNAME:	9					

```
 1 ********************************************
 2 * FNSADHM1.PRG                             *
 3 * written by Edson: 02 June 1988           *
 4 *&Main Menu for Ad Hoc Report Module
 5 * NOTES: Menu choices must start with "A" *
 6 *        Choices limited to 26  (A-Z)      *
 7 * Public Variables Required:               *
 8 *    highlight                             *
 9 *    color                                 *
10 *    fns_retn                              *
11 * Procedures Required:                     *
12 *    FNSSCRN, FNSCENTER in FNSUTIL         *
13 * Return variables:                        *
14 *    fns_retn: numeric value representing *
15 *              selected choice from menu   *
16 ********************************************
17 SET PROCEDURE TO FNSUTIL
18 SET ESCAPE OFF
19 DO FNSSCRN WITH 'FNSADHM1'                                          &&;
19  MODIFY
20 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22    &&;
20  MODIFY
21 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23     &&;
21  MODIFY
22 scrtitle = 'AD HOC REPORT MENU'                                     &&;
22  MODIFY
23 max_ans = 4                                                         &&;
23  MODIFY
24 ans_1 = 'A) SELECT VARIABLES FOR EXTRACTS'
25 ans_2 = 'B) CREATE AD HOC EXTRACTS       '
26 ans_3 = 'C) EXIT TO DOS FOR AD HOC REPORT'
27 ans_4 = 'D) RETURN TO PREVIOUS MENU      '
28 mess_1 = '                                                '
29 mess_2 = '                                                '
30 mess_3 = 'Exit to DOS in order to use RR Report Writer'
31 mess_4 = '             Exit from this Menu                '
32
33 fns_retn = 0 && initialize return code
34 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
35 m_line = 20  && Prompt Message ROW()
36 col = 1      && Cur choice COL()
37 cmult = 22   && Column Incrementer
38 ctr = 1      && Row positioner
39 temp = LTRIM(RTRIM((STR(max_ans))))
40 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
41 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
42
43 SET ESCA OFF
44
45 ******************************
46 * PRINT SCREEN TITLE AND BOX *
47 ******************************
48 SET COLOR TO &highlight
49 @ row-2,32 SAY scrtitle
50 SET COLOR TO &color
51 @ row-1,20 TO max_ans+row+1,58 DOUBLE
52
53 ***********************
54 * PRINT ANSWER LINES *
55 ***********************
56 counter = 1
57 DO WHILE counter < max_ans + 1
58    x = LTRIM(STR(counter,2))
59    @ row+counter-1,23 SAY ans_&x
```

```
60     counter = counter + 1
61  ENDDO
62
63  **********************
64  * START OF MENU LOOP *
65  **********************
66  DO WHILE .T.
67     indx=LTRIM(STR(ctr,2))                  && Current Position of Cursor
68     @ row+(ctr-1),23 GET ans_&indx          && Read answer line
69
70     DO FNSCENTR WITH mess_&indx,m_line      && Print message line
71     * @ m_line,centr SAY mess_&indx
72     * centr = INT((80-LEN(mess_&indx))/2)
73     * @ m_line,2 SAY SPACE(75)
74     CLEAR GETS
75
76     key = 0                                 && Wait for user key
77     DO WHILE key = 0                        && Wait for user key
78        key = INKEY()                         && Wait for user key
79     ENDDO                                   && Wait for user key
80
81     ******************************************
82     * CONVERT lower case values to upper case *
83     ******************************************
84     key = IIF((key > 96 .AND. key < 123),(key-32),key)
85
86     @ m_line,3 CLEAR TO m_line,77           && Clear message line
87
88     *********************************
89     * Evaluate key that was pressed *
90     *********************************
91     DO CASE
92     CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
93        @ row+(ctr-1),23 SAY ans_&indx          && rewrites last selection
94        && in standard video
95        ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
96        ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
97        ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
98     CASE key = 13                  && <Enter>
99        fns_retn = ctr
100       EXIT
101    CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
102        * ignore
103    CASE key >=DECF .AND. key <= DECL    &&CHOICES
104       fns_retn = key - DECF + 1
105       EXIT
106    ENDCASE
107
108 ENDDO
109 *********************
110 * END OF MENU LOOP *
111 *********************
112 RETURN
```

VARIABLES CROSS-REF FOR FNSADHM1.PRG

Variable	Line number in file
ANS_:	59 68 93
ANS_1:	24
ANS_2:	25
ANS_3:	26
ANS_4:	27
CMULT:	37
COL:	36
COUNTER:	56 57 58 59 60
CTR:	38 67 95 96 97 99
DECF:	40 101 103 104
DECL:	41 101 103
FNSCENTR:	20 21 70
FNSSCRN:	19
FNSUTIL:	17
FNS_RETN:	33 99 104
INDX:	67 68 70 93
KEY:	76 78 84
MAX_ANS:	23 34 39 51 57 96 97
MESS_:	70
MESS_1:	28
MESS_2:	29
MESS_3:	30
MESS_4:	31
M_LINE:	35 70 86
ROW:	34 49 51 59 68 93
SCRTITLE:	22 49
TEMP:	39 41
X:	58 59

Table of Contents

```
 1 * FNSADHOC.PRG
 2 * written on 28 August 1988
 3 * modified on 11 Sept 1988
 4 * by David Edson
 5 *&Control Program for Ad Hoc Reporting Module
 6 SET PROCEDURE TO FNSUTIL
 7 SET ESCAPE OFF
 8 m_line = 20  && row for message line
 9 elements = 0 && reset to prevent carryover from display module
10
11 DO WHILE .T.
12
13    DO FNSADHM1
14
15    DO CASE
16    CASE fns_retn = 1
17       DO FNSADH1
18       LOOP
19    CASE fns_retn = 2
20       IF elements = 0
21          DO FNSBELL
22          @m_line,2 SAY SPACE(77)
23          @22,15 SAY "YOU MUST FLAG VARIABLES FROM DATA DICTIONARY FIRST"
24          DO FNSSLEEP
25          LOOP
26       ENDIF
27       DO FNSADH2
28       DO FNSADH4
29       LOOP   && FNSLOOK
30    CASE fns_retn = 3
31       DO FNSADH3  && NOT ENOUGH MEMORY TO ACCESS RR FROM WITHIN DBASE
32       QUIT
33    CASE fns_retn = 4
34       EXIT
35    ENDCASE
36 ENDDO
37 SET ESCAPE ON
38 RETURN
```

VARIABLES CROSS-REF FOR FNSADHOC.PRG

Variable	Line number in file
ELEMENTS:	9 20
FNSADH1:	17
FNSADH2:	27
FNSADH3:	31
FNSADH4:	28
FNSADHM1:	13
FNSBELL:	21
FNSSLEEP:	24
FNSUTIL:	6
FNS_RETN:	16 19 30 33
M_LINE:	8 22

Table of Contents

```
 1 ****************************************
 2 * FNSBACK.PRG                          *
 3 * written on 25 Sept 1988              *
 4 * by David Edson                       *
 5 *&Control Program for Data Backup
 6 ****************************************
 7
 8 SET PROCEDURE TO FNSUTIL
 9 SET ESCAPE OFF
10 SET TALK OFF
```

[redacted]

```
13    fns_retc = ''
14    DO FNSBDISK    && select backup drive
15    IF fns_retc $ 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
16       backdisk = fns_retc
17    ELSE
18       EXIT
19    ENDIF
20
21    **********************
22    DO FNSREQSP    && determine amount of backup space required
23    reqspace = fns_retn
24    **********************
25    DO FNSMENU9    && determine type of backup device
26    backdev  = fns_retn
27    @8,1 CLEAR TO 20,78
28    **********************
29    DO CASE
30    CASE backdev  = 1    &&360K disk
31       numdisks = INT(reqspace/360000)  + 1
32    CASE backdev  = 2    &&720K disk
33       numdisks = INT(reqspace/737000)  + 1
34    CASE backdev  = 3    &&1.2MB disk
35       numdisks = INT(reqspace/1220000) + 1
36    CASE backdev  = 4    &&1.44MB disk
37       numdisks = INT(reqspace/1440000) + 1
38    CASE backdev  = 5    &&Exit to Previous Screen
39       EXIT
40    ENDCASE
41
42    ****** Warning Message when more than one disk required *****
43    IF numdisks > 1
44       @8,1 CLEAR TO 20,78
45       DO FNSCENTR WITH 'WARNING!',12
46       tempchar = str(numdisks,2)
47       @14,20 SAY tempchar + ' blank, formatted diskettes are required.'
48       STORE SPACE(1) TO answer
49       STORE SPACE(1) TO ans2
50       @22,1 CLEAR TO 23,78
51       DO FNSCENTR WITH'Do you wish to continue with the data backup?';
51        + ' (Y/N)?',22
52       DO FNSCENTR WITH'Type [Y] to continue, [N] to Return to';
52        + ' Maintenance Menu',23
53       @23,69 GET answer PICTURE "Y"
54       READ
55       IF answer = "N"
56          RELEASE ALL
57          RETURN
58       ELSE
59          @22,1 CLEAR TO 23,78
60       ENDIF
61    ENDIF &&numdisks > 1
62    ******************************
```

```
63    @8,1 CLEAR TO 20,78
64    @22,1 CLEAR TO 23,78
65    DO FNSCENTR WITH 'Insert a blank formatted disk',12
66    @13,35 SAY 'in drive ' + backdisk
67    DO FNSCENTR WITH'Type [Y] to Begin Backup, [N] to Return to';
67     + ' Maintenance Menu',23
68    @23,71 GET ans2 PICTURE "Y"
69    READ
70    IF ans2 = "N"
71       RELEASE ALL
72       RETURN
73    ELSE
74       @22,1 CLEAR TO 23,78
75    ENDIF
76    @8,1 CLEAR TO 20,78
77    @22,1 CLEAR TO 23,78
78    DO FNSCENTR WITH 'Creating backup copies...',12
79    DO FNSCENTR WITH "Working...",22
80    USE BACKLIST  && file containing list of backup files
81    CALL trexe WITH "CURDRIVE"
82    old_def = tr_retc
83    SET DEFAULT TO &backdisk
84    SET ESCAPE ON
85    ON ESCAPE EXIT
86    @24,3 SAY '[Esc]=Abort'
87    DO WHILE .NOT. EOF()
88       **** check if enough space on backup disk
89       remain = DISKSPACE() - 2000   && add cushion to storage avail.
90       tempc1 = A->FILE_NAME
91       tempchar = "&old_def" - ":" - "&tempc1"
92       CALL trpass WITH tempchar
93       CALL trexe WITH "FILESIZE"
94       tempx1 = str(remain,8)
95       tempx2 = str(tr_retn,8)
96       @7,10 SAY tempchar
97       @8,10 SAY 'Remaining: ' + tempx1
98       @9,10 SAY 'Filesize:  ' + tempx2
99       IF tr_retn > remain
100         DO FNSBELL
101         DO FNSCENTR WITH 'Insert Next Backup Diskette...',12
102         @13,1 CLEAR TO 13,78
103         @22,1 CLEAR TO 23,78
104         DO FNSSLEEP
105         @22,1 CLEAR TO 23,78
106         DO FNSCENTR WITH 'Creating backup copies...',12
107         DO FNSCENTR WITH "Working...",22
108      ENDIF
109      filename = FILE_NAME
110      @13,28 SAY 'Backing Up File ' + FILE_NAME
111      SET CONSOLE OFF
112      RUN COPY &old_def:&filename &backdisk:&filename >NUL
113      SET CONSOLE ON
114      SKIP
115   ENDDO && copy loop
116   SET ESCAPE OFF
117   SET DEFAULT TO &old_def
118   **** FINISHED BACKUP
119   DO FNSSCRN WITH 'FNSBACK'
120   DO FNSBELL
121   DO FNSCENTR WITH 'Backup of Data Files Completed',12
122   DO FNSCENTR WITH 'Remove Backup Disk',22
123   DO FNSSLEEP
124   close DATABASES
125   SET ESCAPE ON
126   RETURN
127 ENDDO && main loop
```

Table of Contents

VARIABLES CROSS-REF FOR FNSBACK.PRG

Variable	Line number in file
A:	90
ANS2:	49 68 70
ANSWER:	48 53 55
BACKDEV:	26 30 32 34 36 38
BACKDISK:	16 66
BACKDISK::	112
BACKLIST:	80
BACKUP:	67
BACKUP?:	51
BEGIN:	67
CONTINUE:	51 52
COPY:	112
DO:	51
ESCAPE:	85
EXIT:	85
FILENAME:	109 112
FILE_NAME:	90 109 110
FNSBDISK:	14
FNSBELL:	100 120
FNSCENTR:	45 51 52 65 67 78 79 101 106 107 121 122
FNSMENU9:	25
FNSREQSP:	22
FNSSCRN:	119
FNSSLEEP:	104 123
FNSUTIL:	8
FNS_RETC:	13 15 16
FNS_RETN:	23 26
MAINTENANC:	52 67
MENU:	52 67
N:	51
NUL:	112
NUMDISKS:	31 33 35 37 43
OLD_DEF:	82
OLD_DEF::	112
REMAIN:	89 99
REQSPACE:	23
RETURN:	52 67
TEMPC1:	90
TEMPCHAR:	46 47 91 92 96
TEMPX1:	94 97
TEMPX2:	95 98
THE:	51
TREXE:	81 93
TRPASS:	92
TR_RETC:	82
TR_RETN:	99
WISH:	51
YOU:	51

```
 1 ******************************************
 2 * FNSBDISK.PRG                           *
 3 * written by Edson: 25 Sept 1988         *
 4 *&Display Menu to Select Backup Drive
 5 * Public Variables Required:             *
 6 *    highlight                           *
 7 *    color                               *
 8 *    fns_retc                            *
 9 * Procedures Required:                   *
10 *    FNSSCRN, FNSCENTER in FNSUTIL       *
11 * Return variables:                      *
12 *    fns_retc: letter of backup drive    *
13 ******************************************
14
15 SET ESCAPE OFF
16 SET BELL OFF
17 SET PROCEDURE TO FNSUTIL
18 *SET CONFIRM ON   &&2/18/89
19 DO FNSSCRN WITH 'FNSBDISK'
20 DO WHILE .T.
21    *** get backup disk drive ******
22    DO FNSCENTR WITH 'DISKETTE DRIVE TO BE USED FOR BACKUP/RESTORE:',10
23    STORE SPACE(1) TO b_drive
24    @24,3 SAY "[Esc]=Abort"
25    DO FNSCENTR WITH 'Enter Letter Of Drive To Be Used For Backup',22
26    DO FNSCENTR WITH '(A-Z)',23
27    ********* process response ******
28    @11,40 GET b_drive PICTURE "!"
29    READ
30    * IF READKEY() = 14 .OR. READKEY() = 270
31    IF READKEY() = 12 .OR. READKEY() = 268
32       STORE SPACE(1) TO answer
33       @22,1 CLEAR TO 23,78
34       DO FNSCENTR WITH "Do you wish to exit and return to previous";
34        + " menu (Y/N)?",22
35       DO FNSCENTR WITH "Type [Y] to exit, [N] to continue",23
36       @24,3 SAY REPLICATE(CHR(205),15)
37       @23,58 GET answer PICTURE "Y"
38       READ
39       IF answer = "Y"
40          RELEASE ALL
41          RETURN
42       ELSE
43          @22,1 CLEAR TO 23,78
44       ENDIF
45    ELSE
46       temp = ASC(b_drive)
47       IF temp < 65 .OR. temp > 90
48          LOOP
49       ELSE
50          fns_retc = b_drive
51          EXIT
52       ENDIF
53    ENDIF &&readkey
54 ENDDO
```

Table of Contents

VARIABLES CROSS-REF FOR FNSBDISK.PRG

Variable	Line number in file				
ANSWER:	32	37	39		
B_DRIVE:	23	28	50		
FNSCENTR:	22	25	26	34	35
FNSSCRN:	19				
FNSUTIL:	17				
FNS_RETC:	50				
TEMP:	46	47			

Table of Contents

```
 1 * FNSCAT.PRG *
 2 * by Richard Lieberman *
 3 * written on 7 June 1988 *
 4 *&Program to Process Display Category Selections
 5 * Modified - Edson 2/12/89
 6 *        2/28/89 - Check to make sure states were selected
 7 * Census Region Removed
 8
 9 DO FNSMENU2
10
11 DO CASE
12 CASE fns_retn = 1
13    DO FNSMENU3
14    disp_cat = 1
15    IF numtog = 0
16       disp_cat = 0
17    ENDIF
18 CASE fns_retn = 2
19    DO FNSMENU4
20    disp_choice = fns_retn
21    disp_cat = 2
22    IF disp_choice = 8
23       disp_cat = 0
24    ENDIF
25    *CASE fns_retn = 3
26    *  DO FNSMENU5
27    *  disp_choice = fns_retn
28    *  disp_cat = 3
29 CASE fns_retn = 3
30    disp_cat = 3
31 ENDCASE
32
33 RETURN  && FNSCAT
```

Table of Contents

VARIABLES CROSS-REF FOR FNSCAT.PRG

Variable	Line number in file				
DISP_CAT:	14	16	21	23	30
DISP_CHOIC:	20	22			
FNSMENU2:	9				
FNSMENU3:	13				
FNSMENU4:	19				
FNS_RETN:	12	18	20	29	
NUMTOG:	15				

```
 1 **********************************
 2 * FNSHELP.PRG                    *
 3 * Written by: EDSON 3/25/89      *
 4 * Purpose: Help control program *
 5 *&Help Facility Control Program *
 6 **********************************
 7 SET PROCEDURE TO FNSUTIL
 8
 9 ******************************
10 *** Store existing screen ****
11 cl = 'TEMP'
12 CALL trpass WITH cl
13 CALL trexe WITH "SAVESCR"
14 ******************************
15 DO WHILE .T.  && Start of main DO loop
16
17    DO FNSHELPM.PRG  && Main Help Menu
18
19    DO CASE
20    CASE fns_retn = 1     && Help for missing codes
21       DO FNSHELP1
22    CASE fns_retn = 2       && Multiple Responses
23       DO FNSHELP2
24    CASE fns_retn = 3       && Constructed Variables
25       DO FNSHELP3   &&menu for selecting constructed variables
26       DO FNSHELP4   &&text for constructed variables
27    CASE fns_retn = 4       && Return to Display Screen
28       EXIT
29    ENDCASE
30 ENDDO      && end of Main DO Loop
31
32 ****** restore original screen ***
33 cl = 'TEMP'
34 CALL trpass WITH cl
35 CALL trexe WITH "RESTSCR"
36
37 RETURN
```

VARIABLES CROSS-REF FOR FNSHELP.PRG

Variable	Line number in file			
C1:	11	12	33	34
FNSHELP1:	21			
FNSHELP2:	23			
FNSHELP3:	25			
FNSHELP4:	26			
FNSHELPM.P:	17			
FNSUTIL:	7			
FNS_RETN:	20	22	24	27
TREXE:	13	35		
TRPASS:	12	34		

Table of Contents

```
 1 **********************************
 2 * FNSHELP1.PRG                   *
 3 * Written by: EDSON 3/25/89      *
 4 * Purpose: Help message for      *
 5 *    missing value codes         *
 6 *&Help text for Missing Codes
 7 **********************************
 8 SET PROCEDURE TO FNSUTIL
 9 @0,0 CLEAR
10 @3,25 SAY "DISPLAY CODES FOR MISSING VALUES"
11 @4,25 SAY "-------------------------------"
12 @6,16  SAY "MIS  - Missing (Respondent did not supply "
13 @7,16  SAY "       information for this item)"
14 @9,16  SAY "NA   - Not Available (Information for this item"
15 @10,16 SAY "       was not available to the respondent)"
16 @12,16 SAY "REF  - Refused (Respondent refused to"
17 @13,16 SAY "       supply information on this item)"
18 @15,16 SAY "SKIP - Logical Skip in Questionnaire"
19 @16,16 SAY "       (Question not relevant to respondent)"
20 @18,16 SAY "DK   - Didn't Know (Respondent did not know"
21 @19,16 SAY "       answer to question)"
22
23 ******** Wait for User Response *******
24 @22,0 TO 24,79 DOUBLE
25 DO FNSCENTR WITH "Press [Return] to continue.",23
26 key = 0
27 DO WHILE key<>13
28    key = INKEY()
29 ENDDO
30 RETURN
```

VARIABLES CROSS-REF FOR FNSHELP1.PRG

Variable	Line number in file
FNSCENTR:	25
FNSUTIL:	8
KEY:	26 28

```
 1 *********************************
 2 * FNSHELP2.PRG                  *
 3 * Written by: EDSON 3/25/89     *
 4 * Purpose: Help message for     *
 5 *   Multiple Responses          *
 6 *&Help text for Multiple Responses
 7 *********************************
 8 SET PROCEDURE TO FNSUTIL
 9 @0,0 CLEAR
10 @3,31 SAY "MULTIPLE RESPONSES"
11 @4,31 SAY "------------------"
12 @6,16  SAY "   Multiple responses will be displayed for states"
13 @7,16  SAY "with more than one system.  Please refer to the written"
14 @8,16  SAY "documentation for information on the system and the"
15 @9,16  SAY "order in which they are displayed."
16
17 ******** Wait for User Response *******
18 @22,0 TO 24,79 DOUBLE
19 DO FNSCENTR WITH "Press [Return] to continue.",23
20 key = 0
21 DO WHILE key<>13
22    key = INKEY()
23 ENDDO
24 RETURN
```

VARIABLES CROSS-REF FOR FNSHELP2.PRG

Variable	Line number in file
FNSCENTR:	19
FNSUTIL:	8
KEY:	20 22

```
 1 ******************************************
 2 * FNSHELP3.PRG                           *
 3 * written by Edson: 02 June 1988         *
 4 * modified by Edson: 14 June 1988        *
 5 * Purpose: Help Menu/Constructed Variables*
 6 * NOTES: Menu choices must start with "A" *
 7 *     Choices limited to 26   (A-Z)      *
 8 * Public Variables Required:             *
 9 *    highlight                           *
10 *    color                               *
11 *    fns_retn                            *
12 * Procedures Required:                   *
13 *    FNSSCRN, FNSCENTER in FNSUTIL       *
14 * Return variables:                      *
15 *    fns_retn: numeric value representing *
16 *           selected choice from menu    *
17 ******************************************
18 *&Help menu for constructed variables
19
20 DO FNSSCRN WITH 'FNSHELP3'                                          &&;
20  MODIFY
21 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22  &&;
21  MODIFY
22 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23   &&;
22  MODIFY
23 scrtitle = ' CONSTRUCTED VARIABLES '    && MODIFY
24 max_ans = 8                                      && MODIFY
25 ans_1 = "A) CLASS86 (ACS)"
26 ans_2 = "B) Q310_A  (CLA)"
27 ans_3 = "C) Q317_A  (CLA)"
28 ans_4 = "D) ACCESS  (COM)"
29 ans_5 = "E) SPMATCH (COM)"
30 ans_6 = "F) Q90S    (SUR)"
31 ans_7 = "G) Q90T    (SUR)"
32 ans_8 = "H) RETURN TO PREVIOUS MENU"
33 mess_1 = '  State ACS Classification in 1986   '
34 mess_2 = '    Tracking System is Automated?   '
35 mess_3 = 'System for Aging Claims is Automated?'
36 mess_4 = '  Mode of Access for CM Procedures   '
37 mess_5 = '       Special Computer Match       '
38 mess_6 = '    Claims are suspended in state    '
39 mess_7 = '   Claims are terminated in state    '
40 mess_8 = '                                     '
41
42 fns_retn = 0 && initialize return code
43 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
44 m_line = 20  && Prompt Message ROW()
45 col = 1      && Cur choice COL()
46 cmult = 22   && Column Incrementer
47 ctr = 1      && Row positioner
48 temp = LTRIM(RTRIM((STR(max_ans))))
49 DECF = ASC(ans_1)   && Decimal equivalent of first selection index
50 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
51
52 SET ESCA OFF
53
54 ******************************
55 * PRINT SCREEN TITLE AND BOX *
56 ******************************
57 SET COLOR TO &highlight
58 @ row-2,29 SAY scrtitle
59 SET COLOR TO &color
60 @ row-1,20 TO max_ans+row+1,58 DOUBLE
61
```

```
 62 **********************
 63 * PRINT ANSWER LINES *
 64 **********************
 65 counter = 1
 66 DO WHILE counter < max_ans + 1
 67    x = LTRIM(STR(counter,2))
 68    @ row+counter-1,23 SAY ans_&x
 69    counter = counter + 1
 70 ENDDO
 71
 72 **********************
 73 * START OF MENU LOOP *
 74 **********************
 75 DO WHILE .T.
 76    indx=LTRIM(STR(ctr,2))            && Current Position of Cursor
 77    @ row+(ctr-1),23 GET ans_&indx    && Read answer line
 78
 79    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
 80    * @ m_line,centr SAY mess_&indx
 81    * centr = INT((80-LEN(mess_&indx))/2)
 82    * @ m_line,2 SAY SPACE(75)
 83    CLEAR GETS
 84
 85    key = 0                        && Wait for user key
 86    DO WHILE key = 0                    && Wait for user key
 87       key = INKEY()                 && Wait for user key
 88    ENDDO                     && Wait for user key
 89
 90    ******************************************
 91    * CONVERT lower case values to upper case *
 92    ******************************************
 93    key = IIF((key > 96 .AND. key < 123),(key-32),key)
 94
 95    @ m_line,3 CLEAR TO m_line,77     && Clear message line
 96
 97    ********************************
 98    * Evaluate key that was pressed *
 99    ********************************
100    DO CASE
101    CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
102       @ row+(ctr-1),23 SAY ans_&indx           && rewrites last selection
103       && in standard video
104       ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
105       ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
106       ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
107    CASE key = 13           && <Enter>
108       fns_retn = ctr
109       EXIT
110    CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
111        * ignore
112    CASE key >=DECF .AND. key <= DECL    &&CHOICES
113       fns_retn = key - DECF + 1
114       EXIT
115    ENDCASE
116
117 ENDDO
118 ********************
119 * END OF MENU LOOP *
120 ********************
121
122 RETURN
```

VARIABLES CROSS-REF FOR FNSHELP3.PRG

Variable	Line number in file
ANS_:	68 77 102
ANS_1:	25
ANS_2:	26
ANS_3:	27
ANS_4:	28
ANS_5:	29
ANS_6:	30
ANS_7:	31
ANS_8:	32
CMULT:	46
COL:	45
COUNTER:	65 66 67 68 69
CTR:	47 76 104 105 106 108
DECF:	49 110 112 113
DECL:	50 110 112
FNSCENTR:	21 22 79
FNSSCRN:	20
FNS_RETN:	42 108 113
INDX:	76 77 79 102
KEY:	85 87 93
MAX_ANS:	24 43 48 60 66 105 106
MESS_:	79
MESS_1:	33
MESS_2:	34
MESS_3:	35
MESS_4:	36
MESS_5:	37
MESS_6:	38
MESS_7:	39
MESS_8:	40
M_LINE:	44 79 95
ROW:	43 58 60 68 77 102
SCRTITLE:	23 58
TEMP:	48 50
X:	67 68

```
 1 *********************************
 2 * FNSHELP4.PRG                  *
 3 * Written by: EDSON 3/25/89     *
 4 * Purpose: Help message for     *
 5 *    constructed variables      *
 6 *&Help text for Constructed Variables
 7 *********************************
 8 SET PROCEDURE TO FNSUTIL
 9 @0,0 CLEAR
10 DO CASE
11 CASE fns_retn = 1  &&CLASS86
12    @2,5 SAY "CLASS86 - State ACS classification in 1986"
13    @3,5 SAY "----------------------------------------"
14    @4,7 SAY "1 = Batch & Basic Input/Recording"
15    @5,7 SAY "2 = Batch & Automated Results Check"
16    @6,7 SAY "3 = Batch & Stand Alone Elig/Benefit Det"
17    @7,7 SAY "4 = Batch & Integ det from Input Form"
18    @8,7 SAY "5 = Batch & App-based Det/Update"
19    @9,7 SAY "6 = Online & Basic Input/Recording"
20    @10,7 SAY "7 = Online & Automated Results Check"
21    @11,7 SAY "8 = Online & Stand Alone Elig/Benefit Det"
22    @12,7 SAY "9 = Online & Integ Det from Input Form"
23    @13,7 SAY "10 = Online & App-Based Det Update"
24    @14,7 SAY "11 = EW Online & Basic Input/ Recording"
25    @15,7 SAY "12 = EW Online & Automated Results Check"
26    @16,7 SAY "13 = EW Online & Stand Alone Elig/Benefit Det"
27    @17,7 SAY "14 = EW Online & Integ Det from Input Form"
28    @18,7 SAY "15 = EW Online & App-Based Det/Update"
29
30 CASE fns_retn = 2  &&Q310_A
31    @2,5 SAY "Q310_A - Tracking system is automated?"
32    @3,5 SAY "-------------------------------------"
33    @4,7 SAY "  0 = No"
34    @5,7 SAY "  1 = Yes"
35    @6,7 SAY "  2 = Partial"
36
37 CASE fns_retn = 3  &&Q317_A
38    @2,5 SAY "Q317_A - System for aging claims is automated?"
39    @3,5 SAY "-----------------------------------------------"
40    @4,7 SAY " 0 = No"
41    @5,7 SAY " 1 = Yes"
42    @6,7 SAY " 2 = Partial"
43
44 CASE fns_retn = 4  &&ACCESS
45    @2,5 SAY "ACCESS - Mode of access for computer matching procedures"
46    @3,5 SAY "-------------------------------------------------------"
47    @4,5 SAY " 0 = No Matching in State"
48    @5,5 SAY " 1 = All Batch Matching in State"
49    @6,5 SAY " 2 = Mostly Batch/Online Access to Nonwage"
50    @7,5 SAY " 3 = Both Batch/Online Access Inc Wage Data"
51
52 CASE fns_retn = 5  &&SPMATCH
53    @2,5 SAY "SPMATCH - Special computer match"
54    @3,5 SAY "--------------------------------"
55    @4,7 SAY "1 = SSN Validation"
56    @5,7 SAY "2 = Neighbor EI/FED File/Intrastate"
57    @6,7 SAY "3 = Asset Matach/Intrastate"
58    @7,7 SAY "4 = Neighbor Earned Income"
59    @8,7 SAY "5 = FNS Region Match/Intrastate"
60    @9,7 SAY "6 = FNS Regional Match"
61    @10,7 SAY "7 = Neighbor Public Assistance"
62    @11,7 SAY "8 = Neighbor PA/Asset Intrastate"
63    @12,7 SAY "9 = Neighbor PA/SSN Validation"
64    @13,7 SAY "10 = Neighbor EI/Asset Match"
```

```
65     @14,7 SAY "11 = Neighbor PA & EI/Assset/Intrastate"
66     @15,7 SAY "12 = Neighbor PA & EI"
67     @16,7 SAY "13 = Neighbor PA/Asset/FED File/Intrastate"
68     @17,7 SAY "14 = Neighbor PA & EI/Intrastate"
69     @18,7 SAY "15 = Federal File"
70     @19,7 SAY "16 = Neighbor PA & EI/Asset/FED/Intrastate"
71
72 CASE fns_retn = 6  &&Q90S
73     @2,5 SAY "Q90S - Claims are suspended in state"
74     @3,5 SAY "-------------------------------------"
75     @4,7 SAY " 0 = Claims not Suspended in this Jurisdiction"
76     @5,7 SAY " 1 = Claims are Suspended in this Jurisdiction"
77     @6,7 SAY " 2 = Suspension is State Level Responsibility"
78
79 CASE fns_retn = 7 &&Q90T
80     @2,5 SAY "Q90T - Claims are terminated in state"
81     @3,5 SAY "-------------------------------------"
82     @4,7 SAY " 0 = Claims not Terminated in this Jurisdiction"
83     @5,7 SAY " 1 = Claims are Terminated in this Jurisdiction"
84     @6,7 SAY " 2 = Termination is State Level Responsibility"
85
86 OTHERWISE
87     RETURN
88 ENDCASE
89 ******** Wait for User Response *******
90 @22,0 TO 24,79 DOUBLE
91 DO FNSCENTR WITH "Press [Return] to continue.",23
92 key = 0
93 DO WHILE key<>13
94     key = INKEY()
95 ENDDO
96 RETURN
```

VARIABLES CROSS-REF FOR FNSHELP4.PRG

Variable	Line number in file
FNSCENTR:	91
FNSUTIL:	8
FNS_RETN:	11 30 37 44 52 72 79
KEY:	92 94
OTHERWISE:	86

```
 1 ******************************************
 2 * FNSHELPM.PRG                           *
 3 * written by Edson: 02 June 1988         *
 4 * modified by Edson: 14 June 1988        *
 5 * Purpose: MAIN MENU                     *
 6 * NOTES: Menu choices must start with "A" *
 7 *     Choices limited to 26   (A-Z)      *
 8 * Public Variables Required:             *
 9 *    highlight                           *
10 *    color                               *
11 *    fns_retn                            *
12 * Procedures Required:                   *
13 *    FNSSCRN, FNSCENTER in FNSUTIL       *
14 * Return variables:                      *
15 *    fns_retn: numeric value representing *
16 *           selected choice from menu    *
17 ******************************************
18 *&Displays Main Help Menu
19
20 DO FNSSCRN WITH 'FNSHELPM'                                          &&;
20  MODIFY
21 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22  &&;
21  MODIFY
22 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23   &&;
22  MODIFY
23 scrtitle = ' MAIN HELP MENU '            && MODIFY
24 max_ans = 4                                    && MODIFY
25 ans_1 = "A) MISSING CODES"
26 ans_2 = "B) MULTIPLE RESPONSES"
27 ans_3 = "C) CONSTRUCTED VARIABLES"
28 ans_4 = "D) RETURN TO DISPLAY SCREEN"
29 mess_1 = '     Definitions of Missing Codes     '
30 mess_2 = '    Multiple Responses for a State    '
31 mess_3 = '   Values for Constructed Variables   '
32 mess_4 = '                                      '
33
34 fns_retn = 0 && initialize return code
35 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
36 m_line = 20  && Prompt Message ROW()
37 col = 1      && Cur choice COL()
38 cmult = 22   && Column Incrementer
39 ctr = 1      && Row positioner
40 temp = LTRIM(RTRIM((STR(max_ans))))
41 DECF = ASC(ans_1)   && Decimal equivalent of first selection index
42 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
43
44 SET ESCA OFF
45
46 ******************************
47 * PRINT SCREEN TITLE AND BOX *
48 ******************************
49 SET COLOR TO &highlight
50 @ row-2,32 SAY scrtitle
51 SET COLOR TO &color
52 @ row-1,20 TO max_ans+row+1,58 DOUBLE
53
54 ***********************
55 * PRINT ANSWER LINES *
56 ***********************
57 counter = 1
58 DO WHILE counter < max_ans + 1
59    x = LTRIM(STR(counter,2))
60    @ row+counter-1,23 SAY ans_&x
61    counter = counter + 1
```

```
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))            && Current Position of Cursor
   @ row+(ctr-1),23 GET ans_&indx    && Read answer line

   DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   * @ m_line,centr SAY mess_&indx
   * centr = INT((80-LEN(mess_&indx))/2)
   * @ m_line,2 SAY SPACE(75)
   CLEAR GETS

   key = 0                       && Wait for user key
   DO WHILE key = 0                   && Wait for user key
      key = INKEY()                && Wait for user key
   ENDDO                     && Wait for user key

   ******************************************
   * CONVERT lower case values to upper case *
   ******************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   @ m_line,3 CLEAR TO m_line,77     && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
      @ row+(ctr-1),23 SAY ans_&indx           && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13            && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
    ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************

RETURN
```

VARIABLES CROSS-REF FOR FNSHELPM.PRG

Variable	Line number in file						
ANS_:	60	69	94				
ANS_1:	25						
ANS_2:	26						
ANS_3:	27						
ANS_4:	28						
CMULT:	38						
COL:	37						
COUNTER:	57	58	59	60	61		
CTR:	39	68	96	97	98	100	
DECF:	41	102	104	105			
DECL:	42	102	104				
FNSCENTR:	21	22	71				
FNSSCRN:	20						
FNS_RETN:	34	100	105				
INDX:	68	69	71	94			
KEY:	77	79	85				
MAX_ANS:	24	35	40	52	58	97	98
MESS_:	71						
MESS_1:	29						
MESS_2:	30						
MESS_3:	31						
MESS_4:	32						
M_LINE:	36	71	87				
ROW:	35	50	52	60	69	94	
SCRTITLE:	23	50					
TEMP:	40	42					
X:	59	60					

```
 1 * FNSINST.PRG *
 2 * by David Edson       *
 3 * written on 12 Feb 1989*
 4 *&Initializes Public Variables Containing State Display info
 5
 6 state_1 = '   '  && initialize return codes from SELECTST.PRG
 7 state_2 = '   '  && initialize return codes from SELECTST.PRG
 8 state_3 = '   '  && initialize return codes from SELECTST.PRG
 9 state_4 = '   '  && initialize return codes from SELECTST.PRG
10 state_5 = '   '  && initialize return codes from SELECTST.PRG
11 state_6 = '   '  && initialize return codes from SELECTST.PRG
12 state_7 = '   '  && initialize return codes from SELECTST.PRG
13 state_8 = '   '  && initialize return codes from SELECTST.PRG
14 state_9 = '   '  && initialize return codes from SELECTST.PRG
15 state_10= '   '  && initialize return codes from SELECTST.PRG
16 RETURN
```

VARIABLES CROSS-REF FOR FNSINST.PRG

Variable	Line number in file
STATE_1:	6
STATE_10:	15
STATE_2:	7
STATE_3:	8
STATE_4:	9
STATE_5:	10
STATE_6:	11
STATE_7:	12
STATE_8:	13
STATE_9:	14

```
 1 * FNSLOOK.PRG *
 2 * written on 6 June 1988
 3 * by Richard Lieberman
 4 *&Control Program for Display Module
 5 SET PROCEDURE TO FNSUTIL
 6 m_line = 20     && position of print line
 7 elements = 0    && reset to prevent carryover from ad hoc module
 8
 9 DO WHILE .T.
10
11    DO FNSMENU1
12
13    DO CASE
14    CASE fns_retn = 1
15       DO FNSLOOK1
16       LOOP
17    CASE fns_retn = 2
18       DO FNSCAT
19       LOOP
20    CASE fns_retn = 3
21       IF elements = 0
22          DO FNSBELL
23          @m_line,2 SAY SPACE(77)
24          @22,15 SAY "YOU MUST FLAG VARIABLES FROM DATA DICTIONARY FIRST"
25          DO FNSSLEEP
26          LOOP
27       ENDIF
28       IF disp_cat = 0  && USER HAS NOT SELECTED DISPLAY CRITERIA
29          DO FNSBELL
30          @22,1 CLEAR TO 23,78
31          DO FNSCENTR WITH "DISPLAY CRITERIA NOT SELECTED",22
32          DO FNSSLEEP
33          LOOP
34       ENDIF
35       DO FNSLOOK2
36       DO FNSSCROL
37       LOOP   && FNSLOOK
38    CASE fns_retn = 4
39       IF elements = 0
40          DO FNSBELL
41          @22,1 CLEAR TO 23,78
42          DO FNSCENTR WITH "YOU MUST DISPLAY VARIABLES ON SCREEN";
42           + " BEFORE PRINTING",22
43          DO FNSSLEEP
44          LOOP
45       ENDIF
46       IF disp_cat = 0  && USER HAS NOT SELECTED DISPLAY CRITERIA
47          DO FNSBELL
48          @22,1 CLEAR TO 23,78
49          DO FNSCENTR WITH "YOU MUST DISPLAY VARIABLES ON SCREEN";
49           + " BEFORE PRINTING",22
50          DO FNSSLEEP
51          LOOP
52       ENDIF
53       DO FNSPRN1
54       LOOP
55    CASE fns_retn = 5
56       EXIT
57    ENDCASE
58 ENDDO
59
60 RETURN  && FNSLOOK
```

Table of Contents

VARIABLES CROSS-REF FOR FNSLOOK.PRG

Variable	Line number in file				
DISP_CAT:	28	46			
ELEMENTS:	7	21	39		
FNSBELL:	22	29	40	47	
FNSCAT:	18				
FNSCENTR:	31	42	49		
FNSLOOK1:	15				
FNSLOOK2:	35				
FNSMENU1:	11				
FNSPRN1:	53				
FNSSCROL:	36				
FNSSLEEP:	25	32	43	50	
FNSUTIL:	5				
FNS_RETN:	14	17	20	38	55
M_LINE:	6	23			

Table of Contents

```
 1 * FNSLOOK1.PRG *
 2 * written on 25 April 1988
 3 * modified: Edson 14 June 1988
 4 * by Richard Lieberman
 5 *&Flags Variables for Display
 6 SET PROCEDURE TO FNSUTIL
 7 SET ESCAPE OFF
 8 DO FNSSCRN WITH 'FNSLOOK1'
 9 arr_len = 81
10 numelt = 100
11 DO DEF_ARR WITH "VAR_LINE","STR",100,arr_len
12
13 STORE SUBSTR(array_id,1,8) TO argument
14 CALL trpass WITH argument
15 CALL trexe  WITH "SEPARATE"
16 segment = tr_retc
17 offset = tr_retn
18
19 USE DATADICT INDEX IDNUM,DATADICT,NAME
20
21 SET COLOR TO &highlight
22 @5,30 SAY "DATA DISPLAY MODULE"
23 @7,6 SAY "ID NUM"
24 @7,15 SAY "FIELD NAME"
25 @7,37 SAY "FIELD DESCRIPTION"
26 @7,70 SAY "SUBJECT"
27 SET COLOR TO &color
28
29 STORE 1 TO elements
30
31 prow = 9
32 @8,1 CLEAR TO 20,78
33
34 DO WHILE elements < 101
35
36
37    STORE SPACE(8) TO lookvar
38    @23,18 SAY "Enter Variable Name or Variable Number Above"
39    SET COLOR TO &highlight
40    @24,3 SAY "[Ctrl][W]=Exit"
41    SET COLOR TO &color
42
43    @22,2 SAY SPACE(77)
44    @22,35 GET lookvar PICTURE REPLICATE("!",8)
45    READ
46
47    IF READKEY() = 14 .OR. READKEY() = 270
48       STORE SPACE(1) TO answer
49       @22,1 CLEAR TO 23,78
50       DO FNSCENTR WITH "Do you wish to exit and return to the";
50        + " previous menu (Y/N)?",22
51       DO FNSCENTR WITH "Type [Y] to exit; [N] to continue selecting";
51        + " variables",23
52       @24,3 SAY REPLICATE(CHR(205),15)
53       @23,68 GET answer PICTURE "Y"
54       READ
55       IF answer = "Y"
56          RELEASE ALL
57          IF elements = 1     && reset elements in cases where
58             elements = 0     && user did not select any variables
59          ENDIF
60          RETURN
61       ELSE
62          @23,1 SAY SPACE(77)
```

Table of Contents

```
63          LOOP
64       ENDIF
65    ENDIF
66
67    IF .NOT. ISALPHA("&lookvar")
68       SET ORDER TO 1
69       SEEK VAL(SUBSTR(lookvar,1,10))
70       IF FOUND()
71          IF elements > 12
72             y = 19
73             x = elements - 2
74             @20,1 SAY SPACE(77)
75             prow = 20
76             DO WHILE y > 8
77                IF TYPE("offset") = "C"
78                   CALL trpass with offset
79                   CALL trexe with "DEC"
80                   offset = tr_retn
81                ENDIF
82                new_offset = offset + (x * arr_len)
83                CALL trpass WITH segment
84                CALL trpass WITH new_offset
85                CALL trexe WITH "PEEKSTR"
86                @y,5 SAY SUBSTR(tr_retc,1,4)
87                @y,15 SAY SUBSTR(tr_retc,5,10)
88                @y,30 SAY SUBSTR(tr_retc,15,40)
89                subval = '   '
90                tempsub = SUBSTR(tr_retc,55,2)
91                DO CASE
92                CASE tempsub = 'MR'
93                   subval = 'MR '
94                CASE tempsub = 'CL'
95                   subval = 'CLA'
96                CASE tempsub = 'CO'
97                   subval = 'COM'
98                CASE tempsub = 'AC'
99                   subval = 'ACS'
100               CASE tempsub = 'QC'
101                  subval = 'QC '
102               CASE tempsub = 'SU'
103                  subval = 'SUR'
104               ENDCASE
105               @y,72 SAY subval
106               x = x - 1
107               y = y - 1
108            ENDDO
109         ENDIF
110         @prow,5 SAY STR(IDNUM,4)
111         @prow,15 SAY FIELD_NAME
112         @prow,30 SAY DESCRIP
113         @prow,72 SAY TRIM(SUBJECT)
114         line = STR(IDNUM,4)+FIELD_NAME+DESCRIP+FILENAME
115         DO FIND_ARR WITH "VAR_LINE"
116         IF TYPE("offset") = "C"
117            CALL trpass with offset
118            CALL trexe with "DEC"
119            offset = tr_retn
120         ENDIF
121         new_offset = offset + ((elements-1) * arr_len)
122         CALL trpass WITH line
123         CALL trexe WITH "retc"
124         CALL trpass WITH segment
125         CALL trpass WITH new_offset
126         CALL trexe WITH TYPE("new_offset")
127         CALL trpass WITH tr_retc
128         CALL trexe WITH "POKESTR"
129         elements = elements + 1
```

```
130         prow = prow + 1
131      ELSE
132         DO FNSBELL
133         DO FNSCENTR WITH "*** ID NUMBER NOT FOUND ***",22
134         DO FNSSLEEP
135      ENDIF
136   ELSE
137      SET EXACT ON
138      COUNT FOR FIELD_NAME = "&lookvar" TO samevar
139      IF samevar > 1
140         cl = 'TEMP'
141         CALL trpass WITH cl
142         CALL trexe WITH "SAVESCR"
143         DO FNSMENU6
144         DO CASE
145         CASE fns_retn = 1
```

```
147         CASE fns_retn = 2
148            subject = 'QC'
149         CASE fns_retn = 3
150            subject = 'MR'
151         CASE fns_retn = 4
152            subject = 'CLA'
153         CASE fns_retn = 5
154            subject = 'COM'
155         CASE fns_retn = 6
156            subject = 'SUR'
157         ENDCASE
158         cl = 'TEMP'
159         CALL trpass WITH cl
160         CALL trexe WITH "RESTSCR"
161         SET ORDER TO 2
162         SEEK "&subject&lookvar"
163      ELSE
164         SET ORDER TO 3
165         SEEK "&lookvar"
166      ENDIF
167      IF FOUND()
168         IF elements > 12
169            y = 19
170            x = elements - 2
171            @20,1 SAY SPACE(77)
172            prow = 20
173            DO WHILE y > 8
174               IF TYPE("offset") = "C"
175                  CALL trpass with offset
176                  CALL trexe with "DEC"
177                  offset = tr_retn
178               ENDIF
179               new_offset = offset + (x * arr_len)
180               CALL trpass WITH segment
181               CALL trpass WITH new_offset
182               CALL trexe WITH "PEEKSTR"
183               @y,5 SAY SUBSTR(tr_retc,1,4)
184               @y,15 SAY SUBSTR(tr_retc,5,10)
185               @y,30 SAY SUBSTR(tr_retc,15,40)
186               subval = '    '
187               tempsub = SUBSTR(tr_retc,55,2)
188               DO CASE
189               CASE tempsub = 'MR'
190                  subval = 'MR  '
191               CASE tempsub = 'CL'
192                  subval = 'CLA'
193               CASE tempsub = 'CO'
194                  subval = 'COM'
195               CASE tempsub = 'AC'
196                  subval = 'ACS'
```

Table of Contents

```
197                CASE tempsub = 'QC'
198                   subval = 'QC '
199                CASE tempsub = 'SU'
200                   subval = 'SUR'
201                ENDCASE
202                @y,72 SAY subval
203                x = x - 1
204                y = y - 1
205             ENDDO
206          ENDIF
207          @prow,2 SAY SPACE(25)
208          @prow,5 SAY STR(IDNUM,4)
209          @prow,15 SAY FIELD_NAME
210          @prow,30 SAY DESCRIP
211          @prow,72 SAY TRIM(SUBJECT)
212          line = STR(IDNUM,4)+FIELD_NAME+DESCRIP+FILENAME
213          DO FIND_ARR WITH "VAR_LINE"
214          IF TYPE("offset") = "C"
215             CALL trpass with offset
216             CALL trexe with "DEC"
217             offset = tr_retn
218          ENDIF
219          new_offset = offset + ((elements-1) * arr_len)
220          CALL trpass WITH line
221          CALL trexe WITH "retc"
222          CALL trpass WITH segment
223          CALL trpass WITH new_offset
224          CALL trexe WITH TYPE("new_offset")
225          CALL trpass WITH tr_retc
226          CALL trexe WITH "POKESTR"
227          elements = elements + 1
228          prow = prow + 1
229       ELSE
230          DO FNSBELL
231          DO FNSCENTR WITH "*** VARIABLE NAME NOT FOUND ***",22
232          DO FNSSLEEP
233       ENDIF
234    ENDIF
235
236 ENDDO elements
237 RELEASE ALL
238 SET ESCAPE ON
239 RETURN
```

VARIABLES CROSS-REF FOR FNSLOOK1.PRG

Variable	Line number in file
ANSWER:	48 53 55
ARGUMENT:	13 14
ARR_LEN:	9 11 82 121 179 219
C1:	140 141 158 159
DATADICT:	19
DEF_ARR:	11
DESCRIP:	112 114 210 212
ELEMENTS:	29 34 57 58 71 73 121 129 168 170 219 227
FIELD_NAME:	111 114 138 209 212
FILENAME:	114 212
FIND_ARR:	115 213
FNSBELL:	132 230
FNSCENTR:	50 51 133 231
FNSMENU6:	143
FNSSCRN:	8
FNSSLEEP:	134 232
FNSUTIL:	6
FNS_RETN:	145 147 149 151 153 155
IDNUM:	19
LINE:	114 122 212 220
LOOKVAR:	37 44 67 69
NAME:	19
NEW_OFFSET:	82 84 121 125 126 179 181 219 223 224
NUMELT:	10
OFFSET:	17 77 78 80 82 116 117 119 121 174 175 177 179 214 215 217 219
PROW:	31 75 110 111 112 113 130 172 207 208 209 210 211 228
SAMEVAR:	138 139
SEGMENT:	16 83 124 180 222
SUBJECT:	146 148 150 152 154 156
SUBVAL:	89 93 95 97 99 101 103 105 186 190 192 194 196 198 200 202
TEMPSUB:	90 92 94 96 98 100 102 187 189 191 193 195 197 199
TREXE:	15 79 85 118 123 126 128 142 160 176 182 216 221 224 226
TRPASS:	14 78 83 84 117 122 124 125 127 141 159 175 180 181 215 220 222 223 225
TR_RETC:	16 127 225
TR_RETN:	17 80 119 177 217
X:	73 106 170 203
Y:	72 76 86 87 88 105 107 169 173 183 184 185 202 204

```
1  * FNSLOOK2.PRG              *
2  * written on 28 April 1988  *
3  * by Richard Lieberman      *
4  *  Modified: Edson 2/12/89  *
5  *    Removed Census Region  *
6  ******************************
7  *&Creates Display File for Data Display Module
8  SET ESCAPE OFF
9
10 ****************************************
11 ***  SET UP FILE CONTAINING DISPLAY DATA
12 ****************************************
13 SELECT C
14 USE DISPLAY
15 COPY TO DISPLAY2
16 USE DISPLAY2
17 ****************************************
18 ** insert begining of display marker **
19 ****************************************
20 APPEND BLANK
21 NEW_LINE = "***** Beginning of Display *****"
22 REPLACE LINE WITH NEW_LINE
23 ****************************************
24 SELECT A
25
26 DO FNSSCRN WITH 'FNSLOOK2'
27 DO FNSCENTR WITH "Working...",22
28 z = 1  && COUNTER FOR MULTIPLE OBSERVATIONS FOR A STATE
29 numelt = 1
30
31
32 DO WHILE numelt < elements
33
34    prow = .T.
35    numobs = .T.
36
37    DO WHILE numobs
38
39       IF disp_cat = 3
40          disp_choice = 1
41       ENDIF
42       more = .T.
43       times = 1  && NUMBER OF TIMES THROUGH numelt LOOP
44       maxobs = 1 && MAXIMUM NUMBER OF OBSERVATIONS FOR A DATA ITEM
45       DO WHILE more
46
47          DO WHILE prow
48             DO FIND_ARR WITH "VAR_LINE"
49             new_offset = offset + ((numelt-1) * arr_len)
50             CALL trpass WITH segment
51             CALL trpass WITH new_offset
52             CALL trexe WITH "PEEKSTR"
53             varnum = SUBSTR(tr_retc,1,4)
54             varname = SUBSTR(tr_retc,5,8)
55             vardes = SUBSTR(tr_retc,13,40)
56             varfile = SUBSTR(tr_retc,55,8)
57
58             @10,24 SAY 'Processing variable '
59             SET COLOR TO &highlight
60             @10,44 SAY varname
61             SET COLOR TO &color
62             @11,33 SAY "for state:"
63
64             * Checks to see if variable is a yes/no question
```

```
65              USE DATADICT INDEX IDNUM
66              SEEK VAL(varnum)
67              IF FOUND()
68                 discrete = YESNO
69              ELSE
70                 discrete = .F.
71              ENDIF
72              USE && CLOSE DATADICT.DBF
73
74              pcol = 1 && COLUMN POSITION
75              SELECT 1
76              USE FNSSTATE
77              SELECT 2
78              USE &varfile
79              SELECT 1
80              USE FNSSTATE
81              DO CASE
82              CASE disp_cat = 1 && DISPLAY SPECIFIC STATES
83                 numstate = numtog
84                 num = 1
85                 DO WHILE num < numstate + 1
86                    y = LTRIM(STR(num,2))
87                    LOCATE FOR ABBREV = state_&y
88                    code = STATECODE
89                    SELECT 2
90                    COUNT FOR STATE = code to numobs&y
91                    IF numobs&y > maxobs
92                       maxobs = numobs&y
93                    ENDIF
94                    SELECT 1
95                    num = num + 1
96                 ENDDO num
97              CASE disp_cat = 2 && DISPLAY FNS REGIONS
98                 COUNT FOR FNSREGION = disp_choice to numstate
99                 LOCATE FOR FNSREGION = disp_choice
100                num = 1
101                DO WHILE num < numstate + 1
102                   y = LTRIM(STR(num,2))
103                   state_&y = ABBREV
104                   code = STATECODE
105                   SELECT 2
106                   COUNT FOR STATE = code to numobs&y
107                   IF numobs&y > maxobs
108                      maxobs = numobs&y
109                   ENDIF
110                   SELECT 1
111                   CONTINUE
112                   num = num + 1
113                ENDDO num
114                *CASE disp_cat = 3   && DISPLAY CENSUS REGIONS
115                *  COUNT FOR CENSUSDIV = disp_choice to numstate
116                *  LOCATE FOR CENSUSDIV = disp_choice
117                *  num = 1
118                *  DO WHILE num < numstate + 1
119                *     y = LTRIM(STR(num,2))
120                *     state_&y = ABBREV
121                *     code = STATECODE
122                *     SELECT 2
123                *     COUNT FOR STATE = code to numobs&y
124                *     IF numobs&y > maxobs
125                *        maxobs = numobs&y
126                *     ENDIF
127                *     SELECT 1
128                *     CONTINUE
129                *     num = num + 1
130                *  ENDDO num
131             CASE disp_cat = 3 && DISPLAY ALL STATES (using FNS REGION)
```

```
132               COUNT FOR FNSREGION = disp_choice to numstate
133               LOCATE FOR FNSREGION = disp_choice
134               num = 1
135               DO WHILE num < numstate + 1
136                  y = LTRIM(STR(num,2))
137                  state_&y = ABBREV
138                  code = STATECODE
139                  SELECT 2
140                  COUNT FOR STATE = code to numobs&y
141                  IF numobs&y > maxobs
142                     maxobs = numobs&y
143                  ENDIF
144                  SELECT 1
145                  CONTINUE
146                  num = num + 1
147               ENDDO num
148            ENDCASE
149            ***********************************************
150            ** PRINT VARIABLE NAME AND DESCRIPTION     ***
151            ***********************************************
152            IF (((disp_cat <> 3).AND.(times=1)).OR. ((disp_cat;
152             =3).AND.(disp_choice=1).AND.(times=1)))
153               SELECT C
154               **** Insert blank line, then Variable Description and Name
155               NEW_LINE = ' '
156               APPEND BLANK
157               REPLACE LINE WITH NEW_LINE
158               NEW_LINE = TRIM(vardes) - " (" + TRIM(varname) + ")"
159               APPEND BLANK
160               APPEND BLANK
161               replace line with NEW_LINE
162               APPEND BLANK
163               SELECT A
164            ENDIF print
165            *********************************
166            * PRINT STATE NAMES ON SCREEN **
167            *********************************
168            IF times=1
169               new_line = ' '
170               SELECT C
171               APPEND BLANK
172               REPLACE LINE with new_line
173               new_line = SPACE(80)
174               j = 1
175               DO WHILE j < numstate + 1
176                  j1 = LTRIM(STR(j,2))
177                  new_line = stuff('&new_line',((j*8)-7),2,state_&j1)
178                  j = j + 1
179               ENDDO
180               APPEND BLANK
181               REPLACE LINE WITH new_line
182               SELECT A
183
184            ENDIF
185            **************************
186            new_line = SPACE(80)
187            **************************
188            DO WHILE pcol < numstate + 1
189               num = LTRIM(STR(pcol,2))
190               LOCATE FOR ABBREV = state_&num
191               code = STATECODE
192               statename = ABBREV
193               SELECT 1    && FNSSTATE
194               code = STATECODE
195               statename = ABBREV
196
197               SET COLOR TO &highlight
```

```
19:8              @11,44 SAY statename
19:9              SET COLOR TO &color
20:0
20:1              SELECT 2    && DATA FILE
20:2              * CHECK FOR MULTIPLE OBSERVATIONS FOR SAME STATE
20:3              IF times = 1
20:4                 COUNT FOR STATE = code to numobs&num
20:5                 dispobs&num = 1
20:6              ENDIF
20:7              *
20:8              IF numobs&num = 1
20:9                 LOCATE FOR STATE = code
21:0              ELSE
21:1                 LOCATE FOR STATE = code
21:2                 z = 1
21:3                 DO WHILE z < dispobs&num
21:4                    CONTINUE
21:5                    z = z + 1
21:6                 ENDDO z
21:7              ENDIF
21:8              IF TYPE("&varname") = "N"
21:9                 DO CASE
22:0                 CASE &varname = -9
22:1                    value = "MIS"
22:2                 CASE &varname = -5
22:3                    value = "MIS"
22:4                 CASE &varname = -4
22:5                    value = "NA"
22:6                 CASE &varname = -3
22:7                    value = "REF"
22:8                 CASE &varname = -2
22:9                    value = "SKIP"
23:0                 CASE &varname = -1
23:1                    value = "DK"
23:2                 CASE &varname = 0 .AND. discrete
23:3                    value = "NO"
23:4                 CASE &varname = 1 .AND. discrete
23:5                    value = "YES"
23:6                 OTHERWISE
23:7                    value = LTRIM(STR(&varname))
23:8                    IF LEN(value) > 7   && Truncate if display >;
23:8                      7 characters
23:9                       numval = VAL(value)
24:0                       lenval = LEN(value)
24:1                       chopval = LEN(value) - 5
24:2                       roundval = ROUND(numval,(chopval* (-1)))
24:3                       dispval = roundval / 10^chopval
24:4                       value   = LTRIM(STR(dispval)) + 'E' +;
24:4                         LTRIM(STR(chopval))
24:5                    ENDIF
24:6                 ENDCASE
24:7              ELSE
24:8                 DO CASE
24:9                 CASE &varname < "0"
25:0                    value = "NA"
25:1                 OTHERWISE
25:2                    value = LTRIM(&varname)
25:3                    IF LEN(value) > 7  && truncate if >7 characters
254                        value = LEFT(value,7)
255                     ENDIF
256                  ENDCASE
257               ENDIF
258               ************************************
259               ** PRINT VALUES                    **
260               ************************************
261               IF numobs&num > dispobs&num - 1
262                  IF (pcol*8)-7 > 1
```

```
263                     templine = SUBSTR(new_line,1,(pcol*8)-8)
264                     new_line = templine + LEFT(LTRIM(value)+SPACE(8),8)
265                  ELSE
266                     new_line = LEFT(LTRIM(value)+SPACE(8),8)
267                  ENDIF
268                  new_line = LEFT(new_line + SPACE(80),80)
269               ENDIF
270               dispobs&num = dispobs&num + 1
271               pcol = pcol + 1
272               SELECT 1
273               CONTINUE
274            ENDDO pcol
275            **********************
276            SELECT C
277            APPEND BLANK
278            REPLACE LINE WITH new_line
279            SELECT A
280            **********************
281            x = 1
282            times = times + 1
283            j = 0
284            DO WHILE x < numstate + 1
285               x1 = LTRIM(STR(x,2))
286               IF numobs&x1 > 1
287                  j = j + (numobs&x1 - (dispobs&num - 1))
288               ENDIF
289               x = x + 1
290            ENDDO x
291            IF j > 0
292               LOOP
293            ELSE
294               EXIT
295            ENDIF
296         ENDDO prow
297         IF disp_cat <> 3
298            more = .F.
299         ELSE
300            IF disp_choice < 7
301               disp_choice = disp_choice + 1
302               times = 1
303            ELSE
304               more = .F.
305            ENDIF
306         ENDIF
307      ENDDO more
308      numelt = numelt + 1
309      IF numelt > elements - 1
310         EXIT
311      ENDIF
312   ENDDO numobs
313   times = times + 1
314
315 ENDDO numelts
316 ***************************************
317 ** insert end of display marker      **
318 ***************************************
319 SELECT C
320 APPEND BLANK
321 NEW_LINE = "****** End of Display ******"
322 REPLACE LINE WITH NEW_LINE
323 ***************************************
324 SET ESCAPE ON
325 CLOSE DATABASES
```

VARIABLES CROSS-REF FOR FNSLOOK2.PRG

Variable	Line number in file
A:	24 163 182 279
ABBREV:	87 103 137 190 192 195
ARR_LEN:	49
C:	13 153 170 276 319
CHOPVAL:	241 243 244
CODE:	88 90 104 106 138 140 191 194 204 209 211
DATADICT:	65
DISCRETE:	68 70 232 234
DISPLAY:	14
DISPLAY2:	15 16
DISPOBS:	205 213 261 270
DISPVAL:	243 244
DISP_CAT:	39 82 97 131 152 297
DISP_CHOIC:	40 98 99 132 133 300 301
ELEMENTS:	32 309
FIND_ARR:	48
FNSCENTR:	27
FNSREGION:	98 99 132 133
FNSSCRN:	26
FNSSTATE:	76 80
IDNUM:	65
J:	174 175 176 177 178 283 287 291
J1:	176 177
LENVAL:	240
LINE:	22 157 161 172 181 278 322
MAXOBS:	44 91 92 107 108 141 142
MORE:	42 45 298 304
NEW_LINE:	21 22 155 157 158 169 172 173 177 181 186 264 266 268 278 321 322
NEW_OFFSET:	49 51
NUM:	84 85 86 95 100 101 102 112 134 135 136 146 189 190 204 205 208 213 261 270 287
NUMELT:	29 32 49 308 309
NUMOBS:	35 37 90 91 92 106 107 108 140 141 142 204 208 261 286
NUMSTATE:	83 85 98 101 132 135 175 188 284
NUMTOG:	83
NUMVAL:	239
OFFSET:	49
OTHERWISE:	236 251
PCOL:	74 188 189 271
PROW:	34 47
ROUNDVAL:	242 243
SEGMENT:	50
STATE:	90 106 140 204 209 211
STATECODE:	88 104 138 191 194
STATENAME:	192 195 198
STATE_:	87 103 137 177 190
TEMPLINE:	263 264
TIMES:	43 168 203 282 302 313
TREXE:	52
TRPASS:	50 51
VALUE:	221 223 225 227 229 231 233 235 237 244 250 252 254 264 266
VARDES:	55
VARFILE:	56 78
VARNAME:	54 60 218 220 222 224 226 228 230 232 234 237 249 252
VARNUM:	53
X:	281 284 285 289
X1:	285 286 287

Table of Contents

```
Y:          86   87   90   91   92  102  103  106  107  108  136  137  140
           141  142
YESNO:      68
Z:          28  212  213  215
```

Table of Contents

```
 1 * FNSMAIN.PRG *
 2 * written on 17 March 1988 *
 3 * modified: Edson 14 June 1988 *
 4 *   added "Maintenance selection" to menu *
 5 *&Main Control Program
 6
 7 * set program parameters *
 8 SET ESCAPE OFF
 9 SET ECHO OFF
10 SET CONFIRM OFF
11 SET EXACT ON
12 SET STATUS OFF
13 SET SCOREBOARD OFF
14 SET BELL OFF
15 SET DEVICE TO SCREEN
16 SET STEP OFF
17 SET TALK OFF
18 SET SAFETY OFF
19
20
21 SET PROCEDURE TO FNSUTIL
22
23
24 DO FNSMEM && declare public memory variables
25 DO FNSVAR
26
27 mdate=DATE()
28
29 displayvar = .T.
30 editvar = .T.
31 mainmenu = .T.
32
33 CLEAR
34
35 IF .NOT. ISCOLOR()
36    STORE "" TO color
37    STORE "I" TO highlight
38 ELSE
39    STORE "BG/N,W/R,N" TO color
40    STORE "GR+/N,W/R,N" TO highlight
41 ENDIF
42
43 SET COLOR TO &color
44
45 DO PREPARE
46 DO DEF_ARR WITH "WINDOW","SCR",1,4096
47 x = 1
48 DO WHILE x < 6
49    y = STR(x,1)
50    DO DEF_ARR WITH 'SCREEN&y','SCR',1,4096
51    x = x + 1
52 ENDDO x
53
54
55 DO WHILE mainmenu
56
57    @0,0 CLEAR
58
59    DO FNSMENU0
60
61    DO WHILE displayvar
62
63       DO CASE
64       CASE fns_retn = 1
```

```
65          DO FNSLOOK
66          EXIT
67       CASE fns_retn = 2
68          DO FNSUPD1
69          EXIT
70       CASE fns_retn = 3
71          DO FNSSTD
72          EXIT
73       CASE fns_retn = 4
74          DO FNSADHOC
75          EXIT
76       CASE fns_retn = 5
77          DO FNSMAINT
78          EXIT
79       CASE fns_retn = 6
80          EXIT
81       CASE fns_retn = 7
82          CLOSE DATABASES
83          QUIT
84       ENDCASE
85    ENDDO
86
87    IF fns_retn = 6
88       EXIT
89    ENDIF
90
91 ENDDO
92 @5,1 CLEAR TO 20,78
93 @22,1 CLEAR TO 23,78
94 @12,20 SAY "CLOSING ALL FILES........RETURNING TO DOS"
95
96
97 CLOSE DATABASES
98
99 * clear variables and return to dBASE system
100 SET ESCAPE ON
101 SET BELL ON
102 SET HEADING ON
103 SET STATUS ON
104 CLEAR
105
106 RETURN
```

VARIABLES CROSS-REF FOR FNSMAIN.PRG

Variable	Line number in file							
DEF_ARR:	46	50						
DISPLAYVAR:	29	61						
EDITVAR:	30							
FNSADHOC:	74							
FNSLOOK:	65							
FNSMAINT:	77							
FNSMEM:	24							
FNSMENU0:	59							
FNSSTD:	71							
FNSUPD1:	68							
FNSUTIL:	21							
FNSVAR:	25							
FNS_RETN:	64	67	70	73	76	79	81	87
HIGHLIGHT:	37	40						
MAINMENU:	31	55						
MDATE:	27							
PREPARE:	45							
X:	47	48	51					
Y:	49							

```
 1 * FNSMAINT.PRG *
 2 * written on 25 Sept 1988
 3 * by David Edson
 4 *&Control Program for Maintenance Module
 5 SET PROCEDURE TO FNSUTIL
 6 m_line = 20    && position of print line
 7
 8 DO WHILE .T.
 9
10    DO FNSMENU8
11
12    DO CASE
13    CASE fns_retn = 1
14       DO FNSBACK
15       LOOP
16    CASE fns_retn = 2
17       DO FNSREST
18       LOOP
19    CASE fns_retn = 3
20       DO FNSPRN2
21       LOOP
22    CASE fns_retn = 4
23       EXIT
24    ENDCASE
25 ENDDO
26
27 RETURN  && FNSLOOK
```

VARIABLES CROSS-REF FOR FNSMAINT.PRG

Variable	Line number in file
FNSBACK:	14
FNSMENU8:	10
FNSPRN2:	20
FNSREST:	17
FNSUTIL:	5
FNS_RETN:	13 16 19 22
M_LINE:	6

```
 1 * FNSMEM.PRG *
 2 * by R. Lieberman *
 3 * written on 17 March 1988 *
 4 *&Creates Public Memory Variables
 5
 6 PUBLIC clipper
 7 PUBLIC color,highlight
 8 PUBLIC segment,offset
 9 PUBLIC debug
10 PUBLIC mdate
11 PUBLIC elements   && FNSLOOK1: number of data dictionary items flagged
12 PUBLIC disp_cat  && display category
13 PUBLIC disp_choice  && display choice
14 PUBLIC array_id
15 PUBLIC num_array    && number of arrays defined
16 PUBLIC arr_len,numelt
17 PUBLIC subject && subject to display
18 PUBLIC indx  && choice variable for menus
19 PUBLIC rtcode && return code variable
20 PUBLIC fns_retn,fns_retc,fns_retl
21 PUBLIC state_1,state_2,state_3,state_4,state_5,state_6,state_7,state_8
22 PUBLIC state_9,state_10,numtog
23 PUBLIC lookvar,code,numsys,rowsys
24 PUBLIC rptnum   && report number for standard reports
25 RETURN
```

VARIABLES CROSS-REF FOR FNSMEM.PRG

Variable	Line number in file
ARRAY_ID:	14
ARR_LEN:	16
CLIPPER:	6
CODE:	23
DEBUG:	9
DISP_CAT:	12
DISP_CHOIC:	13
ELEMENTS:	11
FNS_RETC:	20
FNS_RETL:	20
FNS_RETN:	20
HIGHLIGHT:	7
INDX:	18
LOOKVAR:	23
MDATE:	10
NUMELT:	16
NUMSYS:	23
NUMTOG:	22
NUM_ARRAY:	15
OFFSET:	8
ROWSYS:	23
RPTNUM:	24
RTCODE:	19
SEGMENT:	8
STATE_1:	21
STATE_10:	22
STATE_2:	21
STATE_3:	21
STATE_4:	21
STATE_5:	21
STATE_6:	21
STATE_7:	21
STATE_8:	21
STATE_9:	22
SUBJECT:	17

```
1  ********************************************
2  * FNSMENU0.PRG                             *
3  * written by Edson: 02 June 1988           *
4  * modified by Edson: 14 June 1988          *
5  * Purpose: MAIN MENU                       *
6  * NOTES: Menu choices must start with "A"  *
7  *     Choices limited to 26   (A-Z)        *
8  * Public Variables Required:               *
9  *    highlight                             *
10 *    color                                 *
11 *    fns_retn                              *
12 * Procedures Required:                     *
13 *    FNSSCRN, FNSCENTER in FNSUTIL         *
14 * Return variables:                        *
15 *    fns_retn: numeric value representing  *
16 *          selected choice from menu       *
17 ********************************************
18 *&Displays Main Menu
19
20 DO FNSSCRN WITH 'FNSMENU0'                                          &&;
20  MODIFY
21 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22    &&;
21  MODIFY
22 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23     &&;
22  MODIFY
23 scrtitle = 'MAIN MENU'                                             &&;
23  MODIFY
24 max_ans = 7                                        && MODIFY
25 ans_1 = "A) DATA DISPLAY MODULE"
26 ans_2 = "B) UPDATE MODULE"
27 ans_3 = "C) STANDARD REPORTS"
28 ans_4 = "D) AD HOC REPORTS"
29 ans_5 = "E) SYSTEM MAINTENANCE"
30 ans_6 = "F) QUIT TO DBASE III PLUS"
31 ans_7 = "G) EXIT TO DISK OPERATING SYSTEM"
32 mess_1 = '     Display Data Items on Screen      '
33 mess_2 = '     Change Values of Data Items       '
34 mess_3 = '        Print Standard Reports         '
35 mess_4 = ' Prepare for Creating Ad Hoc Reports   '
36 mess_5 = '    Perform Maintenance Procedures     '
37 mess_6 = '     Stop Program, Return to dBase     '
38 mess_7 = '     Stop Program, Return to DOS       '
39
40 fns_retn = 0 && initialize return code
41 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
42 m_line = 20  && Prompt Message ROW()
43 col = 1      && Cur choice COL()
44 cmult = 22   && Column Incrementer
45 ctr = 1      && Row positioner
46 temp = LTRIM(RTRIM((STR(max_ans))))
47 DECF = ASC(ans_1)   && Decimal equivalent of first selection index
48 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
49
50 SET ESCA OFF
51
52 *******************************
53 * PRINT SCREEN TITLE AND BOX *
54 *******************************
55 SET COLOR TO &highlight
56 @ row-2,32 SAY scrtitle
57 SET COLOR TO &color
58 @ row-1,20 TO max_ans+row+1,58 DOUBLE
59
60 ***********************
```

```
 61 * PRINT ANSWER LINES *
 62 ************************
 63 counter = 1
 64 DO WHILE counter < max_ans + 1
 65    x = LTRIM(STR(counter,2))
 66    @ row+counter-1,23 SAY ans_&x
 67    counter = counter + 1
 68 ENDDO
 69
 70 ************************
 71 * START OF MENU LOOP *
 72 ************************
 73 DO WHILE .T.
 74    indx=LTRIM(STR(ctr,2))            && Current Position of Cursor
 75    @ row+(ctr-1),23 GET ans_&indx    && Read answer line
 76
 77    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
 78    * @ m_line,centr SAY mess_&indx
 79    * centr = INT((80-LEN(mess_&indx))/2)
 80    * @ m_line,2 SAY SPACE(75)
 81    CLEAR GETS
 82
 83    key = 0                       && Wait for user key
 84    DO WHILE key = 0                   && Wait for user key
 85       key = INKEY()                && Wait for user key
 86    ENDDO                      && Wait for user key
 87
 88    *******************************************
 89    * CONVERT lower case values to upper case *
 90    *******************************************
 91    key = IIF((key > 96 .AND. key < 123),(key-32),key)
 92
 93    @ m_line,3 CLEAR TO m_line,77     && Clear message line
 94
 95    ********************************
 96    * Evaluate key that was pressed *
 97    ********************************
 98    DO CASE
 99    CASE STR(key,2)$" 5,24"               && Dn/Up Arrows
100       @ row+(ctr-1),23 SAY ans_&indx           && rewrites last selection
101       && in standard video
102       ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
103       ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
104       ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
105    CASE key = 13            && <Enter>
106       fns_retn = ctr
107       EXIT
108    CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
109        * ignore
110    CASE key >=DECF .AND. key <= DECL    &&CHOICES
111       fns_retn = key - DECF + 1
112       EXIT
113    ENDCASE
114
115 ENDDO
116 ********************
117 * END OF MENU LOOP *
118 ********************
119 SET ESCAPE ON
120 RETURN
```

VARIABLES CROSS-REF FOR FNSMENU0.PRG

Variable	Line number in file
ANS_:	66 75 100
ANS_1:	25
ANS_2:	26
ANS_3:	27
ANS_4:	28
ANS_5:	29
ANS_6:	30
ANS_7:	31
CMULT:	44
COL:	43
COUNTER:	63 64 65 66 67
CTR:	45 74 102 103 104 106
DECF:	47 108 110 111
DECL:	48 108 110
FNSCENTR:	21 22 77
FNSSCRN:	20
FNS_RETN:	40 106 111
INDX:	74 75 77 100
KEY:	83 85 91
MAX_ANS:	24 41 46 58 64 103 104
MESS_:	77
MESS_1:	32
MESS_2:	33
MESS_3:	34
MESS_4:	35
MESS_5:	36
MESS_6:	37
MESS_7:	38
M_LINE:	42 77 93
ROW:	41 56 58 66 75 100
SCRTITLE:	23 56
TEMP:	46 48
X:	65 66

```
 1 ******************************************
 2 * FNSMENU1.PRG                           *
 3 * written by Edson: 02 June 1988         *
 4 *&Displays Main Menu for Data Display Module
 5 * NOTES: Menu choices must start with "A" *
 6 *        Choices limited to 26  (A-Z)    *
 7 * Public Variables Required:             *
 8 *    highlight                           *
 9 *    color                               *
10 *    fns_retn                            *
11 * Procedures Required:                   *
12 *    FNSSCRN, FNSCENTER in FNSUTIL       *
13 * Return variables:                      *
14 *    fns_retn: numeric value representing *
15 *              selected choice from menu  *
16 ******************************************
17 SET PROCEDURE TO FNSUTIL
18
19 DO FNSSCRN WITH 'FNSMENU1'                                          &&;
19  MODIFY
20 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22    &&;
20  MODIFY
21 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23     &&;
21  MODIFY
22 scrtitle = 'DATA DISPLAY MENU'                                      &&;
22  MODIFY
23 max_ans = 5                                                         &&;
23  MODIFY
24 ans_1 = 'A) SELECT VARIABLES FOR DISPLAY'
25 ans_2 = 'B) CHOOSE DISPLAY CRITERIA'
26 ans_3 = 'C) DISPLAY VARIABLES ON SCREEN'
27 ans_4 = 'D) PRINT MOST RECENT DISPLAY'
28 ans_5 = 'E) RETURN TO PREVIOUS MENU'
29 mess_1 = '                                        '
30 mess_2 = '                                        '
31 mess_3 = '                                        '
32 mess_4 = 'Print results from most recent request'
33 mess_5 = '       Exit from this Menu              '
34
35 fns_retn = 0 && initialize return code
36 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
37 m_line = 20  && Prompt Message ROW()
38 col = 1      && Cur choice COL()
39 cmult = 22   && Column Incrementer
40 ctr = 1      && Row positioner
41 temp = LTRIM(RTRIM((STR(max_ans))))
42 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
43 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
44
45 SET ESCA OFF
46
47 ******************************
48 * PRINT SCREEN TITLE AND BOX *
49 ******************************
50 SET COLOR TO &highlight
51 @ row-2,32 SAY scrtitle
52 SET COLOR TO &color
53 @ row-1,20 TO max_ans+row+1,58 DOUBLE
54
55 ***********************
56 * PRINT ANSWER LINES *
57 ***********************
58 counter = 1
59 DO WHILE counter < max_ans + 1
```

```
60    x = LTRIM(STR(counter,2))
61    @ row+counter-1,23 SAY ans_&x
62    counter = counter + 1
63 ENDDO
64
65 ***********************
66 * START OF MENU LOOP *
67 ***********************
68 DO WHILE .T.
69    indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
70    @ row+(ctr-1),23 GET ans_&indx         && Read answer line
71
72    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
73    * @ m_line,centr SAY mess_&indx
74    * centr = INT((80-LEN(mess_&indx))/2)
75    * @ m_line,2 SAY SPACE(75)
76    CLEAR GETS
77
78    key = 0                                && Wait for user key
79    DO WHILE key = 0                       && Wait for user key
80       key = INKEY()                        && Wait for user key
81    ENDDO                                  && Wait for user key
82
83    ******************************************
84    * CONVERT lower case values to upper case *
85    ******************************************
86    key = IIF((key > 96 .AND. key < 123),(key-32),key)
87
88    @ m_line,3 CLEAR TO m_line,77          && Clear message line
89
90    *********************************
91    * Evaluate key that was pressed *
92    *********************************
93    DO CASE
94    CASE STR(key,2)$" 5,24"               && Dn/Up Arrows
95       @ row+(ctr-1),23 SAY ans_&indx        && rewrites last selection
96       && in standard video
97       ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
98       ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
99       ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
100   CASE key = 13                   && <Enter>
101      fns_retn = ctr
102      EXIT
103   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
104       * ignore
105   CASE key >=DECF .AND. key <= DECL    &&CHOICES
106      fns_retn = key - DECF + 1
107      EXIT
108    ENDCASE
109
110 ENDDO
111 *********************
112 * END OF MENU LOOP *
113 *********************
114 SET ESCAPE ON
115 RETURN
```

VARIABLES CROSS-REF FOR FNSMENU1.PRG

Variable	Line number in file						
ANS_:	61	70	95				
ANS_1:	24						
ANS_2:	25						
ANS_3:	26						
ANS_4:	27						
ANS_5:	28						
CMULT:	39						
COL:	38						
COUNTER:	58	59	60	61	62		
CTR:	40	69	97	98	99	101	
DECF:	42	103	105	106			
DECL:	43	103	105				
FNSCENTR:	20	21	72				
FNSSCRN:	19						
FNSUTIL:	17						
FNS_RETN:	35	101	106				
INDX:	69	70	72	95			
KEY:	78	80	86				
MAX_ANS:	23	36	41	53	59	98	99
MESS_:	72						
MESS_1:	29						
MESS_2:	30						
MESS_3:	31						
MESS_4:	32						
MESS_5:	33						
M_LINE:	37	72	88				
ROW:	36	51	53	61	70	95	
SCRTITLE:	22	51					
TEMP:	41	43					
X:	60	61					

```
 1 *******************************************
 2 * FNSMENU2.PRG                            *
 3 * written by Edson: 02 June 1988          *
 4 *&Displays Menu for Display Category Selection
 5 * NOTES: Menu choices must start with "A" *
 6 *        Choices limited to 26  (A-Z)     *
 7 *   Removed Census Region choice          *
 8 *      Edson 2/12/89                      *
 9 * Public Variables Required:              *
10 *    highlight                            *
11 *    color                                *
12 *    fns_retn                             *
13 * Procedures Required:                    *
14 *    FNSSCRN, FNSCENTER in FNSUTIL        *
15 * Return variables:                       *
16 *    fns_retn: numeric value representing *
17 *              selected choice from menu  *
18 *******************************************
19 SET PROCEDURE TO FNSUTIL
20 DO FNSINST && initialize state display codes  Edson 2/12/89
21 DO FNSSCRN WITH 'FNSMENU2'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 scrtitle = "DISPLAY CATEGORY MENU"
25 max_ans = 4                                                    &&;
25  MODIFY
26 ans_1 = 'A) SELECTED STATES'
27 ans_2 = 'B) FNS REGION'
28 * ans_3 = 'C) CENSUS DIVISIONS'             &&EDSON 2/12/89
29 ans_3 = 'C) ALL STATES'
30 ans_4 = 'D) RETURN TO PREVIOUS MENU'
31 mess_1 = 'View Data for Up to Nine States (Regardless of Region)'
32 mess_2 = '              View Data for a FNS Region              '
33 * mess_3 = '            View Data for Census Divisions           '
34 mess_3 = '    View Data for All States (in FNS Region Order)    '
35 mess_4 = '                  Exit from this Menu                '
36
37 fns_retn = 0 && initialize return code
38 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
39 m_line = 20  && Prompt Message ROW()
40 col = 1      && Cur choice COL()
41 cmult = 22   && Column Incrementer
42 ctr = 1      && Row positioner
43 temp = LTRIM(RTRIM((STR(max_ans))))
44 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
45 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
46
47 SET ESCA OFF
48
49 ******************************
50 * PRINT SCREEN TITLE AND BOX *
51 ******************************
52 SET COLOR TO &highlight
53 @ row-2,32 SAY scrtitle
54 SET COLOR TO &color
55 @ row-1,20 TO max_ans+row+1,58 DOUBLE
56
57 ***********************
58 * PRINT ANSWER LINES *
59 ***********************
60 counter = 1
61 DO WHILE counter < max_ans + 1
62    x = LTRIM(STR(counter,2))
63    @ row+counter-1,23 SAY ans_&x
```

```
64    counter = counter + 1
65 ENDDO
66
67 **********************
68 * START OF MENU LOOP *
69 **********************
70 DO WHILE .T.
71    indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
72    @ row+(ctr-1),23 GET ans_&indx         && Read answer line
73    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
74    CLEAR GETS
75
76    key = 0                                && Wait for user key
77    DO WHILE key = 0                       && Wait for user key
78       key = INKEY()                        && Wait for user key
79    ENDDO                                  && Wait for user key
80
81    ********************************************
```

[redacted]

```
84    key = IIF((key > 96 .AND. key < 123),(key-32),key)
85
86    @ m_line,3 CLEAR TO m_line,77          && Clear message line
87
88    *********************************
89    * Evaluate key that was pressed *
90    *********************************
91
92    DO CASE
93    CASE STR(key,2)$" 5,24"               && Dn/Up Arrows
94       @ row+(ctr-1),23 SAY ans_&indx        && rewrites last selection
95       && in standard video
96       ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
97       ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
98       ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
99    CASE key = 13                   && <Enter>
100      fns_retn = ctr
101      EXIT
102   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
103       * ignore
104   CASE key >=DECF .AND. key <= DECL    &&CHOICES
105      fns_retn = key - DECF + 1
106      EXIT
107   ENDCASE
108
109 ENDDO
110 *********************
111 * END OF MENU LOOP *
112 *********************
113
114 SET ESCAPE ON
```

VARIABLES CROSS-REF FOR FNSMENU2.PRG

Variable	Line number in file
ANS_:	63 72 94
ANS_1:	26
ANS_2:	27
ANS_3:	29
ANS_4:	30
CMULT:	41
COL:	40
COUNTER:	60 61 62 63 64
CTR:	42 71 96 97 98 100
DECF:	44 102 104 105
DECL:	45 102 104
FNSCENTR:	22 23 73
FNSINST:	20
FNSSCRN:	21
FNSUTIL:	19
FNS_RETN:	37 100 105
INDX:	71 72 73 94
KEY:	76 78 84
MAX_ANS:	25 38 43 55 61 97 98
MESS_:	73
MESS_1:	31
MESS_2:	32
MESS_3:	34
MESS_4:	35
M_LINE:	39 73 86
ROW:	38 53 55 63 72 94
SCRTITLE:	24 53
TEMP:	43 45
X:	62 63

```
 1 ******************************************
 2 * FNSMENU3.PRG                           *
 3 * written by Edson: 02 June 1988         *
 4 * modified by Edson: 14 June 1988        *
 5 *&State Select Menu
 6 * NOTES: Menu choices must start with "A" *
 7 *     Choices limited to 26   (A-Z)      *
 8 * Public Variables Required:             *
 9 *    highlight                           *
10 *    color                               *
11 *    state_1 through state_9             *
12 *    numtog                              *
13 *    disp_choice                         *
14 * Procedures Required:                   *
15 *    FNSSCRN, FNSCENTER in FNSUTIL       *
16 * Return variables:                      *
17 *    state_1 through state_9             *
18 *    numtog                              *
19 *    disp_choice                         *
20 ******************************************
21
22 SET PROCEDURE TO FNSUTIL
23
24 DO FNSSCRN WITH 'FNSMENU3'
25
26 scrtitle = "STATE SELECT MENU"
27
28 togon = '#'
29 togoff = ' '
30 maxtog = 9
31 numtog = 0
32 disp_choice = 0   && EDSON 2/26/89
33
34 ctrcol = "A"    && initialize cursor column position
35 ctrrow = "A"    && initialize cursor row position
36 firstcol = "A"  && initialize position for first column
37 lastcol  = "G"  && initialize position for last column
38 firstrow = "A"  && initialize position for first row (ROW A)
39 last_A = "G"  && initialize position for last row in column A
40 last_B = "H"  && initialize position for last row in column B
41 last_C = "H"  && initialize position for last row in column C
42 last_D = "F"  && initialize position for last row in column D
43 last_E = "E"  && initialize position for last row in column E
44 last_F = "J"  && initialize position for last row in column F
45 last_G = "I"  && initialize position for last row in column G
46
47 ** location of columns
48 col_A = 6
49 col_B = 16
50 col_C = 29
51 col_D = 39
52 col_E = 49
53 col_F = 59
54 col_G = 69
55
56 ** location of rows
57 row_A = 9
58 row_B = 10
59 row_C = 11
60 row_D = 12
61 row_E = 13
62 row_F = 14
63 row_G = 15
64 row_H = 16
```

```
 65 row_I = 17
 66 row_J = 18
 67
 68 SET ESCA OFF
 69
 70 ************************
 71 * PRINT SCREEN TITLE *
 72 ************************
 73 @ 5,32 SAY scrtitle
 74
 75 *****************
 76 * PRINT STATES *
 77 *****************
 78 * Region Titles*
 79 *****************
 80 @ 7,3  SAY "Northeast"
 81 @ 8,3  SAY "
 82 ans_AA =    " CT"
 83 ans_AB =    " ME"
 84 ans_AC =    " MA"
 85 ans_AD =    " NH"
 86 ans_AE =    " NY"
 87 ans_AF =    " RI"
 88 ans_AG =    " VT"
 89 @row_A,col_A SAY ans_AA
 90 @row_B,col_A SAY ans_AB
 91 @row_C,col_A SAY ans_AC
 92 @row_D,col_A SAY ans_AD
 93 @row_E,col_A SAY ans_AE
 94 @row_F,col_A SAY ans_AF
 95 @row_G,col_A SAY ans_AG
 96
 97 @ 7,13 SAY "MidAtlantic"
 98 @ 8,13 SAY "
 99 ans_BA = " DE"
100 ans_BB = " DC"
101 ans_BC = " MD"
102 ans_BD = " NJ"
103 ans_BE = " PA"
104 ans_BF = " VA"
105 ans_BG = " VQ"
106 ans_BH = " WV"
107 @row_A,col_B SAY ans_BA
108 @row_B,col_B SAY ans_BB
109 @row_C,col_B SAY ans_BC
110 @row_D,col_B SAY ans_BD
111 @row_E,col_B SAY ans_BE
112 @row_F,col_B SAY ans_BF
113 @row_G,col_B SAY ans_BG
114 @row_H,col_B SAY ans_BH
115
116 @ 7,26 SAY "Southeast"
117 @ 8,26 SAY "
118 ans_CA = " AL"
119 ans_CB = " FL"
120 ans_CC = " GA"
121 ans_CD = " KY"
122 ans_CE = " MS"
123 ans_CF = " NC"
124 ans_CG = " SC"
125 ans_CH = " TN"
126 @row_A,col_C SAY ans_CA
127 @row_B,col_C SAY ans_CB
128 @row_C,col_C SAY ans_CC
129 @row_D,col_C SAY ans_CD
130 @row_E,col_C SAY ans_CE
131 @row_F,col_C SAY ans_CF
```

```
132 @row_G,col_C SAY ans_CG
133 @row_H,col_C SAY ans_CH
134
135 @ 7,36 SAY "Mid-West "
136 @ 8,36 SAY "
137 ans_DA = " IL"
138 ans_DB = " IN"
139 ans_DC = " MI"
140 ans_DD = " MN"
141 ans_DE = " OH"
142 ans_DF = " WI"
143 @row_A,col_D SAY ans_DA
144 @row_B,col_D SAY ans_DB
145 @row_C,col_D SAY ans_DC
146 @row_D,col_D SAY ans_DD
147 @row_E,col_D SAY ans_DE
148 @row_F,col_D SAY ans_DF
149
150 @ 7,46 SAY "Southwest"
151 @ 8,46 SAY "
152 ans_EA = " AR"
153 ans_EB = " LA"
154 ans_EC = " NM"
155 ans_ED = " OK"
156 ans_EE = " TX"
157 @row_A,col_E SAY ans_EA
158 @row_B,col_E SAY ans_EB
159 @row_C,col_E SAY ans_EC
160 @row_D,col_E SAY ans_ED
161 @row_E,col_E SAY ans_EE
162
163 @ 7,56 SAY "Mt/Plain "
164 @ 8,56 SAY "
165 ans_FA = " CO"
166 ans_FB = " IA"
167 ans_FC = " KS"
168 ans_FD = " MO"
169 ans_FE = " MT"
170 ans_FF = " NE"
171 ans_FG = " ND"
172 ans_FH = " SD"
173 ans_FI = " UT"
174 ans_FJ = " WY"
175 @row_A,col_F SAY ans_FA
176 @row_B,col_F SAY ans_FB
177 @row_C,col_F SAY ans_FC
178 @row_D,col_F SAY ans_FD
179 @row_E,col_F SAY ans_FE
180 @row_F,col_F SAY ans_FF
181 @row_G,col_F SAY ans_FG
182 @row_H,col_F SAY ans_FH
183 @row_I,col_F SAY ans_FI
184 @row_J,col_F SAY ans_FJ
185
186 @ 7,66 SAY " Western "
187 @ 8,66 SAY "
188 ans_GA = " AK"
189 ans_GB = " AZ"
190 ans_GC = " CA"
191 ans_GD = " GU"
192 ans_GE = " HI"
193 ans_GF = " ID"
194 ans_GG = " NV"
195 ans_GH = " OR"
196 ans_GI = " WA"
197 @row_A,col_G SAY ans_GA
198 @row_B,col_G SAY ans_GB
```

```
199 @row_C,col_G SAY ans_GC
200 @row_D,col_G SAY ans_GD
201 @row_E,col_G SAY ans_GE
202 @row_F,col_G SAY ans_GF
203 @row_G,col_G SAY ans_GG
204 @row_H,col_G SAY ans_GH
205 @row_I,col_G SAY ans_GI
206
207 @22,1 CLEAR TO 23,78     && clear instruction box
208 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
209 DO FNSCENTR WITH 'Use space bar to toggle on/off choices.',23
210 SET COLOR TO &highlight   && place exit message
211 @24,3 SAY "[Ctrl][W]=Exit"    && in lower left
212 SET COLOR TO &color  && corner of screen
213
214 DO WHILE .T.   && start of menu loop
215    @row_&ctrrow,col_&ctrcol GET ans_&ctrcol&ctrrow
216    CLEAR GETS
217
218    @20,1 CLEAR TO 20,78     && clear message line
219
220    IF numtog = maxtog
221       DO FNSCENTR WITH "Maximum number of states have already been";
221        + " selected",20
222    ENDIF
223
224    *  @22,1 CLEAR TO 23,78     && clear instruction box
225    *  DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
226    *  DO FNSCENTR WITH 'Use space bar to toggle on/off choices.',23
227    *  * SET COLOR TO &highlight   && place exit message
228    *  @24,3 SAY "[Ctrl][W]=Exit"    && in lower left
229    *  * SET COLOR TO &color  && corner of screen
230
231    key = 0           && wait for user key
232    DO WHILE key = 0  && wait for user key
233       key = INKEY()  && wait for user key
234    ENDDO        && wait for user key
235    ************************************
236    ** Evaluate key that was pressed **
237    ************************************
238    DO CASE
239    CASE key = 19     && left arrow
240       @row_&ctrrow,col_&ctrcol SAY ans_&ctrcol&ctrrow  && rewrite;
240         last selection
241       && in standard video
242       ctrcol = LTRIM(CHR(ASC(ctrcol)-1))    && move one column to left
243       ctrcol = IIF((ASC(ctrcol)<ASC(firstcol)),lastcol,ctrcol)
244       && if first col, flip to right side
245        * if previous col shorter than new col, set row to last row of;
245        * new column
246       ctrrow = IIF(ctrrow>last_&ctrcol,last_&ctrcol,ctrrow)
247    CASE key = 4     && right arrow
248       @row_&ctrrow,col_&ctrcol SAY ans_&ctrcol&ctrrow  && rewrite;
248         last selection
249       && in standard video
250       oldcol = ctrcol
251       ctrcol = LTRIM(CHR(ASC(ctrcol)+1))    && move one column to right
252       ctrcol = IIF((ASC(ctrcol)>ASC(lastcol)),firstcol,ctrcol)
253       && if last col, flip to left side
254        * if previous col shorter than new col, set row to last row of;
254        * new column
255       ctrrow = IIF(ctrrow>last_&ctrcol,last_&ctrcol,ctrrow)
256    CASE key = 24      && down arrow
257       @row_&ctrrow,col_&ctrcol SAY ans_&ctrcol&ctrrow  && rewrite;
257         last selection
258       && in standard video
259       ctrrow = LTRIM(CHR(ASC(ctrrow)+1))     && move down one row
```

```
260       ctrrow = IIF((ASC(ctrrow)>ASC(last_&ctrcol)),firstrow,ctrrow)
261    CASE key = 5       && up arrow
262       @row_&ctrrow,col_&ctrcol SAY ans_&ctrcol&ctrrow  && rewrite;
262         last selection
263       && in standard video
264       ctrrow = LTRIM(CHR(ASC(ctrrow)-1))     && move up one row
265       ctrrow = IIF((ASC(ctrrow)<ASC(firstrow)),last_&ctrcol,ctrrow)
266    CASE KEY = 32      && SPACE KEY
267       ans_&ctrcol&ctrrow = IIF((ASC(ans_&ctrcol&ctrrow);
267        =32),STUFF(ans_&ctrcol&ctrrow,1,1,togon),STUFF(ans_&ctrcol&ctrr;
267        ow,1,1,togoff))
268       numtog = IIF(ASC(ans_&ctrcol&ctrrow)<>32,numtog+1,numtog-1)
269       IF numtog > maxtog
270          DO FNSBELL
271          ans_&ctrcol&ctrrow = STUFF(ans_&ctrcol&ctrrow,1,1,togoff)
272          numtog = numtog - 1
273          @22,1 CLEAR TO 23,78     && clear instruction box
274          DO FNSCENTR WITH 'You may not select additional states',22
275          DO FNSCENTR WITH 'Press any key to continue',23
276          tkey = 0           && wait for user key
277          DO WHILE tkey = 0  && wait for user key
278             tkey = INKEY()  && wait for user key
279          ENDDO         && wait for user key
280          @22,1 CLEAR TO 23,78     && clear instruction box
281          DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
282          DO FNSCENTR WITH 'Use space bar to toggle on/off choices.',23
283       ENDIF
284    CASE key = 23      && Ctrl-W key (for exiting from menu)
285       @row_&ctrrow,col_&ctrcol SAY ans_&ctrcol&ctrrow  &&rewrite last;
285         selection
286       &&in standard video
287       @22,1 CLEAR TO 23,78
288       @24,3 SAY REPLICATE(CHR(205),15)
289       DO FNSCENTR WITH "Do you wish to exit and return to previous";
289        + " menu (Y/N)?",22
290       DO FNSCENTR WITH "Type [Y] to exit, [N] to continue selecting";
290        + " states",23
291       STORE SPACE(1) TO answer
292       @23,67 GET answer PICTURE "Y"
293       READ
294       IF answer = "Y"
295          ********************************************************
296          ** Place codes for selected states in output "array" *
297          ********************************************************
298          @22,1 CLEAR TO 23,78
299          DO FNSCENTR WITH "Working...",22
300          IF numtog > 0   && process only if at least one state was;
300            selected
301             tempctr = 1   && temporary output variable counter
302             ctrcol = firstcol
303             DO WHILE ASC(ctrcol) <= ASC(lastcol)
304                ctrrow = firstrow
305                DO WHILE ASC(ctrrow) <= ASC(last_&ctrcol)
306                   IF SUBSTR(ans_&ctrcol&ctrrow,1,1) = togon
307                      tempchr = LTRIM(STR(tempctr))
308                      State_&tempchr = SUBSTR(ans_&ctrcol&ctrrow,2,2)
309                      tempctr = tempctr + 1
310                   ENDIF
311                   ctrrow = LTRIM(CHR(ASC(ctrrow) + 1))
312                ENDDO
313                ctrcol = LTRIM(CHR(ASC(ctrcol) + 1))
314             ENDDO
315             disp_choice = 1
316          ENDIF
317          RETURN
318       ELSE
319          LOOP
```

```
320        ENDIF
321     ENDCASE
322
323 ENDDO         && end of menu loop
324
325 SET ESCAPE ON
```

VARIABLES CROSS-REF FOR FNSMENU3.PRG

Variable	Line number in file
ANSWER:	291 292 294
ANS_:	215 240 248 257 262 267 268 271 285
ANS_AA:	82 89
ANS_AB:	83 90
ANS_AC:	84 91
ANS_AD:	85 92
ANS_AE:	86 93
ANS_AF:	87 94
ANS_AG:	88 95
ANS_BA:	99 107
ANS_BB:	100 108
ANS_BC:	101 109
ANS_BD:	102 110
ANS_BE:	103 111
ANS_BF:	104 112
ANS_BG:	105 113
ANS_BH:	106 114
ANS_CA:	118 126
ANS_CB:	119 127
ANS_CC:	120 128
ANS_CD:	121 129
ANS_CE:	122 130
ANS_CF:	123 131
ANS_CG:	124 132
ANS_CH:	125 133
ANS_DA:	137 143
ANS_DB:	138 144
ANS_DC:	139 145
ANS_DD:	140 146
ANS_DE:	141 147
ANS_DF:	142 148
ANS_EA:	152 157
ANS_EB:	153 158
ANS_EC:	154 159
ANS_ED:	155 160
ANS_EE:	156 161
ANS_FA:	165 175
ANS_FB:	166 176
ANS_FC:	167 177
ANS_FD:	168 178
ANS_FE:	169 179
ANS_FF:	170 180
ANS_FG:	171 181
ANS_FH:	172 182
ANS_FI:	173 183
ANS_FJ:	174 184
ANS_GA:	188 197
ANS_GB:	189 198
ANS_GC:	190 199
ANS_GD:	191 200
ANS_GE:	192 201
ANS_GF:	193 202
ANS_GG:	194 203
ANS_GH:	195 204
ANS_GI:	196 205
COL_:	215 240 248 257 262 285
COL_A:	48 89 90 91 92 93 94 95
COL_B:	49 107 108 109 110 111 112 113 114
COL_C:	50 126 127 128 129 130 131 132 133
COL_D:	51 143 144 145 146 147 148
COL_E:	52 157 158 159 160 161

```
 1 ********************************************
 2 * FNSMENU4.PRG                             *
 3 * written by Edson: 02 June 1988           *
 4 *&Selects FNS Regions for Display
 5 * NOTES: Menu choices must start with "A" *
 6 *        Choices limited to 26  (A-Z)      *
 7 * Public Variables Required:               *
 8 *    highlight                             *
 9 *    color                                 *
10 *    fns_retn                              *
11 * Procedures Required:                     *
12 *    FNSSCRN, FNSCENTER in FNSUTIL         *
13 * Return variables:                        *
14 *    fns_retn: numeric value representing *
15 *              selected choice from menu   *
16 ********************************************
17 SET PROCEDURE TO FNSUTIL
18
19 DO FNSSCRN WITH 'FNSMENU4'
20 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
21 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
22 max_ans = 8
23 scrtitle = "FNS REGION MENU"
24 ans_1 = 'A) Northeast Region'
25 ans_2 = 'B) Mid-Atlantic Region'
26 ans_3 = 'C) Southeast Region'
27 ans_4 = 'D) Mid-West Region'
28 ans_5 = 'E) Southwest Region'
29 ans_6 = 'F) Mountain Plains Region'
30 ans_7 = 'G) Western Region'
31 ans_8 = 'H) RETURN TO PREVIOUS MENU'
32 mess_1 = 'CT, ME, MA, NH, NY, RI, VT'
33 mess_2 = 'DE, DC, MD, NJ, PA, VA, VI, WV'
34 mess_3 = 'AL, FL, GA, KY, MS, NC, SC, TN'
35 mess_4 = 'IL, IN, MI, OH, WI'
36 mess_5 = 'AS, LA, NM, OK, TX'
37 mess_6 = 'CO, IA, KS, MO, MT, NE, ND, SD, UT, WY'
38 mess_7 = 'AK, AR, CA, GU, HI, ID, NV, OR, WA'
39 mess_8 = 'Exit from this Menu'
40
41 fns_retn = 0 && initialize return code
42 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
43 m_line = 20  && Prompt Message ROW()
44 col = 1      && Cur choice COL()
45 cmult = 22   && Column Incrementer
46 ctr = 1      && Row positioner
47 temp = LTRIM(RTRIM((STR(max_ans))))
48 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
49 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
50
51 SET ESCA OFF
52
53 ******************************
54 * PRINT SCREEN TITLE AND BOX *
55 ******************************
56 SET COLOR TO &highlight
57 @ row-2,32 SAY scrtitle
58 SET COLOR TO &color
59 @ row-1,20 TO max_ans+row+1,58 DOUBLE
60
61 ***********************
62 * PRINT ANSWER LINES *
63 ***********************
64 counter = 1
```

```
DO WHILE counter < max_ans + 1
   x = LTRIM(STR(counter,2))
   @ row+counter-1,23 SAY ans_&x
   counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
   @ row+(ctr-1),23 GET ans_&indx         && Read answer line

   DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   CLEAR GETS

   key = 0                                && Wait for user key
   DO WHILE key = 0                       && Wait for user key
      key = INKEY()                        && Wait for user key
   ENDDO                                  && Wait for user key

   ******************************************
   * CONVERT lower case values to upper case *
   ******************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   @ m_line,2 CLEAR TO m_line,78          && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"               && Dn/Up Arrows
      @ row+(ctr-1),23 SAY ans_&indx        && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13                   && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
   ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************
SET ESCAPE ON
RETURN  && FNSMENU4
```

VARIABLES CROSS-REF FOR FNSMENU4.PRG

Variable	Line number in file
ANS_:	67 76 98
ANS_1:	24
ANS_2:	25
ANS_3:	26
ANS_4:	27
ANS_5:	28
ANS_6:	29
ANS_7:	30
ANS_8:	31
CMULT:	45
COL:	44
COUNTER:	64 65 66 67 68
CTR:	46 75 100 101 102 104
DECF:	48 106 108 109
DECL:	49 106 108
FNSCENTR:	20 21 78
FNSSCRN:	19
FNSUTIL:	17
FNS_RETN:	41 104 109
INDX:	75 76 78 98
KEY:	81 83 89
MAX_ANS:	22 42 47 59 65 101 102
MESS_:	78
MESS_1:	32
MESS_2:	33
MESS_3:	34
MESS_4:	35
MESS_5:	36
MESS_6:	37
MESS_7:	38
MESS_8:	39
M_LINE:	43 78 91
ROW:	42 57 59 67 76 98
SCRTITLE:	23 57
TEMP:	47 49
X:	66 67

```
 1 ********************************************
 2 * FNSMENU5.PRG                             *
 3 * written by Edson: 02 June 1988           *
 4 *&Selects Census Region for Display        *
 5 * NOTE: Not used - Edson 2/12/89           *
 6 *                                          *
 7 * NOTES: Menu choices must start with "A" *
 8 *        Choices limited to 26  (A-Z)      *
 9 * Public Variables Required:               *
10 *    highlight                             *
11 *    color                                 *
12 *    fns_retn                              *
13 * Procedures Required:                     *
14 *    FNSSCRN, FNSCENTER in FNSUTIL         *
15 * Return variables:                       *
16 *    fns_retn: numeric value representing *
17 *              selected choice from menu  *
18 ********************************************
19
20 SET PROCEDURE TO FNSUTIL
21 DO FNSSCRN WITH 'FNSMENU5'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 max_ans = 10
25 scrtitle = "CENSUS DIVISION MENU"
26 ans_1 = 'A) NEW ENGLAND'
27 ans_2 = 'B) MIDDLE ATLANTIC'
28 ans_3 = 'C) EAST NORTH CENTRAL'
29 ans_4 = 'D) WEST NORTH CENTRAL'
30 ans_5 = 'E) SOUTH ATLANTIC'
31 ans_6 = 'F) EAST SOUTH CENTRAL'
32 ans_7 = 'G) WEST SOUTH CENTRAL'
33 ans_8 = 'H) MOUNTAIN'
34 ans_9 = 'I) PACIFIC'
35 ans_10= 'J) RETURN TO PREVIOUS MENU'
36 mess_1 = 'MA, RI, CT, ME, NH, VT'
37 mess_2 = 'NY, NJ, PA'
38 mess_3 = 'OH, IN, IL, MI, WI'
39 mess_4 = 'MN, IA, MO, ND, SD, NE, KS'
40 mess_5 = 'DE, MD, DC, VA, WV, NC, SC, GA, FL'
41 mess_6 = 'KY, TN, AL, MS'
42 mess_7 = 'AK, LA, OK, TX'
43 mess_8 = 'MT, ID, WY, CO, NM, AZ, UT, NV'
44 mess_9 = 'WA, OR, CA, AK, HI'
45 mess_10= 'Exit from this Menu'
46
47 fns_retn = 0 && initialize return code
48 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
49 m_line = 20  && Prompt Message ROW()
50 col = 1      && Cur choice COL()
51 cmult = 22   && Column Incrementer
52 ctr = 1      && Row positioner
53 temp = LTRIM(RTRIM((STR(max_ans))))
54 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
55 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
56
57 SET ESCA OFF
58
59 *******************************
60 * PRINT SCREEN TITLE AND BOX *
61 *******************************
62 SET COLOR TO &highlight
63 @ row-2,32 SAY scrtitle
64 SET COLOR TO &color
```

```
@ row-1,20 TO max_ans+row+1,58 DOUBLE

**********************
* PRINT ANSWER LINES *
**********************
counter = 1
DO WHILE counter < max_ans + 1
   x = LTRIM(STR(counter,2))
   @ row+counter-1,23 SAY ans_&x
   counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
   @ row+(ctr-1),23 GET ans_&indx         && Read answer line

   DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   * @ m_line,centr SAY mess_&indx
   * centr = INT((80-LEN(mess_&indx))/2)
   * @ m_line,2 SAY SPACE(75)
   CLEAR GETS

   key = 0                                && Wait for user key
   DO WHILE key = 0                       && Wait for user key
      key = INKEY()                        && Wait for user key
   ENDDO                                  && Wait for user key

   ******************************************
   * CONVERT lower case values to upper case *
   ******************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   @ m_line,30                             && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
      @ row+(ctr-1),23 SAY ans_&indx        && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13                  && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
   ENDCASE

ENDDO
********************
* END OF MENU LOOP *
********************
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSMENU5.PRG

Variable	Line number in file
ANS_:	73 82 107
ANS_1:	26
ANS_10:	35
ANS_2:	27
ANS_3:	28
ANS_4:	29
ANS_5:	30
ANS_6:	31
ANS_7:	32
ANS_8:	33
ANS_9:	34
CMULT:	51
COL:	50
COUNTER:	70 71 72 73 74
CTR:	52 81 109 110 111 113
DECF:	54 115 117 118
DECL:	55 115 117
FNSCENTR:	22 23 84
FNSSCRN:	21
FNSUTIL:	20
FNS_RETN:	47 113 118
INDX:	81 82 84 107
KEY:	90 92 98
MAX_ANS:	24 48 53 65 71 110 111
MESS_:	84
MESS_1:	36
MESS_10:	45
MESS_2:	37
MESS_3:	38
MESS_4:	39
MESS_5:	40
MESS_6:	41
MESS_7:	42
MESS_8:	43
MESS_9:	44
M_LINE:	49 84 100
ROW:	48 63 65 73 82 107
SCRTITLE:	25 63
TEMP:	53 55
X:	72 73

Table of Contents

```
 1 ******************************************
 2 * FNSMENU6.PRG                           *
 3 * written by Edson: 02 June 1988         *
 4 *&Displays Menu for Subject Area Selection
 5 * NOTES: Menu choices must start with "A" *
 6 *        Choices limited to 26  (A-Z)    *
 7 * Public Variables Required:             *
 8 *    highlight                           *
 9 *    color                               *
10 *    fns_retn                            *
11 * Procedures Required:                   *
12 *    FNSSCRN, FNSCENTER in FNSUTIL       *
13 * Return variables:                      *
14 *    fns_retn: numeric value representing *
15 *              selected choice from menu  *
16 ******************************************
17
18 SET PROCEDURE TO FNSUTIL
19
20 DO FNSSCRN WITH 'FNSMENU6'
21 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
22 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
23 DO FNSBELL
24 DO FNSCENTR WITH 'DATA ITEM CHOSEN APPEARS IN MORE THAN 1 SUBJECT',20
25 max_ans = 7
26 scrtitle = "CHOOSE SUBJECT AREA"
27 ans_1 = 'A) AUTO CERTIFICATION'
28 ans_2 = 'B) QUALITY CONTROL'
29 ans_3 = 'C) MONTHLY REPORTING'
30 ans_4 = 'D) CLAIMS (CENSUS)'
31 ans_5 = 'E) COMPUTER MATCHING'
32 ans_6 = 'F) CLAIMS (SURVEY)'
33 ans_7 = 'G) EXIT TO PREVIOUS MENU'
34 mess_1 = ''
35 mess_2 = ''
36 mess_3 = ''
37 mess_4 = ''
38 mess_5 = ''
39 mess_6 = ''
40 mess_7 = ''
41 fns_retn = 0 && initialize return code
42 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
43 m_line = 20  && Prompt Message ROW()
44 col = 1      && Cur choice COL()
45 cmult = 22   && Column Incrementer
46 ctr = 1      && Row positioner
47 temp = LTRIM(RTRIM((STR(max_ans))))
48 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
49 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
50
51 SET ESCA OFF
52
53 ******************************
54 * PRINT SCREEN TITLE AND BOX *
55 ******************************
56 SET COLOR TO &highlight
57 @ row-2,32 SAY scrtitle
58 SET COLOR TO &color
59 @ row-1,20 TO max_ans+row+1,58 DOUBLE
60
61 ***********************
62 * PRINT ANSWER LINES *
63 ***********************
64 counter = 1
```

```
DO WHILE counter < max_ans + 1
   x = LTRIM(STR(counter,2))
   @ row+counter-1,23 SAY ans_&x
   counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
   @ row+(ctr-1),23 GET ans_&indx         && Read answer line

   *  DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   CLEAR GETS

   key = 0                                && Wait for user key
   DO WHILE key = 0                       && Wait for user key
      key = INKEY()                        && Wait for user key
   ENDDO                                  && Wait for user key

   ******************************************
   * CONVERT lower case values to upper case *
   ******************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   *   @ m_line,3 CLEAR TO m_line,77          && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
      @ row+(ctr-1),23 SAY ans_&indx         && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13                  && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
   ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSMENU6.PRG

Variable	Line number in file
ANS_:	67 76 98
ANS_1:	27
ANS_2:	28
ANS_3:	29
ANS_4:	30
ANS_5:	31
ANS_6:	32
ANS_7:	33
CMULT:	45
COL:	44
COUNTER:	64 65 66 67 68
CTR:	46 75 100 101 102 104
DECF:	48 106 108 109
DECL:	49 106 108
FNSBELL:	23
FNSCENTR:	21 22 24
FNSSCRN:	20
FNSUTIL:	18
FNS_RETN:	41 104 109
INDX:	75 76 98
KEY:	81 83 89
MAX_ANS:	25 42 47 59 65 101 102
MESS_1:	34
MESS_2:	35
MESS_3:	36
MESS_4:	37
MESS_5:	38
MESS_6:	39
MESS_7:	40
M_LINE:	43
ROW:	42 57 59 67 76 98
SCRTITLE:	26 57
TEMP:	47 49
X:	66 67

```
 1 ************************************************
 2 * FNSMENU7.PRG                                 *
 3 * written by Edson  9/25/88                    *
 4 * Purpose:  select desired system in states    *
 5 *           with multiple systems              *
 6 ************************************************
 7 * PUBLIC VARS USED BY PROGRAM                  *
 8 *      lookvar = variable name                 *
 9 *      subject = subject area                  *
10 *      code    = state code                    *
11 *      numsys  = number of systems in state    *
12 * PUBLIC VARS PRODUCED BY PROGRAM              *
13 *      rowsys  = row number of desired system  *
14 ************************************************
15 *&Menu to Select Desired System in Multi-System States
16 SET PROCEDURE TO FNSUTIL
17 ***** USE FILE WITH ID_NUM *****
18 STORE "          " TO des_var
19 DO CASE
20 CASE subject = "ACS"
21    USE ACS1
22    des_var = "Q101_A"
23 CASE subject = "COM"
24    USE COM
25    des_var = "SYS_NAME"
26 CASE subject = "SUR"
27    USE SURVEY1
28    des_var = "JURISD_S"
29 ENDCASE
30
31 ***** CREATE VARIABLES TO BE USED BY MENU *****
32 DO FNSSCRN WITH 'FNSMENU7'
33 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
34 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
35 max_ans = numsys
36 ***** CREATE TEXT FOR MENU CHOICES *****
37 temp=1
38 LOCATE FOR STATE = code
39 DO WHILE temp < numsys + 1
40    X = LTRIM(STR(temp,2))
41    ans_&X = LTRIM(STR(ID_NUM)) + " (" + LTRIM(RTRIM(&des_var)) + ")"
42    record_&X = ROWNUM   && store row number for ID_NUM
43    temp = temp + 1
44    CONTINUE
45 ENDDO
46
47
48 fns_retn = 0 && initialize return code
49 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
50 m_line = 20  && Prompt Message ROW()
51 col = 1      && Cur choice COL()
52 cmult = 22   && Column Incrementer
53 ctr = 1      && Row positioner
54 temp = LTRIM(RTRIM((STR(max_ans))))
55 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
56 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
57
58 SET ESCA OFF
59
60 ******************************
61 * PRINT SCREEN TITLE AND BOX *
62 ******************************
63 SET COLOR TO &highlight
64 @ row-4,18 SAY "    THIS STATE CONTAINS MULTIPLE SYSTEMS    "
```

```
@ row-3,18 SAY "MOVE THE CURSOR TO SELECT THE DESIRED SYSTEM"
SET COLOR TO &color
@ row-1,10 TO max_ans+row+1,70 DOUBLE

**********************
* PRINT ANSWER LINES *
**********************
counter = 1
DO WHILE counter < max_ans + 1
   x = LTRIM(STR(counter,2))
   @ row+counter-1,12 SAY ans_&x
   counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
   @ row+(ctr-1),12 GET ans_&indx         && Read answer line

   CLEAR GETS

   key = 0                                && Wait for user key
   DO WHILE key = 0                       && Wait for user key
      key = INKEY()                        && Wait for user key
   ENDDO                                  && Wait for user key

   ******************************************
   * CONVERT lower case values to upper case *
   ******************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)


   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"               && Dn/Up Arrows
      @ row+(ctr-1),12 SAY ans_&indx        && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13                  && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
   ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************

****** DETERMINE DESIRED RECORD NUMBER *******
x = LTRIM(STR(fns_retn))
rowsys = record_&x
@5,1 CLEAR to 20,78
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSMENU7.PRG

Variable	Line number in file
ACS1:	21
ANS_:	41 75 84 104
CMULT:	52
CODE:	38
COL:	51
COM:	24
COUNTER:	72 73 74 75 76
CTR:	53 83 106 107 108 110
DECF:	55 112 114 115
DECL:	56 112 114
DES_VAR:	18 22 25 28 41
FNSCENTR:	33 34
FNSSCRN:	32
FNSUTIL:	16
FNS_RETN:	48 110 115 125
ID_NUM:	41
INDX:	83 84 104
KEY:	88 90 96
MAX_ANS:	35 49 54 67 73 107 108
M_LINE:	50
NUMSYS:	35 39
RECORD_:	42 126
ROW:	49 64 65 67 75 84 104
ROWNUM:	42
ROWSYS:	126
STATE:	38
SUBJECT:	20 23 26
SURVEY1:	27
TEMP:	37 39 40 43 54 56
X:	40 41 42 74 75 125 126

```
 1 ******************************************
 2 * FNSMENU8.PRG                           *
 3 * written by Edson: 25 Sept 1988         *
 4 *&Display Main Menu for Maintenance Procedure
 5 * NOTES: Menu choices must start with "A" *
 6 *        Choices limited to 26  (A-Z)     *
 7 * Public Variables Required:             *
 8 *    highlight                           *
 9 *    color                               *
10 *    fns_retn                            *
11 * Procedures Required:                   *
12 *    FNSSCRN, FNSCENTER in FNSUTIL       *
13 * Return variables:                      *
14 *    fns_retn: numeric value representing *
15 *              selected choice from menu  *
16 ******************************************
17 SET PROCEDURE TO FNSUTIL
18
19 DO FNSSCRN WITH 'FNSMENU1'                                          &&;
19  MODIFY
20 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22    &&;
20  MODIFY
21 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23     &&;
21  MODIFY
22 scrtitle = ' MAINTENANCE MENU '                                    &&;
22  MODIFY
23 max_ans = 4                                                        &&;
23  MODIFY
24 ans_1 = 'A) BACK UP DATA FILES '
25 ans_2 = 'B) RESTORE DATA FILES '
26 ans_3 = 'C) PRINT HISTORY FILE '
27 ans_4 = 'D) RETURN TO PREVIOUS SCREEN '
28 mess_1 = 'Make a copy on floppy diskette '
29 mess_2 = 'Copy files from floppy diskette'
30 mess_3 = 'Print a list of data updates   '
31 mess_4 = 'Exit from this Menu            '
32
33 fns_retn = 0 && initialize return code
34 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
35 m_line = 20  && Prompt Message ROW()
36 col = 1      && Cur choice COL()
37 cmult = 22   && Column Incrementer
38 ctr = 1      && Row positioner
39 temp = LTRIM(RTRIM((STR(max_ans))))
40 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
41 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
42
43 SET ESCA OFF
44
45 ******************************
46 * PRINT SCREEN TITLE AND BOX *
47 ******************************
48 SET COLOR TO &highlight
49 @ row-2,32 SAY scrtitle
50 SET COLOR TO &color
51 @ row-1,20 TO max_ans+row+1,58 DOUBLE
52
53 ***********************
54 * PRINT ANSWER LINES *
55 ***********************
56 counter = 1
57 DO WHILE counter < max_ans + 1
58    x = LTRIM(STR(counter,2))
59    @ row+counter-1,23 SAY ans_&x
```

```
   counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
   @ row+(ctr-1),23 GET ans_&indx         && Read answer line

   DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   * @ m_line,centr SAY mess_&indx
   * centr = INT((80-LEN(mess_&indx))/2)
   * @ m_line,2 SAY SPACE(75)
   CLEAR GETS

   key = 0                                && Wait for user key
   DO WHILE key = 0                       && Wait for user key
      key = INKEY()                        && Wait for user key
   ENDDO                                  && Wait for user key

   ******************************************
   * CONVERT lower case values to upper case *
   ******************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   @ m_line,3 CLEAR TO m_line,77          && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"               && Dn/Up Arrows
      @ row+(ctr-1),23 SAY ans_&indx        && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13                   && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
   ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSMENU8.PRG

Variable	Line number in file						
ANS_:	59	68	93				
ANS_1:	24						
ANS_2:	25						
ANS_3:	26						
ANS_4:	27						
CMULT:	37						
COL:	36						
COUNTER:	56	57	58	59	60		
CTR:	38	67	95	96	97	99	
DECF:	40	101	103	104			
DECL:	41	101	103				
FNSCENTR:	20	21	70				
FNSSCRN:	19						
FNSUTIL:	17						
FNS_RETN:	33	99	104				
INDX:	67	68	70	93			
KEY:	76	78	84				
MAX_ANS:	23	34	39	51	57	96	97
MESS_:	70						
MESS_1:	28						
MESS_2:	29						
MESS_3:	30						
MESS_4:	31						
M_LINE:	35	70	86				
ROW:	34	49	51	59	68	93	
SCRTITLE:	22	49					
TEMP:	39	41					
X:	58	59					

```
 1 ****************************************
 2 *  FNSMENU9.PRG                        *
 3 *  written by Edson 9/25/88            *
 4 *&Prompts for Type of Backup Device
 5 ****************************************
 6
 7 SET PROCEDURE TO FNSUTIL
 8 DO FNSSCRN WITH 'FNSMENU0'                                          &&;
 8  MODIFY
 9 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22    &&;
 9  MODIFY
10 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23     &&;
10  MODIFY
11 scrtitle = "TYPE OF BACKUP DISK"                                    &&;
11  MODIFY
12 max_ans = 5                                    && MODIFY
13 ans_1 = "A) 360K 5¼-inch Floppy"
14 ans_2 = "B) 720K 3½-inch Floppy"
15 ans_3 = "C) 1.2MB High Density Floppy"
16 ans_4 = "D) 1.44MB 3½-inch Floppy"
17 ans_5 = "E) RETURN TO PREVIOUS SCREEN"
18 mess_1 = ''
19 mess_2 = ''
20 mess_3 = ''
21 mess_4 = ''
22 mess_5 = ''
23
24 fns_retn = 0 && initialize return code
25 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
26 m_line = 20  && Prompt Message ROW()
27 col = 1      && Cur choice COL()
28 cmult = 22   && Column Incrementer
29 ctr = 1      && Row positioner
30 temp = LTRIM(RTRIM((STR(max_ans))))
31 DECF = ASC(ans_1)   && Decimal equivalent of first selection index
32 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
33
34 SET ESCA OFF
35
36 ******************************
37 * PRINT SCREEN TITLE AND BOX *
38 ******************************
39 SET COLOR TO &highlight
40 @ row-2,32 SAY scrtitle
41 SET COLOR TO &color
42 @ row-1,20 TO max_ans+row+1,58 DOUBLE
43
44 ***********************
45 * PRINT ANSWER LINES *
46 ***********************
47 counter = 1
48 DO WHILE counter < max_ans + 1
49    x = LTRIM(STR(counter,2))
50    @ row+counter-1,23 SAY ans_&x
51    counter = counter + 1
52 ENDDO
53
54 ***********************
55 * START OF MENU LOOP *
56 ***********************
57 DO WHILE .T.
58    indx=LTRIM(STR(ctr,2))            && Current Position of Cursor
59    @ row+(ctr-1),23 GET ans_&indx    && Read answer line
60
```

```
   DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   * @ m_line,centr SAY mess_&indx
   * centr = INT((80-LEN(mess_&indx))/2)
   * @ m_line,2 SAY SPACE(75)
   CLEAR GETS

   key = 0                        && Wait for user key
   DO WHILE key = 0                    && Wait for user key
      key = INKEY()                 && Wait for user key
   ENDDO                       && Wait for user key

   ******************************************
   * CONVERT lower case values to upper case *
   ******************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   @ m_line,3 CLEAR TO m_line,77     && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
      @ row+(ctr-1),23 SAY ans_&indx           && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13             && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
   ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSMENU9.PRG

Variable	Line number in file						
ANS_:	50	59	84				
ANS_1:	13						
ANS_2:	14						
ANS_3:	15						
ANS_4:	16						
ANS_5:	17						
CMULT:	28						
COL:	27						
COUNTER:	47	48	49	50	51		
CTR:	29	58	86	87	88	90	
DECF:	31	92	94	95			
DECL:	32	92	94				
FNSCENTR:	9	10	61				
FNSSCRN:	8						
FNSUTIL:	7						
FNS_RETN:	24	90	95				
INDX:	58	59	61	84			
KEY:	67	69	75				
MAX_ANS:	12	25	30	42	48	87	88
MESS_:	61						
MESS_1:	18						
MESS_2:	19						
MESS_3:	20						
MESS_4:	21						
MESS_5:	22						
M_LINE:	26	61	77				
ROW:	25	40	42	50	59	84	
SCRTITLE:	11	40					
TEMP:	30	32					
X:	49	50					

```
 1 * FNSPRN1.PRG *
 2 * written on 22 Sept 1988
 3 * by David Edson
 4 *&Prints Data Module Display File
 5
 6 SET TALK OFF
 7
 8 DO WHILE .T.  && screen
 9    SET ESCAPE OFF
10    ******* Display screen ******
11    DO FNSSCRN WITH "FNSPRN1"
12    SET COLOR TO &highlight
13    @9,28 SAY " PLEASE PREPARE PRINTER "
14    SET COLOR TO &color
15    DO FNSCENTR WITH "- Make sure your printer is turned on",11
16    DO FNSCENTR WITH "- Make sure the paper is loaded properly",13
17    DO FNSCENTR WITH "and positioned to the top of form",14
18    DO FNSCENTR WITH "Press [P] to Print,",22
19    DO FNSCENTR WITH "Press [E] to Exit without Printing.",23
20
21    key = 0              && get response from user
22    DO WHILE key=0
23       key = INKEY()
24    ENDDO
25
26    DO CASE
27    CASE (key = 80 .OR. key = 112)  && "P" response
28       @5,1 CLEAR TO 20,78
29       @22,1 CLEAR TO 23,78
30       CALL trexe WITH "ISPRINTER"
31       IF .NOT. tr_retl
32          DO FNSCENTR WITH "PRINTER ERROR !!!",10
33          DO FNSCENTR WITH "Please make sure that your printer is:",12
34          DO FNSCENTR WITH "       - connected to your computer   ",13
35          DO FNSCENTR WITH "       - turned on                    ",14
36          DO FNSCENTR WITH "       - has paper                    ",15
37          DO FNSBELL
38          DO FNSSLEEP
39          LOOP
40       ENDIF
41
42       @5,1 CLEAR TO 20,78
43       DO FNSCENTR WITH "Printing...",10
44       @22,1 CLEAR TO 23,78
45       DO FNSCENTR WITH "Press [Esc] to Cancel Printing.",22
46       USE DISPLAY2
47       SET ESCAPE ON
48       ON ESCAPE EXIT
49       SET CONSOLE OFF
50       SET PRINT ON
51       REPORT FORM FNSPRN1
52       SET PRINT OFF
53       SET CONSOLE ON
54       ON ESCAPE
55       SET ESCAPE OFF
56       USE  &&close database
57       EXIT
58    CASE (key = 69 .OR. key = 101)  && "E" response
59       RETURN    &&return to calling program
60    ENDCASE
61 ENDDO  && screen
62 SET PRINT OFF && make sure print is off in event of "ESC"
63 RETURN  && FNSLOOK
```

VARIABLES CROSS-REF FOR FNSPRN1.PRG

Variable	Line number in file											
DISPLAY2:	46											
ESCAPE:	48	54										
EXIT:	48											
FNSBELL:	37											
FNSCENTR:	15	16	17	18	19	32	33	34	35	36	43	45
FNSPRN1:	51											
FNSSCRN:	11											
FNSSLEEP:	38											
KEY:	21	23										
TREXE:	30											
TR_RETL:	31											

```
 1 * FNSPRN1.PRG *
 2 * written on 22 Sept 1988
 3 * by David Edson
 4 *&Print Data Update History File
 5 SET PROCEDURE TO FNSUTIL
 6 SET TALK OFF
 7
 8 DO WHILE .T.  && screen
 9    SET ESCAPE OFF
10    ******* Display screen ******
11    DO FNSSCRN WITH "FNSPRN1"
12    SET COLOR TO &highlight
13    @9,28 SAY " PLEASE PREPARE PRINTER "
14    SET COLOR TO &color
15    DO FNSCENTR WITH "- Make sure your printer is turned on",11
16    DO FNSCENTR WITH "- Make sure the paper is loaded properly",13
17    DO FNSCENTR WITH "and positioned to the top of form",14
18    DO FNSCENTR WITH "Press [P] to Print,",22
19    DO FNSCENTR WITH "Press [E] to Exit without Printing.",23
20
21    key = 0              && get response from user
22    DO WHILE key=0
23       key = INKEY()
24    ENDDO
25
26    DO CASE
27    CASE (key = 80 .OR. key = 112)  && "P" response
28       @5,1 CLEAR TO 20,78
29       @22,1 CLEAR TO 23,78
30       CALL trexe WITH "ISPRINTER"
31       IF .NOT. tr_ret1
32          DO FNSCENTR WITH "PRINTER ERROR !!!",10
33          DO FNSCENTR WITH "Please make sure that your printer is:",12
34          DO FNSCENTR WITH "       - connected to your computer   ",13
35          DO FNSCENTR WITH "       - turned on                    ",14
36          DO FNSCENTR WITH "       - has paper                    ",15
37          DO FNSBELL
38          DO FNSSLEEP
39          LOOP
40       ENDIF
41
42       @5,1 CLEAR TO 20,78
43       DO FNSCENTR WITH "Printing...",10
44       @22,1 CLEAR TO 23,78
45       DO FNSCENTR WITH "Press [Esc] to Cancel Printing.",22
46       USE FNSLOG
47       SET ESCAPE ON
48       ON ESCAPE EXIT
49       SET CONSOLE OFF
50       SET PRINT ON
51       REPORT FORM FNSPRN2
52       SET PRINT OFF
53       SET CONSOLE ON
54       ON ESCAPE
55       SET ESCAPE OFF
56       USE  &&close database
57       EXIT
58    CASE (key = 69 .OR. key = 101)  && "E" response
59       RETURN    &&return to calling program
60    ENDCASE
61 ENDDO  && screen
62 SET PRINT OFF && make sure print is off in event of "ESC"
63 RETURN  && FNSLOOK
```

VARIABLES CROSS-REF FOR FNSPRN2.PRG

Variable	Line number in file
ESCAPE:	48 54
EXIT:	48
FNSBELL:	37
FNSCENTR:	15 16 17 18 19 32 33 34 35 36 43 45
FNSLOG:	46
FNSPRN2:	51
FNSSCRN:	11
FNSSLEEP:	38
FNSUTIL:	5
KEY:	21 23
TREXE:	30
TR_RETL:	31

```
 1 * FNSPRN3.PRG *
 2 * written on 26 Sept 1988
 3 * by David Edson
 4 *&Prints Standard Report Display File
 5 SET PROCEDURE TO FNSUTIL
 6 SET TALK OFF
 7
 8 DO WHILE .T.  && screen
 9    SET ESCAPE OFF
10    ******* Display screen ******
11    DO FNSSCRN WITH "FNSPRN1"
12    SET COLOR TO &highlight
13    @9,28 SAY " PLEASE PREPARE PRINTER "
14    SET COLOR TO &color
15    DO FNSCENTR WITH "- Make sure your printer is turned on",11
16    DO FNSCENTR WITH "- Make sure the paper is loaded properly",13
17    DO FNSCENTR WITH "and positioned to the top of form",14
18    DO FNSCENTR WITH "Press [P] to Print,",22
19    DO FNSCENTR WITH "Press [E] to Exit without Printing.",23
20
21    key = 0              && get response from user
22    DO WHILE key=0
23       key = INKEY()
24    ENDDO
25
26    DO CASE
27    CASE (key = 80 .OR. key = 112)  && "P" response
28       @5,1 CLEAR TO 20,78
29       @22,1 CLEAR TO 23,78
30       CALL trexe WITH "ISPRINTER"
31       IF .NOT. tr_ret1
32          DO FNSCENTR WITH "PRINTER ERROR !!!",10
33          DO FNSCENTR WITH "Please make sure that your printer is:",12
34          DO FNSCENTR WITH "       - connected to your computer   ",13
35          DO FNSCENTR WITH "       - turned on                    ",14
36          DO FNSCENTR WITH "       - has paper                    ",15
37          DO FNSBELL
38          DO FNSSLEEP
39          LOOP
40       ENDIF
41
42       @5,1 CLEAR TO 20,78
43       DO FNSCENTR WITH "Printing...",10
44       @22,1 CLEAR TO 23,78
45       DO FNSCENTR WITH "Press [Esc] to Cancel Printing.",22
46       USE RESULT2
47       SET ESCAPE ON
48       ON ESCAPE EXIT
49       SET CONSOLE OFF
50       SET PRINT ON
51       REPORT FORM FNSPRN3
52       SET PRINT OFF
53       SET CONSOLE ON
54       ON ESCAPE
55       SET ESCAPE OFF
56       USE  &&close database
57       EXIT
58    CASE (key = 69 .OR. key = 101)  && "E" response
59       CLOSE ALL
60       RETURN    &&return to calling program
61    ENDCASE
62 ENDDO  && screen
63 SET PRINT OFF && make sure print is off in event of "ESC"
64 CLOSE ALL
```

```
65 RETURN  && FNSLOOK
```

VARIABLES CROSS-REF FOR FNSPRN3.PRG

Variable	Line number in file
ESCAPE:	48 54
EXIT:	48
FNSBELL:	37
FNSCENTR:	15 16 17 18 19 32 33 34 35 36 43 45
FNSPRN3:	51
FNSSCRN:	11
FNSSLEEP:	38
FNSUTIL:	5
KEY:	21 23
RESULT2:	46
TREXE:	30
TR_RETL:	31

```
 1 ******************************************
 2 * FNSREQSP.PRG                           *
 3 * written by Edson: 25 Sept 1988         *
 4 * Purpose: Computes space required for   *
 5 *          storing backup files          *
 6 * Uses BACKLIST.DBF for list of .dbf to  *
 7 *  be backed up                          *
 8 * Returns required space in fns_retn     *
 9 ******************************************
10 *&Determines Amount of Storage Space Required for Backup
11 SET ESCAPE OFF
12 SET BELL OFF
13 SET PROCEDURE TO FNSUTIL
14
15 DO FNSSCRN WITH 'FNSREQSP'
16 DO FNSCENTR WITH 'Working...',23
17
18 reqspace = 0
19 USE BACKLIST
20 DO WHILE .NOT. EOF()
21    tempchar = FILE_NAME
22    CALL trpass WITH tempchar
23    CALL trexe WITH "FILESIZE"
24    reqspace = reqspace + tr_retn
25    SKIP
26 ENDDO
27 fns_retn = reqspace
28 USE                    &&closes database
29 RETURN
```

VARIABLES CROSS-REF FOR FNSREQSP.PRG

Variable	Line number in file		
BACKLIST:	19		
FILE_NAME:	21		
FNSCENTR:	16		
FNSSCRN:	15		
FNSUTIL:	13		
FNS_RETN:	27		
REQSPACE:	18	24	27
TEMPCHAR:	21	22	
TREXE:	23		
TRPASS:	22		
TR_RETN:	24		

```
 1 ****************************************
 2 * FNSREST.PRG                          *
 3 * written on 26 Sept 1988              *
 4 * by David Edson                       *
 5 *&Control Program for Restoring Data from Backup Disks
 6 ****************************************
 7
 8 SET PROCEDURE TO FNSUTIL
 9 SET ESCAPE OFF
10 SET TALK OFF
11 *SET CONFIRM ON   &&2/19/89
12 more = .T.
13 DO WHILE .T.
14    **********************
15    fns_retc = ''
16    DO FNSBDISK   && select backup drive
17    IF fns_retc $ 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
18       backdisk = fns_retc
19    ELSE
20       EXIT
21    ENDIF
22    @24,3 SAY REPLICATE(CHR(205),15)  &&remove "Ctrl-W" message
23    @8,1 CLEAR TO 20,78
24    DO WHILE more
25       @22,1 CLEAR TO 23,78
26       DO FNSCENTR WITH 'Retrieving backup copies...',23
27       SET CONSOLE OFF
28       RUN COPY &backdisk:*.* *.* >NUL
29       SET CONSOLE ON
30       ***** ask if additional backup disks
31       @8,1 CLEAR TO 20,78
32       @22,1 CLEAR TO 23,78
33       STORE SPACE(1) TO answer
34       DO FNSCENTR WITH'Are there additional backup diskettes (Y/N)?',22
35       DO FNSCENTR WITH'Type [Y] to Continue, [N] to Conclude Restore';
35        + ' Session',23
36       @23,72 GET answer PICTURE "Y"
37       READ
38       IF answer = "Y"
39          @22,1 CLEAR TO 23,78
40          DO FNSCENTR WITH "Place Next Backup Disk in Drive.",22
41          DO FNSSLEEP
42          LOOP
43       ELSE
44          EXIT
45       ENDIF
46    ENDDO && more loop
47    EXIT
48 ENDDO && main loop                  && ?Follows EXIT never executed
49 SET DEFAULT TO &old_def
50 *SET CONFIRM OFF  &&2/19/89
51 RETURN  && FNSMAINT
```

VARIABLES CROSS-REF FOR FNSREST.PRG

Variable	Line number in file
ADDITIONAL:	34
ANSWER:	33 36 38
ARE:	34
BACKDISK:	18
BACKDISK::	28
BACKUP:	34
CONCLUDE:	35
CONTINUE:	35
COPY:	28
DISKETTES:	34
FNSBDISK:	16
FNSCENTR:	26 34 35 40
FNSSLEEP:	41
FNSUTIL:	8
FNS_RETC:	15 17 18
MORE:	12 24
N:	34
NUL:	28
RESTORE:	35
SESSION:	35
THERE:	34

```
 1 ***** FNSSCROL.PRG ******
 2 *** EDSON 9/19/88
 3 *&Displays dBASE III File Containing Results of Display Module
 4 *************************
 5 set talk off
 6 set escape off
 7 set echo off
 8 ************************
 9 ** Parameters        ***
10 ************************
11 box_len = 14
12 top_box = 6
13 bot_box = top_box + box_len - 1
14 left = 0
15 right = 79
16 *********************************
17 ** Display the starting screen *
18 *********************************
19 DO FNSSCRN WITH 'FNSSCROL'
20 **** SPECIAL DISPLAY SCREEN ****
21 @5,0 CLEAR TO 20,79
22 @4,0 SAY CHR(211)
23 @4,79 SAY CHR(189)
24 @21,0 SAY CHR(214)
25 @21,79 SAY CHR(183)
26 @22,15 SAY " Use [PgUp], [PgDn], ^X, ^Y keys to scroll up and down."
27 @23,31 SAY "Press [E] to exit."
28 @24,5 SAY "[F1] Help"
29 *************************
30 USE DISPLAY2
31 GO TOP
32 top_rec = 1
33 bot_rec = top_rec + box_len - 1
34 IF RECCOUNT() < bot_rec  && cases where number of lines on screen
35    bot_rec = RECCOUNT()  && is greater than number of lines to be
36 ENDIF                   && to be displayed
37 prn_row = top_box
38 DO WHILE RECNO() < bot_rec + 1
39    @prn_row,left SAY LINE
40    prn_row = prn_row + 1
41    SKIP
42 ENDDO
43 ******* END OF STARTING SCREEN *****
44 ************************************
45 DO WHILE .T.
46    key = INKEY()              && Wait for user key
47    DO CASE
48    CASE key= 5  &&CursorUp
49       IF top_rec > 1
50          n1=top_box
51          n2=left
52          n3=bot_box
53          n4=right
54          n5=1
55          c1='d'
56          CALL Trpass WITH n1
57          CALL Trpass WITH n2
58          CALL Trpass WITH n3
59          CALL Trpass WITH n4
60          CALL Trpass WITH n5
61          CALL Trpass WITH c1
62          CALL Trexe WITH "SCROLL"
63          goto top_rec
64          skip -1
```

```
65          @top_box,left SAY LINE  && top blank line
66          top_rec = RECNO()
67          bot_rec = top_rec + box_len - 1
68       ENDIF
69    CASE key = 24  &&CursorDown
70       IF bot_rec < RECCOUNT()
71          n1=top_box
72          n2=left
73          n3=bot_box
74          n4=right
75          n5=1
76          c1='U'
77          CALL Trpass WITH n1
78          CALL Trpass WITH n2
79          CALL Trpass WITH n3
80          CALL Trpass WITH n4
81          CALL Trpass WITH n5
82          CALL Trpass WITH c1
83          CALL Trexe WITH "SCROLL"
84          goto bot_rec
85          skip
86          @bot_box,left SAY LINE && bottom blank line
87          bot_rec = RECNO()
88          top_rec = bot_rec - box_len + 1
89       ENDIF
90    CASE key=3  &&PgDn
91       IF top_rec + box_len < RECCOUNT() + 1
92          @top_box,left CLEAR TO bot_box,right  &&clear display screen
93          top_rec = top_rec + box_len           &&determine new top;
93           /bot rec
94          bot_rec = bot_rec + box_len
95          IF bot_rec > RECCOUNT()        && check for end of file
96             bot_rec = RECCOUNT()
97          ENDIF
98          GOTO top_rec
99          prn_row = top_box
100         DO WHILE RECNO() < bot_rec + 1
101            @prn_row,left SAY LINE
102            prn_row = prn_row + 1
103            SKIP
104         ENDDO
105      ENDIF
106   CASE key=18 &&PgUp
107      IF top_rec > 1
108         @top_box,left CLEAR TO bot_box,right  &&clear display screen
109         top_rec = top_rec - box_len           &&determine new top;
109          /bot rec
110         IF top_rec < 1                 && check for top of file
111            top_rec = 1
112         ENDIF
113         bot_rec = top_rec + box_len - 1
114         GOTO top_rec
115         prn_row = top_box
116         DO WHILE RECNO() < bot_rec + 1
117            @prn_row,left SAY LINE
118            prn_row = prn_row + 1
119            SKIP
120         ENDDO
121      ENDIF
122   CASE STR(key,3) $" 69,101"              &&Exit
123      USE  && close files when done
124      RETURN
125   CASE key = 28   && HELP
126      DO FNSHELP
127   ENDCASE
128 ENDDO
129 SET ESCAPE ON
```

VARIABLES CROSS-REF FOR FNSSCROL.PRG

Variable	Line number in file
BOT_BOX:	13 52 86 92 108
BOT_REC:	33 34 35 38 67 70 84 87 88 94 95 96 100 113 116
BOX_LEN:	11 13 33 67 88 91 93 94 109 113
C1:	55 61 76 82
DISPLAY2:	30
FNSHELP:	126
FNSSCRN:	19
KEY:	46
LEFT:	14 39 51 65 72 86 92 101 108 117
LINE:	39 65 86 101 117
N1:	50 56 71 77
N2:	51 57 72 78
N3:	52 58 73 79
N4:	53 59 74 80
N5:	54 60 75 81
PRN_ROW:	37 39 40 99 101 102 115 117 118
RIGHT:	15 53 74 92 108
TOP_BOX:	12 13 37 50 65 92 99 108 115
TOP_REC:	32 33 49 63 66 67 88 91 93 98 107 109 110 111 113 114
TREXE:	62 83
TRPASS:	56 57 58 59 60 61 77 78 79 80 81 82

```
 1 * FNSSTD.PRG *
 2 * written on 26 Sept 1988
 3 * by David Edson
 4 *&Control Program for Standard Report Module
 5
 6 m_line = 20
 7 rptnum = 99
 8 DO WHILE .T.
 9
10    DO FNSSTD0  &&select subject area
11
12    DO CASE   &&select module
13    CASE fns_retn = 1
14       subject = 'ACS'
15       DO FNSSTDAC
16       IF fns_retn = 12
17          LOOP
18       ENDIF
19    CASE fns_retn = 2
20       subject = 'QC'
21       DO FNSSTDQC
22       IF fns_retn = 6
23          LOOP
24       ENDIF
25    CASE fns_retn = 3
26       subject = 'MR'
27       DO FNSSTDMR
28       IF fns_retn = 6
29          LOOP
30       ENDIF
31    CASE fns_retn = 4
32       subject = 'CLA'
33       DO FNSSTDCL
34       IF fns_retn = 9
35          LOOP
36       ENDIF
37    CASE fns_retn = 5
38       subject = 'COM'
39       DO FNSSTDCO
40       IF fns_retn = 7
41          LOOP
42       ENDIF
43    CASE fns_retn = 6
44       subject = 'SUR'
45       DO FNSSTDSC
46       IF fns_retn = 9
47          LOOP
48       ENDIF
49    CASE fns_retn = 7
50       EXIT
51    ENDCASE
52    rptnum = fns_retn
53    ***********************************
54    DO FNSCAT
55    DO FILTER
56    ***********************************
57    @5,1 CLEAR TO 20,78
58    @22,1 CLEAR TO 23,78
59    DO FNSCENTR WITH 'Working...',22
60
61    DO FNSSTD1      &&create display file
62    DO FNSPRN3      &&print display file
63
64 ENDDO
```

```
65
66 RETURN
```

VARIABLES CROSS-REF FOR FNSSTD.PRG

Variable	Line number in file												
FNSCAT:	54												
FNSCENTR:	59												
FNSPRN3:	62												
FNSSTD0:	10												
FNSSTD1:	61												
FNSSTDAC:	15												
FNSSTDCL:	33												
FNSSTDCO:	39												
FNSSTDMR:	27												
FNSSTDQC:	21												
FNSSTDSC:	45												
FNS_RETN:	13	16	19	22	25	28	31	34	37	40	43	46	49
	52												
M_LINE:	6												
RPTNUM:	7	52											
SUBJECT:	14	20	26	32	38	44							

```
 1 ************************************************
 2 * FNSSTD0.PRG                                  *
 3 * written by Edson: 26 Sept 1988               *
 4 * Purpose: SELECT SUBJECT AREA FOR STD REPORT *
 5 *&Menu for Selecting Standard Report Subject Area
 6 * NOTES: Menu choices must start with "A"      *
 7 *        Choices limited to 26  (A-Z)          *
 8 * Public Variables Required:                   *
 9 *    highlight                                 *
10 *    color                                     *
11 *    fns_retn                                  *
12 * Procedures Required:                         *
13 *    FNSSCRN, FNSCENTER in FNSUTIL             *
14 * Return variables:                           *
15 *    fns_retn: numeric value representing      *
16 *              selected choice from menu       *
17 ************************************************
18
19 SET PROCEDURE TO FNSUTIL
20
21 DO FNSSCRN WITH 'FNSSTD0'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 max_ans = 7
25 scrtitle = "CHOOOSE SUBJECT AREA FOR STANDARD REPORT"
26 ans_1 = 'A) AUTO CERTIFICATION '
27 ans_2 = 'B) QUALITY CONTROL '
28 ans_3 = 'C) MONTHLY REPORTING '
29 ans_4 = 'D) CLAIMS (CENSUS) '
30 ans_5 = 'E) COMPUTER MATCHING '
31 ans_6 = 'F) CLAIMS (SURVEY) '
32 ans_7 = 'G) EXIT TO PREVIOUS MENU '
33 mess_1 = ''
34 mess_2 = ''
35 mess_3 = ''
36 mess_4 = ''
37 mess_5 = ''
38 mess_6 = ''
39 mess_7 = ''
40
41 fns_retn = 0 && initialize return code
42 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
43 m_line = 20  && Prompt Message ROW()
44 col = 1      && Cur choice COL()
45 cmult = 22   && Column Incrementer
46 ctr = 1      && Row positioner
47 temp = LTRIM(RTRIM((STR(max_ans))))
48 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
49 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
50
51 SET ESCA OFF
52
53 ******************************
54 * PRINT SCREEN TITLE AND BOX *
55 ******************************
56 * SET COLOR TO &highlight
57 * @ row-2,32 SAY scrtitle
58 DO FNSCENTR WITH '&scrtitle',row-2
59 * SET COLOR TO &color
60 @ row-1,20 TO max_ans+row+1,58 DOUBLE
61
62 ************************
63 * PRINT ANSWER LINES *
64 ************************
```

```
counter = 1
DO WHILE counter < max_ans + 1
   x = LTRIM(STR(counter,2))
   @ row+counter-1,23 SAY ans_&x
   counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
   @ row+(ctr-1),23 GET ans_&indx         && Read answer line

   DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   CLEAR GETS

   key = 0                                && Wait for user key
   DO WHILE key = 0                       && Wait for user key
      key = INKEY()                        && Wait for user key
   ENDDO                                  && Wait for user key

   ****************************************
   * CONVERT lower case values to upper case *
   ****************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   *   @ m_line,3 CLEAR TO m_line,77          && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
      @ row+(ctr-1),23 SAY ans_&indx         && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13                  && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
    ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSSTD0.PRG

Variable	Line number in file						
ANS_:	68	77	99				
ANS_1:	26						
ANS_2:	27						
ANS_3:	28						
ANS_4:	29						
ANS_5:	30						
ANS_6:	31						
ANS_7:	32						
CMULT:	45						
COL:	44						
COUNTER:	65	66	67	68	69		
CTR:	46	76	101	102	103	105	
DECF:	48	107	109	110			
DECL:	49	107	109				
FNSCENTR:	22	23	58	79			
FNSSCRN:	21						
FNSUTIL:	19						
FNS_RETN:	41	105	110				
INDX:	76	77	79	99			
KEY:	82	84	90				
MAX_ANS:	24	42	47	60	66	102	103
MESS_:	79						
MESS_1:	33						
MESS_2:	34						
MESS_3:	35						
MESS_4:	36						
MESS_5:	37						
MESS_6:	38						
MESS_7:	39						
M_LINE:	43	79					
ROW:	42	58	60	68	77	99	
SCRTITLE:	25						
TEMP:	47	49					
X:	67	68					

```
 1 ************************************************
 2 ** FNSSTD1.PRG - EDSON, 9/26/88
 3 **  CREATES DISPLAY FILE FOR R&R STANDARD REPORTS
 4 **  PROBLEMS:  Q101 (ACS1) AND Q1403 (ACS2) CHARACTER VARIABLES
 5 **               CAUSE PROGRAM TO CRASH WITH DATA TYPE MISMATCH
 6 **             SHOULD INSERT ROUTINE TO CREATE AN EXTRACT FIRST
 7 **             NEEDS TO BE INTEGRATED INTO THE REST OF THE SYSTEM
 8 ************************************************
 9 **  "A" - EXTENDED STRUCTURE FILE
10 **  "B" - DATA FILE
11 **  "C" - RESULT FILE (REQUIRES SHELLRE2.DBF)
12 **  "D" - DATA DICTIONARY (DATADICT.DBF WITH INDEX NAME.NDX)
13 **  "E" - REPORT DEFINITION FILE (REPDEF.DBF)
14 **  "F" - FNS STATE FILE (FNSSTATE.DBF)
15 ************************************************
16 *&Creates Display File for R&R Report Writer
17 SET ESCAPE OFF
18 SET PROCEDURE TO FNSUTIL
19 DO FNSBELL
20 DO FNSCENTR WITH 'System is determining which variables',10
21 DO FNSCENTR WITH 'should be included in the report...',11
22 **** Prepare Structure Extended File ***
23 SELECT A
24 USE STRUCT2
25 COPY TO STDSTRUC.DBF
26 USE STDSTRUC
27 ***** Prepare Report Definition File ***
28 SELECT B
29 USE RPTDEF
30 LOCATE FOR (REPORTNUM=rptnum .AND. SUBJECT = '&subject')
31 varfile = FILENAME
32 ***** MAIN LOOP, PART 1 *****
33 DO WHILE .NOT. EOF()
34    IF FOUND()
35       SELECT A
36       APPEND BLANK
37       REPLACE FIELD_NAME WITH B->FIELD_NAMEFIELD_TYPE WITH B;
37        ->FIELD_TYPEFIELD_LEN  WITH B->FIELD_LENFIELD_DEC  WITH B;
37        ->FIELD_DEC
38       SELECT B
39       CONTINUE
40    ENDIF
41 ENDDO &&main loop, part 1
42 CLOSE ALL
43 SELECT A
44 USE STDSTRUC
45 GO TOP
46 ********************************************
47 SELECT B
48 USE &varfile
49 GO TOP
50 ****************************
51 SELECT C
52 USE SHELLRE2.DBF
53 COPY TO RESULT2.DBF
54 USE RESULT2.DBF
55 GO TOP
56 ****************************
57 SELECT D
58 USE DATADICT INDEX NAME
59 ****************************
60 SET PROCEDURE TO FNSUTIL
61 DO FNSBELL
62 @10,1 CLEAR TO 10,78
```

```
 63 DO FNSCENTR WITH 'Creating Standard Report...',10
 64 DO FNSCENTR WITH 'Processing State;',11
 65 DO FNSCENTR WITH 'Press [Esc] to Cancel Processing.',23
 66 ******************
 67 SET ESCAPE ON
 68 ON ESCAPE RETURN
 69 SELECT F
 70 USE FNSSTATE
 71 max_states = 10
 72 store "  " to null_state
 73 num = 1
 74 DO WHILE num <= max_states
 75    numchar = LTRIM(str(num,2))
 76    IF state_&numchar <> null_state
 77       SELECT F
 78       LOCATE FOR ABBREV = state_&numchar
 79       SET COLOR TO &highlight
 80       @12,39 SAY state_&numchar
 81       SET COLOR TO &color
 82       code = STATECODE
 83       SELECT B
 84       GO TOP
 85       LOCATE FOR STATE = code
 86       DO WHILE .NOT. EOF()
 87          SELECT A
 88          GO TOP
 89          ************
 90          DO WHILE .NOT. EOF() && "A" FILE (EXTENDED STRUCTURE)
 91             SELECT C
 92             APPEND BLANK
 93             INDENT = SPACE(5)
 94             TEMPNAME = TRIM(A->FIELD_NAME)
 95             IF A->FIELD_TYPE = 'N'
 96                TEMPVAR=INDENT + A->FIELD_NAME -' ='+ LTRIM(STR(B;
 96                 ->&TEMPNAME,15,2))
 97             ELSE
 98                TEMPVAR=INDENT + A->FIELD_NAME - ' ='+ LTRIM(B->&TEMPNAME)
 99             ENDIF
100             REPLACE LINE WITH TEMPVAR
101             SELECT D
102             FIND &TEMPNAME
103             SELECT C
104             APPEND BLANK                      &&NEW
105             IF A->FIELD_NAME = 'STATE'
106                TEMPVAR = INDENT + INDENT + "(" + RTRIM(LTRIM(F;
106                 ->NAME))+ ")"
107             ELSE
108                TEMPVAR = INDENT + INDENT + "(" + RTRIM(LTRIM(D;
108                 ->DESCRIP))+ ")"
109             ENDIF
110             REPLACE LINE WITH TEMPVAR       &&NEW
111             APPEND BLANK && append blank line between variables
112             SELECT A
113             SKIP
114          ENDDO                  && "A" FILE (EXTENDED STRUCTURE)
115          ************
116          SELECT C               && ADD SPACER BETWEEN STATES
117          APPEND BLANK
118          TEMPVAR = REPLICATE('-',10)
119          REPLACE LINE WITH TEMPVAR
120          SELECT B
121          CONTINUE  && GET NEXT RECORD FOR CURRENT STATE, IF ANY
122       ENDDO
123       num = num + 1
124    ELSE
125       EXIT
126    ENDIF
```

```
ENDDO                    && "B" FILE (DATA FILE)
ON ESCAPE
CLOSE ALL
**** INSERT MESSAGE IF STATE NOT INCLUDED IN SURVEY/CENSUS
SELECT A
USE RESULT2
COUNT TO num_lines
IF num_lines = 0
   APPEND BLANK
   TEMPVAR = "No information for this state collected."
   REPLACE LINE WITH TEMPVAR
ENDIF
close all
SET ESCAPE ON
```

VARIABLES CROSS-REF FOR FNSSTD1.PRG

Variable	Line number in file
A:	23 43 87 95 96 98 105 112 131
ABBREV:	78
B:	28 37 38 47 83 96 120
C:	51 91 103 116
CODE:	82 85
D:	57 101 108
DATADICT:	58
DESCRIP:	108
ESCAPE:	68 128
F:	69 77 106
FIELD_DEC:	37
FIELD_LENF:	37
FIELD_NAME:	37 94 96 98 105
FIELD_TYPE:	37 95
FILENAME:	31
FNSBELL:	19 61
FNSCENTR:	20 21 63 64 65
FNSSTATE:	70
FNSUTIL:	18 60
INDENT:	93 96 98 106 108
LINE:	100 110 119 137
MAX_STATES:	71 74
NAME:	58 106
NULL_STATE:	72 76
NUM:	73 74 75 123
NUMCHAR:	75 76 78 80
NUM_LINES:	133 134
RESULT2:	132
RESULT2.DB:	53 54
RETURN:	68
RPTDEF:	29
RPTNUM:	30
SHELLRE2.D:	52
STATE:	85
STATECODE:	82
STATE_:	76 78 80
STDSTRUC:	26 44
STDSTRUC.D:	25
STRUCT2:	24
SUBJECT:	30
TEMPNAME:	94 96 98 102
TEMPVAR:	96 98 100 106 108 110 118 119 136 137
VARFILE:	31 48

```
 1 ***********************************************
 2 * FNSSTDAC.PRG                                *
 3 * written by Edson: 26 Sept 1988              *
 4 * Purpose: SELECT ACS MODULE FOR STD REPORT   *
 5 *&Menu to Select ACS Module for Standard Report
 6 * NOTES: Menu choices must start with "A"     *
 7 *        Choices limited to 26  (A-Z)         *
 8 * Public Variables Required:                  *
 9 *    highlight                                *
10 *    color                                    *
11 *    fns_retn                                 *
12 * Procedures Required:                        *
13 *    FNSSCRN, FNSCENTER in FNSUTIL            *
14 * Return variables:                          *
15 *    fns_retn: numeric value representing     *
16 *              selected choice from menu      *
17 ***********************************************
18
19 SET PROCEDURE TO FNSUTIL
20
21 DO FNSSCRN WITH 'FNSSTDAC'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 * DO FNSBELL
25 max_ans = 12
26 scrtitle = "CHOOSE MODULE FOR ACS STANDARD REPORT"
27 ans_1  = 'A) SYSTEM ID/DATA COLLECTION SUPPORT FUNCTIONS '
28 ans_2  = 'B) DATA BASE CONTENT '
29 ans_3  = 'C) AUTOMATION ELIG./BENEFIT CALCULATIONS '
30 ans_4  = 'D) FUNCTIONAL INTEGRATION '
31 ans_5  = 'E) SCOPE OF AUTOMATED DATA BASE '
32 ans_6  = 'F) WORKER ACCESS/CONVENIENCE '
33 ans_7  = 'G) TERMINAL ACCESS/FUNCTIONS '
34 ans_8  = 'H) FOOD STAMP/AFDC INTERACTION '
35 ans_9  = 'I) AGENCY STAFF, COSTS, AND CASE VOLUME '
36 ans_10 = 'J) HARDWARE/SOFTWARE '
37 ans_11 = 'K) CONSTRUCTED VARIABLES '
38 ans_12 = 'L) RETURN TO PREVIOUS SCREEN'
39 mess_1  = '      Modules 1 and 2       '
40 mess_2  = '          Module 3          '
41 mess_3  = '      Modules 4 and 5       '
42 mess_4  = '          Module 6          '
43 mess_5  = '          Module 7          '
44 mess_6  = '      Modules 8 and 9       '
45 mess_7  = '      Modules 10 and 11     '
46 mess_8  = '          Module 12         '
47 mess_9  = '          Module 13         '
48 mess_10 = '          Module 14         '
49 mess_11 = 'Derived from Other Variables'
50 mess_12 = '                            '
51 fns_retn = 0 && initialize return code
52 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
53 m_line = 20  && Prompt Message ROW()
54 col = 1      && Cur choice COL()
55 cmult = 22   && Column Incrementer
56 ctr = 1      && Row positioner
57 temp = LTRIM(RTRIM((STR(max_ans))))
58 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
59 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
60
61 SET ESCA OFF
62
63 ******************************
64 * PRINT SCREEN TITLE AND BOX *
```

```
 65 ******************************
 66 *SET COLOR TO &highlight
 67 *@ row-2,32 SAY scrtitle
 68 *SET COLOR TO &color
 69 DO FNSCENTR WITH '&scrtitle',row-2
 70 @ row-1,14 TO max_ans+row,64 DOUBLE
 71
 72 ***********************
 73 * PRINT ANSWER LINES *
 74 ***********************
 75 counter = 1
 76 DO WHILE counter < max_ans + 1
 77    x = LTRIM(STR(counter,2))
 78    @ row+counter-1,17 SAY ans_&x
 79    counter = counter + 1
 80 ENDDO
 81
 82 ***********************
 83 * START OF MENU LOOP *
 84 ***********************
 85 DO WHILE .T.
 86    indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
 87    @ row+(ctr-1),17 GET ans_&indx         && Read answer line
 88
 89    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
 90    CLEAR GETS
 91
 92    key = 0                                && Wait for user key
 93    DO WHILE key = 0                       && Wait for user key
 94       key = INKEY()                        && Wait for user key
 95    ENDDO                                  && Wait for user key
 96
 97    ****************************************
 98    * CONVERT lower case values to upper case *
 99    ****************************************
100    key = IIF((key > 96 .AND. key < 123),(key-32),key)
101
102    *    @ m_line,3 CLEAR TO m_line,77          && Clear message line
103
104    *********************************
105    * Evaluate key that was pressed *
106    *********************************
107    DO CASE
108    CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
109       @ row+(ctr-1),17 SAY ans_&indx         && rewrites last selection
110       && in standard video
111       ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
112       ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
113       ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
114    CASE key = 13                  && <Enter>
115       fns_retn = ctr
116       EXIT
117    CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
118        * ignore
119    CASE key >=DECF .AND. key <= DECL    &&CHOICES
120       fns_retn = key - DECF + 1
121       EXIT
122    ENDCASE
123
124 ENDDO
125 *********************
126 * END OF MENU LOOP *
127 *********************
128 SET ESCAPE ON
129 RETURN
```

VARIABLES CROSS-REF FOR FNSSTDAC.PRG

Variable	Line number in file
ANS_:	78 87 109
ANS_1:	27
ANS_10:	36
ANS_11:	37
ANS_12:	38
ANS_2:	28
ANS_3:	29
ANS_4:	30
ANS_5:	31
ANS_6:	32
ANS_7:	33
ANS_8:	34
ANS_9:	35
CMULT:	55
COL:	54
COUNTER:	75 76 77 78 79
CTR:	56 86 111 112 113 115
DECF:	58 117 119 120
DECL:	59 117 119
FNSCENTR:	22 23 69 89
FNSSCRN:	21
FNSUTIL:	19
FNS_RETN:	51 115 120
INDX:	86 87 89 109
KEY:	92 94 100
MAX_ANS:	25 52 57 70 76 112 113
MESS_:	89
MESS_1:	39
MESS_10:	48
MESS_11:	49
MESS_12:	50
MESS_2:	40
MESS_3:	41
MESS_4:	42
MESS_5:	43
MESS_6:	44
MESS_7:	45
MESS_8:	46
MESS_9:	47
M_LINE:	53 89
ROW:	52 69 70 78 87 109
SCRTITLE:	26
TEMP:	57 59
X:	77 78

```
1  ************************************************
2  * FNSSTDCL.PRG                                 *
3  * written by Edson: 26 Sept 1988               *
4  * Purpose: SELECT SUBJECT AREA FOR STD REPORT *
5  *&Menu to Select Claims (Census) Module for Standard Report
6  * NOTES: Menu choices must start with "A"      *
7  *        Choices limited to 26  (A-Z)          *
8  * Public Variables Required:                   *
9  *    highlight                                 *
10 *    color                                     *
11 *    fns_retn                                  *
12 * Procedures Required:                         *
13 *    FNSSCRN, FNSCENTER in FNSUTIL             *
14 * Return variables:                           *
15 *    fns_retn: numeric value representing      *
16 *              selected choice from menu       *
17 ************************************************
18
19 SET PROCEDURE TO FNSUTIL
20
21 DO FNSSCRN WITH 'FNSSTDCL'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 * DO FNSBELL
25 max_ans = 9
26 scrtitle = "CHOOSE MODULE FOR CLAIMS (CENSUS) STANDARD REPORT"
27 ans_1  = 'A) AGENCY ORGANIZATION '
28 ans_2  = 'B) ADMINISTRATION CONTROL '
29 ans_3  = 'C) CLAIMS MONITORING '
30 ans_4  = 'D) CLAIM ESTABLISHMENT '
31 ans_5  = 'E) COLLECTION OF PAYMENTS '
32 ans_6  = 'F) FOLLOW-UP FOR DELINQUENT CLAIMS '
33 ans_7  = 'G) SUSPENSION/TERMINATION OF CLAIMS '
34 ans_8  = 'H) CONSTRUCTED VARIABLES '
35 ans_9  = 'I) RETURN TO PREVIOUS SCREEN '
36 mess_1  = '           Module 1          '
37 mess_2  = '           Module 2          '
38 mess_3  = '           Module 3          '
39 mess_4  = '           Module 6          '
40 mess_5  = '           Module 7          '
41 mess_6  = '           Module 8          '
42 mess_7  = '           Module 9          '
43 mess_8  = 'Derived from Other Variables'
44 mess_9  = '                             '
45 fns_retn = 0 && initialize return code
46 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
47 m_line = 20  && Prompt Message ROW()
48 col = 1      && Cur choice COL()
49 cmult = 22   && Column Incrementer
50 ctr = 1      && Row positioner
51 temp = LTRIM(RTRIM((STR(max_ans))))
52 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
53 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
54
55 SET ESCA OFF
56
57 *******************************
58 * PRINT SCREEN TITLE AND BOX *
59 *******************************
60 *SET COLOR TO &highlight
61 *@ row-2,32 SAY scrtitle
62 *SET COLOR TO &color
63 DO FNSCENTR WITH '&scrtitle',row-2
64 @ row-1,19 TO max_ans+row,58 DOUBLE
```

```
 65
 66 ***********************
 67 * PRINT ANSWER LINES *
 68 ***********************
 69 counter = 1
 70 DO WHILE counter < max_ans + 1
 71    x = LTRIM(STR(counter,2))
 72    @ row+counter-1,22 SAY ans_&x
 73    counter = counter + 1
 74 ENDDO
 75
 76 ***********************
 77 * START OF MENU LOOP *
 78 ***********************
 79 DO WHILE .T.
 80    indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
 81    @ row+(ctr-1),22 GET ans_&indx         && Read answer line
 82
 83    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
 84    CLEAR GETS
 85
 86    key = 0                                && Wait for user key
 87    DO WHILE key = 0                       && Wait for user key
 88       key = INKEY()                        && Wait for user key
 89    ENDDO                                  && Wait for user key
 90
 91    ****************************************
 92    * CONVERT lower case values to upper case *
 93    ****************************************
 94    key = IIF((key > 96 .AND. key < 123),(key-32),key)
 95
 96    *   @ m_line,3 CLEAR TO m_line,77          && Clear message line
 97
 98    *********************************
 99    * Evaluate key that was pressed *
100    *********************************
101    DO CASE
102    CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
103       @ row+(ctr-1),22 SAY ans_&indx         && rewrites last selection
104       && in standard video
105       ctr=ctr+IIF(STR(key,2)$"24",1,-1)      && if Dn, increment ctr
106       ctr=IIF(ctr>max_ans,1,ctr)    && flip to top if down and last line
107       ctr=IIF(ctr<1,max_ans,ctr)    && flip to bottom if up and top line
108    CASE key = 13                    && <Enter>
109       fns_retn = ctr
110       EXIT
111    CASE key <DECF .OR. key > DECL        &&JUNK (ignore remaining keys)
112        * ignore
113    CASE key >=DECF .AND. key <= DECL     &&CHOICES
114       fns_retn = key - DECF + 1
115       EXIT
116    ENDCASE
117
118 ENDDO
119 *********************
120 * END OF MENU LOOP *
121 *********************
122 SET ESCAPE ON
123 RETURN
```

VARIABLES CROSS-REF FOR FNSSTDCL.PRG

Variable	Line number in file
ANS_:	72 81 103
ANS_1:	27
ANS_2:	28
ANS_3:	29
ANS_4:	30
ANS_5:	31
ANS_6:	32
ANS_7:	33
ANS_8:	34
ANS_9:	35
CMULT:	49
COL:	48
COUNTER:	69 70 71 72 73
CTR:	50 80 105 106 107 109
DECF:	52 111 113 114
DECL:	53 111 113
FNSCENTR:	22 23 63 83
FNSSCRN:	21
FNSUTIL:	19
FNS_RETN:	45 109 114
INDX:	80 81 83 103
KEY:	86 88 94
MAX_ANS:	25 46 51 64 70 106 107
MESS_:	83
MESS_1:	36
MESS_2:	37
MESS_3:	38
MESS_4:	39
MESS_5:	40
MESS_6:	41
MESS_7:	42
MESS_8:	43
MESS_9:	44
M_LINE:	47 83
ROW:	46 63 64 72 81 103
SCRTITLE:	26
TEMP:	51 53
X:	71 72

```
1  ************************************************
2  * FNSSTDCO.PRG                                 *
3  * written by Edson: 26 Sept 1988               *
4  * Purpose: SELECT SUBJECT AREA FOR STD REPORT *
5  *&Menu to Select Computer Matching Module for Standard Report
6  * NOTES: Menu choices must start with "A"      *
7  *        Choices limited to 26  (A-Z)          *
8  * Public Variables Required:                   *
9  *    highlight                                 *
10 *    color                                     *
11 *    fns_retn                                  *
12 * Procedures Required:                         *
13 *    FNSSCRN, FNSCENTER in FNSUTIL             *
14 * Return variables:                           *
15 *    fns_retn: numeric value representing      *
16 *              selected choice from menu       *
17 ************************************************
18
19 SET PROCEDURE TO FNSUTIL
20
21 DO FNSSCRN WITH 'FNSSTDCO'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 * DO FNSBELL
25 max_ans = 7
26 scrtitle = "CHOOSE MODULE FOR COMPUTER MATCHING STANDARD REPORT"
27 ans_1  = 'A) SYSTEM IDENTIFICATION/INFORMATION '
28 ans_2  = 'B) DATA BASES '
29 ans_3  = 'C) SYSTEM ACCESS AND CASE IDENTIFIER '
30 ans_4  = 'D) FRONT-END MATCHING '
31 ans_5  = 'E) ON-GOING MATCHING '
32 ans_6  = 'F) CONSTRUCTED VARIABLES '
33 ans_7  = 'G) RETURN TO PREVIOUS SCREEN '
34 mess_1  = '       Modules 1 and 2         '
35 mess_2  = '           Module 4            '
36 mess_3  = '           Module 5            '
37 mess_4  = '           Module 6            '
38 mess_5  = '           Module 8            '
39 mess_6  = '  Derived for Other Variables  '
40 mess_7  = '                               '
41 fns_retn = 0 && initialize return code
42 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
43 m_line = 20  && Prompt Message ROW()
44 col = 1      && Cur choice COL()
45 cmult = 22   && Column Incrementer
46 ctr = 1      && Row positioner
47 temp = LTRIM(RTRIM((STR(max_ans))))
48 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
49 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
50
51 SET ESCA OFF
52
53 *******************************
54 * PRINT SCREEN TITLE AND BOX *
55 *******************************
56 *SET COLOR TO &highlight
57 *@ row-2,32 SAY scrtitle
58 *SET COLOR TO &color
59 DO FNSCENTR WITH '&scrtitle',row-2
60 @ row-1,18 TO max_ans+row,59 DOUBLE
61
62 ************************
63 * PRINT ANSWER LINES *
64 ************************
```

```
counter = 1
DO WHILE counter < max_ans + 1
   x = LTRIM(STR(counter,2))
   @ row+counter-1,21 SAY ans_&x
   counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
   indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
   @ row+(ctr-1),21 GET ans_&indx         && Read answer line

   DO FNSCENTR WITH mess_&indx,m_line     && Print message line
   CLEAR GETS

   key = 0                                && Wait for user key
   DO WHILE key = 0                       && Wait for user key
      key = INKEY()                        && Wait for user key
   ENDDO                                  && Wait for user key

   *****************************************
   * CONVERT lower case values to upper case *
   *****************************************
   key = IIF((key > 96 .AND. key < 123),(key-32),key)

   *   @ m_line,3 CLEAR TO m_line,77          && Clear message line

   *********************************
   * Evaluate key that was pressed *
   *********************************
   DO CASE
   CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
      @ row+(ctr-1),21 SAY ans_&indx         && rewrites last selection
      && in standard video
      ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
      ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
      ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
   CASE key = 13                  && <Enter>
      fns_retn = ctr
      EXIT
   CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
       * ignore
   CASE key >=DECF .AND. key <= DECL    &&CHOICES
      fns_retn = key - DECF + 1
      EXIT
    ENDCASE

ENDDO
*********************
* END OF MENU LOOP *
*********************
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSSTDCO.PRG

Variable	Line number in file						
ANS_:	68	77	99				
ANS_1:	27						
ANS_2:	28						
ANS_3:	29						
ANS_4:	30						
ANS_5:	31						
ANS_6:	32						
ANS_7:	33						
CMULT:	45						
COL:	44						
COUNTER:	65	66	67	68	69		
CTR:	46	76	101	102	103	105	
DECF:	48	107	109	110			
DECL:	49	107	109				
FNSCENTR:	22	23	59	79			
FNSSCRN:	21						
FNSUTIL:	19						
FNS_RETN:	41	105	110				
INDX:	76	77	79	99			
KEY:	82	84	90				
MAX_ANS:	25	42	47	60	66	102	103
MESS_:	79						
MESS_1:	34						
MESS_2:	35						
MESS_3:	36						
MESS_4:	37						
MESS_5:	38						
MESS_6:	39						
MESS_7:	40						
M_LINE:	43	79					
ROW:	42	59	60	68	77	99	
SCRTITLE:	26						
TEMP:	47	49					
X:	67	68					

```
 1 ***********************************************
 2 * FNSSTDMR.PRG                                *
 3 * written by Edson: 26 Sept 1988              *
 4 * Purpose: SELECT SUBJECT AREA FOR STD REPORT *
 5 *&Menu to Select Monthly Reporting Module for Standard Report
 6 * NOTES: Menu choices must start with "A"     *
 7 *        Choices limited to 26  (A-Z)         *
 8 * Public Variables Required:                  *
 9 *    highlight                                *
10 *    color                                    *
11 *    fns_retn                                 *
12 * Procedures Required:                        *
13 *    FNSSCRN, FNSCENTER in FNSUTIL            *
14 * Return variables:                          *
15 *    fns_retn: numeric value representing     *
16 *              selected choice from menu      *
17 ***********************************************
18
19 SET PROCEDURE TO FNSUTIL
20
21 DO FNSSCRN WITH 'FNSSTDMR'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 * DO FNSBELL
25 max_ans = 6
26 scrtitle = "CHOOSE MODULE FOR MONTHLY REPORTING STANDARD REPORT"
27 ans_1  = 'A) CATEGORIES OF CASES REQUIRED TO REPORT MONTHLY '
28 ans_2  = 'B) OPERATING PROCEDURES FOR MONTHLY REPORTING '
29 ans_3  = 'C) MONTHLY REPORTING COSTS '
30 ans_4  = 'D) EFFECTS OF MONTHLY REPORTING '
31 ans_5  = 'E) CONSTRUCTED VARIABLES '
32 ans_6  = 'F) RETURN TO PREVIOUS SCREEN '
33 mess_1  = '         Module 1           '
34 mess_2  = '         Module 2           '
35 mess_3  = '         Module 5           '
36 mess_4  = '         Module 6           '
37 mess_5  = 'Derived from Other Variables'
38 mess_6  = '                            '
39 fns_retn = 0 && initialize return code
40 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
41 m_line = 20  && Prompt Message ROW()
42 col = 1      && Cur choice COL()
43 cmult = 22   && Column Incrementer
44 ctr = 1      && Row positioner
45 temp = LTRIM(RTRIM((STR(max_ans))))
46 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
47 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
48
49 SET ESCA OFF
50
51 ******************************
52 * PRINT SCREEN TITLE AND BOX *
53 ******************************
54 *SET COLOR TO &highlight
55 *@ row-2,32 SAY scrtitle
56 *SET COLOR TO &color
57 DO FNSCENTR WITH '&scrtitle',row-2
58 @ row-1,12 TO max_ans+row,70 DOUBLE
59
60 ***********************
61 * PRINT ANSWER LINES *
62 ***********************
63 counter = 1
64 DO WHILE counter < max_ans + 1
```

```
    x = LTRIM(STR(counter,2))
    @ row+counter-1,14 SAY ans_&x
    counter = counter + 1
ENDDO

**********************
* START OF MENU LOOP *
**********************
DO WHILE .T.
    indx=LTRIM(STR(ctr,2))                  && Current Position of Cursor
    @ row+(ctr-1),14 GET ans_&indx          && Read answer line

    DO FNSCENTR WITH mess_&indx,m_line      && Print message line
    CLEAR GETS

    key = 0                                 && Wait for user key
    DO WHILE key = 0                        && Wait for user key
       key = INKEY()                         && Wait for user key
    ENDDO                                   && Wait for user key

    ****************************************
    * CONVERT lower case values to upper case *
    ****************************************
    key = IIF((key > 96 .AND. key < 123),(key-32),key)

    *   @ m_line,3 CLEAR TO m_line,77           && Clear message line

    *********************************
    * Evaluate key that was pressed *
    *********************************
    DO CASE
    CASE STR(key,2)$" 5,24"                 && Dn/Up Arrows
       @ row+(ctr-1),14 SAY ans_&indx          && rewrites last selection
       && in standard video
       ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
       ctr=IIF(ctr>max_ans,1,ctr)   && flip to top if down and last line
       ctr=IIF(ctr<1,max_ans,ctr)   && flip to bottom if up and top line
    CASE key = 13                   && <Enter>
       fns_retn = ctr
       EXIT
    CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
        * ignore
    CASE key >=DECF .AND. key <= DECL    &&CHOICES
       fns_retn = key - DECF + 1
       EXIT
    ENDCASE

ENDDO
********************
* END OF MENU LOOP *
********************
SET ESCAPE ON
RETURN
```

VARIABLES CROSS-REF FOR FNSSTDMR.PRG

Variable	Line number in file					
ANS_:	66	75	97			
ANS_1:	27					
ANS_2:	28					
ANS_3:	29					
ANS_4:	30					
ANS_5:	31					
ANS_6:	32					
CMULT:	43					
COL:	42					
COUNTER:	63	64	65	66	67	
CTR:	44	74	99	100	101	103
DECF:	46	105	107	108		
DECL:	47	105	107			
FNSCENTR:	22	23	57	77		
FNSSCRN:	21					
FNSUTIL:	19					
FNS_RETN:	39	103	108			
INDX:	74	75	77	97		
KEY:	80	82	88			
MAX_ANS:	25	40	45	58	64	100 101
MESS_:	77					
MESS_1:	33					
MESS_2:	34					
MESS_3:	35					
MESS_4:	36					
MESS_5:	37					
MESS_6:	38					
M_LINE:	41	77				
ROW:	40	57	58	66	75	97
SCRTITLE:	26					
TEMP:	45	47				
X:	65	66				

```
 1 ************************************************
 2 * FNSSTDQC.PRG                                 *
 3 * written by Edson: 26 Sept 1988               *
 4 * Purpose: SELECT MODULE  AREA FOR STD REPORT *
 5 *&Menu to Select Quality Control Module for Standard Report
 6 * NOTES: Menu choices must start with "A"      *
 7 *        Choices limited to 26  (A-Z)          *
 8 * Public Variables Required:                   *
 9 *    highlight                                 *
10 *    color                                     *
11 *    fns_retn                                  *
12 * Procedures Required:                         *
13 *    FNSSCRN, FNSCENTER in FNSUTIL             *
14 * Return variables:                           *
15 *    fns_retn: numeric value representing      *
16 *              selected choice from menu       *
17 ************************************************
18
19 SET PROCEDURE TO FNSUTIL
20
21 DO FNSSCRN WITH 'FNSSTDQC'
22 DO FNSCENTR WITH 'Use <Arrow> keys to highlight your choice.',22
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 * DO FNSBELL
25 max_ans = 6
26 scrtitle = "CHOOSE MODULE FOR QC STANDARD REPORT"
27 ans_1  = 'A) ORGANIZATIONAL INFORMATION '
28 ans_2  = 'B) COMPOSITION OF QC REVIEWER COSTS '
29 ans_3  = 'C) SUPPLEMENTATION OF DESIGN '
30 ans_4  = 'D) QUALITY CONTROL PROCEDURES '
31 ans_5  = 'E) CONSTRUCTED VARIABLES '
32 ans_6  = 'F) RETURN TO PREVIOUS SCREEN'
33 mess_1  = '           Module 1          '
34 mess_2  = '           Module 3          '
35 mess_3  = '           Module 4          '
36 mess_4  = '           Module 6          '
37 mess_5  = 'Derived from Other Variables'
38 mess_6  = '                             '
39 fns_retn = 0 && initialize return code
40 row = 5 + INT((16 - max_ans)/2)     && Menu ROW() to begin at
41 m_line = 20  && Prompt Message ROW()
42 col = 1      && Cur choice COL()
43 cmult = 22   && Column Incrementer
44 ctr = 1      && Row positioner
45 temp = LTRIM(RTRIM((STR(max_ans))))
46 DECF = ASC(ans_1)       && Decimal equivalent of first selection index
47 DECL = ASC(ans_&TEMP)   && Decimal equivalent of last selection index
48
49 SET ESCA OFF
50
51 ******************************
52 * PRINT SCREEN TITLE AND BOX *
53 ******************************
54 *SET COLOR TO &highlight
55 *@ row-2,32 SAY scrtitle
56 *SET COLOR TO &color
57 DO FNSCENTR WITH '&scrtitle',row-2
58 @ row-1,20 TO max_ans+row,60 DOUBLE
59
60 ***********************
61 * PRINT ANSWER LINES *
62 ***********************
63 counter = 1
64 DO WHILE counter < max_ans + 1
```

```
 65    x = LTRIM(STR(counter,2))
 66    @ row+counter-1,23 SAY ans_&x
 67    counter = counter + 1
 68 ENDDO
 69
 70 ***********************
 71 * START OF MENU LOOP *
 72 ***********************
 73 DO WHILE .T.
 74    indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
 75    @ row+(ctr-1),23 GET ans_&indx         && Read answer line
 76
 77    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
 78    CLEAR GETS
 79
 80    key = 0                                && Wait for user key
 81    DO WHILE key = 0                       && Wait for user key
 82       key = INKEY()                        && Wait for user key
 83    ENDDO                                  && Wait for user key
 84
 85    ****************************************
 86    * CONVERT lower case values to upper case *
 87    ****************************************
 88    key = IIF((key > 96 .AND. key < 123),(key-32),key)
 89
 90    *   @ m_line,3 CLEAR TO m_line,77          && Clear message line
 91
 92    *********************************
 93    * Evaluate key that was pressed *
 94    *********************************
 95    DO CASE
 96    CASE STR(key,2)$" 5,24"               && Dn/Up Arrows
 97       @ row+(ctr-1),23 SAY ans_&indx        && rewrites last selection
 98       && in standard video
 99       ctr=ctr+IIF(STR(key,2)$"24",1,-1)     && if Dn, increment ctr
100       ctr=IIF(ctr>max_ans,1,ctr)    && flip to top if down and last line
101       ctr=IIF(ctr<1,max_ans,ctr)    && flip to bottom if up and top line
102    CASE key = 13                    && <Enter>
103       fns_retn = ctr
104       EXIT
105    CASE key <DECF .OR. key > DECL       &&JUNK (ignore remaining keys)
106        * ignore
107    CASE key >=DECF .AND. key <= DECL    &&CHOICES
108       fns_retn = key - DECF + 1
109       EXIT
110    ENDCASE
111
112 ENDDO
113 *********************
114 * END OF MENU LOOP *
115 *********************
116 SET ESCAPE ON
117 RETURN
```

VARIABLES CROSS-REF FOR FNSSTDQC.PRG

Variable	Line number in file
ANS_:	66 75 97
ANS_1:	27
ANS_2:	28
ANS_3:	29
ANS_4:	30
ANS_5:	31
ANS_6:	32
CMULT:	43
COL:	42
COUNTER:	63 64 65 66 67
CTR:	44 74 99 100 101 103
DECF:	46 105 107 108
DECL:	47 105 107
FNSCENTR:	22 23 57 77
FNSSCRN:	21
FNSUTIL:	19
FNS_RETN:	39 103 108
INDX:	74 75 77 97
KEY:	80 82 88
MAX_ANS:	25 40 45 58 64 100 101
MESS_:	77
MESS_1:	33
MESS_2:	34
MESS_3:	35
MESS_4:	36
MESS_5:	37
MESS_6:	38
M_LINE:	41 77
ROW:	40 57 58 66 75 97
SCRTITLE:	26
TEMP:	45 47
X:	65 66

```
 1 ************************************************
 2 * FNSSTDSC.PRG                                 *
 3 * written by Edson: 26 Sept 1988               *
 4 * Purpose: SELECT SUBJECT AREA FOR STD REPORT *
 5 *&Menu to Select Claims (Survey) Module for Standard Report
 6 * NOTES: Menu choices must start with "A"      *
 7 *        Choices limited to 26  (A-Z)          *
 8 * Public Variables Required:                   *
 9 *    highlight                                 *
10 *    color                                     *
11 *    fns_retn                                  *
12 * Procedures Required:                         *
13 *    FNSSCRN, FNSCENTER in FNSUTIL             *
14 * Return variables:                           *
15 *    fns_retn: numeric value representing      *
16 *              selected choice from menu       *
17 ************************************************
18
19 SET PROCEDURE TO FNSUTIL
20
21 DO FNSSCRN WITH 'FNSSTDSC'
```

[redacted]

```
23 DO FNSCENTR WITH 'Use <Return> key to register that choice.',23
24 * DO FNSBELL
25 max_ans = 9
26 scrtitle = "CHOOSE MODULE FOR CLAIMS (SURVEY) STANDARD REPORT"
27 ans_1  = 'A) AGENCY ORGANIZATION '
28 ans_2  = 'B) ADMINISTRATION CONTROL '
29 ans_3  = 'C) CLAIMS MONITORING '
30 ans_4  = 'D) CLAIMS ESTABLISHMENT '
31 ans_5  = 'E) COLLECTION OF PAYMENTS '
32 ans_6  = 'F) DELINQUENT CLAIMS FOLLOW-UP '
33 ans_7  = 'G) SUSPENSION/TERMINATION OF CLAIMS '
34 ans_8  = 'H) CONSTRUCTED VARIABLES '
35 ans_9  = 'I) RETURN TO PREVIOUS SCREEN '
36 mess_1  = '         Module 1           '
37 mess_2  = '         Module 2           '
38 mess_3  = '         Module 3           '
39 mess_4  = '         Module 6           '
40 mess_5  = '         Module 7           '
41 mess_6  = '         Module 8           '
42 mess_7  = '         Module 9           '
43 mess_8  = 'Derived from Other Variables'
44 mess_9  = '                            '
45 fns_retn = 0 && initialize return code
```

```
 65
 66 ***********************
 67 * PRINT ANSWER LINES *
 68 ***********************
 69 counter = 1
 70 DO WHILE counter < max_ans + 1
 71    x = LTRIM(STR(counter,2))
 72    @ row+counter-1,23 SAY ans_&x
 73    counter = counter + 1
 74 ENDDO
 75
 76 ***********************
 77 * START OF MENU LOOP *
 78 ***********************
 79 DO WHILE .T.
 80    indx=LTRIM(STR(ctr,2))                 && Current Position of Cursor
 81    @ row+(ctr-1),23 GET ans_&indx         && Read answer line
 82
 83    DO FNSCENTR WITH mess_&indx,m_line     && Print message line
 84    CLEAR GETS
 85
 86    key = 0                                && Wait for user key
 87    DO WHILE key = 0                       && Wait for user key
 88       key = INKEY()                        && Wait for user key
 89    ENDDO                                  && Wait for user key
 90
 91    ****************************************
 92    * CONVERT lower case values to upper case *
 93    ****************************************
 94    key = IIF((key > 96 .AND. key < 123),(key-32),key)
 95
 96    *   @ m_line,3 CLEAR TO m_line,77          && Clear message line
 97
 98    *********************************
 99    * Evaluate key that was pressed *
100    *********************************
101    DO CASE
102    CASE STR(key,2)$" 5,24"                && Dn/Up Arrows
103       @ row+(ctr-1),23 SAY ans_&indx         && rewrites last selection
104       && in standard video
105       ctr=ctr+IIF(STR(key,2)$"24",1,-1)      && if Dn, increment ctr
106       ctr=IIF(ctr>max_ans,1,ctr)    && flip to top if down and last line
107       ctr=IIF(ctr<1,max_ans,ctr)    && flip to bottom if up and top line
108    CASE key = 13                    && <Enter>
109       fns_retn = ctr
110       EXIT
111    CASE key <DECF .OR. key > DECL        &&JUNK (ignore remaining keys)
112       * ignore
113    CASE key >=DECF .AND. key <= DECL     &&CHOICES
114       fns_retn = key - DECF + 1
115       EXIT
116    ENDCASE
117
118 ENDDO
119 ********************
120 * END OF MENU LOOP *
121 ********************
122 SET ESCAPE ON
123 RETURN
```

VARIABLES CROSS-REF FOR FNSSTDSC.PRG

Variable	Line number in file
ANS_:	72 81 103
ANS_1:	27
ANS_2:	28
ANS_3:	29
ANS_4:	30
ANS_5:	31
ANS_6:	32
ANS_7:	33
ANS_8:	34
ANS_9:	35
CMULT:	49
COL:	48
COUNTER:	69 70 71 72 73
CTR:	50 80 105 106 107 109
DECF:	52 111 113 114
DECL:	53 111 113
FNSCENTR:	22 23 63 83
FNSSCRN:	21
FNSUTIL:	19
FNS_RETN:	45 109 114
INDX:	80 81 83 103
KEY:	86 88 94
MAX_ANS:	25 46 51 64 70 106 107
MESS_:	83
MESS_1:	36
MESS_2:	37
MESS_3:	38
MESS_4:	39
MESS_5:	40
MESS_6:	41
MESS_7:	42
MESS_8:	43
MESS_9:	44
M_LINE:	47 83
ROW:	46 63 64 72 81 103
SCRTITLE:	26
TEMP:	51 53
X:	71 72

```
 1 * FNSUPD1.PRG *
 2 * written on 5 June 1988
 3 * modified: Edson 6/14/88
 4 * by Richard Lieberman
 5 *&Module for Updating Data Items
 6 SET PROCEDURE TO FNSUTIL
 7 SET CONFIRM ON
 8 DO FNSSCRN WITH 'FNSUPD1'
 9
10 USE DATADICT INDEX IDNUM,DATADICT,NAME
11
12 SET COLOR TO &highlight
13 @5,31 SAY "DATA UPDATE MODULE"
14 @7,6 SAY "ID NUM"
15 @7,15 SAY "FIELD NAME"
16 @7,37 SAY "FIELD DESCRIPTION"
17 @7,70 SAY "SUBJECT"
18 SET COLOR TO &color
19
20 DO WHILE .T.
21
22    prow = 9
23    @8,1 CLEAR TO 20,78
24
25    STORE SPACE(8) TO lookvar
26    @23,1 CLEAR TO 23,78
27    @23,18 SAY "Enter Variable Name or Variable Number Above"
28    SET COLOR TO &highlight
29    @24,3 SAY "[Ctrl][W]=Exit"
30    SET COLOR TO &color
31
32    @22,2 SAY SPACE(77)
33    @22,35 GET lookvar PICTURE REPLICATE("!",8)
34    READ
35
36    IF READKEY() = 14 .OR. READKEY() = 270
37       STORE SPACE(1) TO answer
38       @22,1 CLEAR TO 23,78
39       DO FNSCENTR WITH "Do you wish to exit and return to main menu";
39        + " (Y/N)?",22
40       DO FNSCENTR WITH "Type [Y] to exit, [N] to continue",23
41       SET CONFIRM OFF &&2/18/89
42       @24,3 SAY REPLICATE(CHR(205),15)
43       @23,58 GET answer PICTURE "Y"
44       READ
45       SET CONFIRM ON  &&2/18/89
46       IF answer = "Y"
47          RELEASE ALL
48          SET CONFIRM OFF  &&2/18/89
49          RETURN
50       ELSE
51          @22,1 CLEAR TO 23,78
52          LOOP
53       ENDIF
54    ENDIF
55
56    IF .NOT. ISALPHA("&lookvar")
57       USE DATADICT INDEX IDNUM,DATADICT,NAME
58       SET ORDER TO 1
59       SEEK VAL(SUBSTR(lookvar,1,8))
60       IF FOUND()
61          @prow,5 SAY STR(IDNUM,4)
62          @prow,15 SAY FIELD_NAME
63          @prow,30 SAY DESCRIP
```

```
64          @prow,72 SAY TRIM(SUBJECT)
65          varfile = FILENAME
66          varnum = IDNUM
67          varname = FIELD_NAME
68          subject = A->SUBJECT
69          descrip = A->DESCRIP
70          varlen  = A->FIELD_LEN
71          discrete = A->YESNO
72       ELSE
73          DO FNSBELL
74          DO FNSCENTR WITH "*** ID NUMBER NOT FOUND ***",22
75          DO FNSSLEEP
76          LOOP
77       ENDIF
78    ELSE
79       SET EXACT ON
80       USE DATADICT INDEX IDNUM,DATADICT,NAME
81       COUNT FOR FIELD_NAME = "&lookvar" TO samevar
82       IF samevar > 1
83          cl = 'TEMP'
84          CALL trpass WITH cl
85          CALL trexe WITH "SAVESCR"
86          DO FNSMENU6
87          DO CASE
88          CASE fns_retn = 1
89             subject = 'ACS'
90          CASE fns_retn = 2
91             subject = 'QC'
92          CASE fns_retn = 3
93             subject = 'MR'
94          CASE fns_retn = 4
95             subject = 'CLA'
96          CASE fns_retn = 5
97             subject = 'COM'
98          CASE fns_retn = 6
99             subject = 'SUR'
100         ENDCASE
101         cl = 'TEMP'
102         CALL trpass WITH cl
103         CALL trexe WITH "RESTSCR"
104         SET ORDER TO 2
105         SEEK "&subject&lookvar"
106      ELSE
107         SET ORDER TO 3
108         SEEK "&lookvar"
109      ENDIF
110      IF FOUND()
111         @prow,2 SAY SPACE(25)
112         @prow,5 SAY STR(IDNUM,4)
113         @prow,15 SAY FIELD_NAME
114         @prow,30 SAY DESCRIP
115         @prow,72 SAY TRIM(SUBJECT)
116         varname = FIELD_NAME
117         varfile = FILENAME
118         varnum = IDNUM
119         subject = A->SUBJECT
120         descrip = A->DESCRIP
121         varlen  = A->FIELD_LEN
122         discrete = A->YESNO
123      ELSE
124         DO FNSBELL
125         DO FNSCENTR WITH "*** VARIABLE NAME NOT FOUND ***",22
126         DO FNSSLEEP
127         LOOP
128      ENDIF
129   ENDIF
130
```

```
131     STORE SPACE(2) TO statevar
132     @23,1 CLEAR TO 23,78
133     @23,23 SAY "Enter 2-Character State Name Above"
134     SET COLOR TO &highlight
135     @24,3 SAY "[Ctrl][W]=Exit"
136     SET COLOR TO &color
137
138     @22,2 SAY SPACE(77)
139     @22,39 GET statevar PICTURE REPLICATE("!",2)
140     READ
141
142     IF READKEY() = 14 .OR. READKEY() = 270
143        STORE SPACE(1) TO answer
144        @22,1 CLEAR TO 23,78
145        DO FNSCENTR WITH "Do you wish to exit and return to main menu";
145         + " (Y/N)?",22
146        DO FNSCENTR WITH "Type [Y] to exit, [N] to continue",23
147        SET CONFIRM OFF &&2/18/89
148        @24,3 SAY REPLICATE(CHR(205),15)
149        @23,58 GET answer PICTURE "Y"
150        READ
151        SET CONFIRM ON &&2/18/89
152        IF answer = "Y"
153           RELEASE ALL
154           SET CONFIRM OFF   &&2/18/89
155           RETURN
156        ELSE
157           @22,1 CLEAR TO 23,78
158        ENDIF
159     ELSE
160        USE FNSSTATE
161        LOCATE FOR ABBREV = statevar
162        IF FOUND()
163           @22,2 CLEAR TO 23,78
164           code = STATECODE
165        ELSE
166           DO FNSBELL
167           DO FNSCENTR WITH "*** STATE NAME NOT FOUND ***",22
168           DO FNSSLEEP
169           LOOP
170        ENDIF
171        USE &varfile
172        ***** Select desired system for states with multiple systems ***
173        COUNT FOR STATE = code to numsys
174        IF NUMSYS < 2
175           LOCATE FOR STATE = code
176           rowsys = RECNO()
177        ELSE
178           store 0 to rowsys
179           DO FNSMENU7
180           USE     && close any open files
181           USE &varfile
182           LOCATE FOR ROWNUM = rowsys
183           * restore screen
184           @8,1 CLEAR TO 20,78
185           SET COLOR TO &highlight
186           @5,31 SAY "DATA UPDATE MODULE"
187           @7,6 SAY "ID NUM"
188           @7,15 SAY "FIELD NAME"
189           @7,37 SAY "FIELD DESCRIPTION"
190           @7,70 SAY "SUBJECT"
191           SET COLOR TO &color
192           @prow,5 SAY STR(varnum,4)
193           @prow,15 SAY varname
194           @prow,30 SAY descrip
195           @prow,72 SAY subject
196        ENDIF
```

```
197       IF FOUND()
198          @10,1 CLEAR TO 20,78
199          @12,9 SAY "Current Value:"
200          @12,31 SAY &varname
201          old_value = &varname
202          @14,9 SAY "New Value:"
203          STORE &varname TO new_value
204          @22,1 CLEAR TO 23,78
205          @22,13 SAY "Enter Updated Value or Code Above, then Press";
205           + " [Return]"
206          SET COLOR TO &highlight
207          @24,3 SAY "[Ctrl][W]=Exit"
208          SET COLOR TO &color
209          ****************************
210          IF TYPE ("&VARNAME") = "N"
211             IF discrete
212                @16,23 SAY "DISCRETE (YES/NO) AND MISSING CODES"
213                @17,8 TO 17,72
214                @18,9 SAY "  1 =YES    -1 =DON'T KNOW    -3 =REFUSED     ";
214                 + "        -5 =MISSING"
215                @19,9 SAY "  0 =NO     -2 =SKIP          -4 =NOT";
215                 + " APPLICABLE     -9 =MISSING"
216                @15,8 to 20,72
217             ELSE
218                @16,22 SAY "MISSING CODES FOR NUMERIC VARIABLES"
219                @17,12 TO 17,68
220                @18,13 SAY "  -1 =DON'T KNOW    -3 =REFUSED           ";
220                 + "-5 =MISSING"
221                @19,13 SAY "  -2 =SKIP          -4 =NOT APPLICABLE    ";
221                 + "-9 =MISSING"
222                @15,12 to 20,68
223             ENDIF
224          ENDIF
225          *****************
226          IF TYPE("&varname") = "N"                                  &&2/17/89
227             @14,31 GET new_value PICTURE REPLICATE("9",varlen) &&2/17/89
228          ELSE                                                     &&2/17/89
229             @14,31 GET new_value PICTURE REPLICATE("!",varlen) &&2/17/89
230          ENDIF                                                     &&2/17/89
231          READ
232          IF READKEY() = 14 .OR. READKEY() = 270
233             STORE SPACE(1) TO answer
234             @22,1 CLEAR TO 23,78
235             DO FNSCENTR WITH "Do you wish to exit and return to main";
235              + " menu (Y/N)?",22
236             DO FNSCENTR WITH 'Type [Y] to exit, [N] to continue',23
237             SET CONFIRM OFF  &&2/18/89
238             @24,3 SAY REPLICATE(CHR(205),15)
239             @23,58 GET answer PICTURE "Y"
240             READ
241             SET CONFIRM ON &&2/18/89
242             IF answer = "Y"
243                RELEASE ALL
244                SET CONFIRM OFF &&2/18/89
245                RETURN
246             ELSE
247                @23,1 SAY SPACE(77)
248                LOOP
249             ENDIF
250          ELSE
251             IF new_value <> &varname
252                DO FNSBELL
253                @22,2 CLEAR TO 23,78
254                @22,14 SAY "Press [Return] to Confirm Change,";
254                 + " [Ctrl][W] to Cancel"
255                READ
256                IF READKEY() <> 14 .AND. READKEY() <> 270
```

```
257                   REPLACE &varname WITH new_value
258                   ********** start of history file update
259                   USE FNSLOG
260                   APPEND BLANK
261                   REPLACE IDNUM WITH varnum
262                   REPLACE DATE WITH mdate
263                   REPLACE STATE WITH code
264
265                   REPLACE ROW WITH rowsys
266                   REPLACE FILE WITH varfile
267                   IF TYPE("old_value") = "N"
268                      REPLACE OLDVALUE WITH STR(old_value,10)
269                   ELSE
270                      REPLACE OLDVALUE WITH old_value
271                   ENDIF
272                   IF TYPE("new_value") = "N"
273                      REPLACE NEWVALUE WITH STR(new_value,10)
274                   ELSE
275                      REPLACE NEWVALUE WITH new_value
276                   ENDIF
277                   USE
278                   ********** end of history file update
279                   LOOP
280                ELSE
281                   LOOP
282                ENDIF
283             ENDIF
284          ENDIF
285       ENDIF   &&found
286    ENDIF   &&readkey
287 ENDDO
288 RELEASE ALL
289 SET CONFIRM OFF  &&2/18/89
290 RETURN
```

VARIABLES CROSS-REF FOR FNSUPD1.PRG

Variable	Line number in file
A:	68 69 70 71 119 120 121 122
ABBREV:	161
ANSWER:	37 43 46 143 149 152 233 239 242
C1:	83 84 101 102
CODE:	164 173 175 263
DATADICT:	10 57 80
DATE:	262
DESCRIP:	63 69 114 120 194
DISCRETE:	71 122 211
FIELD_LEN:	70 121
FIELD_NAME:	62 67 81 113 116
FILENAME:	65 117
FNSBELL:	73 124 166 252
FNSCENTR:	39 40 74 125 145 146 167 235 236
FNSLOG:	259
FNSMENU6:	86
FNSMENU7:	179
FNSSCRN:	8
FNSSLEEP:	75 126 168
FNSSTATE:	160
FNSUTIL:	6
FNS_RETN:	88 90 92 94 96 98
IDNUM:	10 57 66 80 118 261
LOOKVAR:	25 33 56 59
MDATE:	262
NAME:	10 57 80
NEWVALUE:	273 275
NEW_VALUE:	203 227 229 251 257 272 275
NUMSYS:	173 174
OLDVALUE:	268 270
OLD_VALUE:	201 267 270
PROW:	22 61 62 63 64 111 112 113 114 115 192 193 194 195
ROW:	265
ROWNUM:	182
ROWSYS:	176 178 182 265
SAMEVAR:	81 82
STATE:	173 175 263
STATECODE:	164
STATEVAR:	131 139 161
SUBJECT:	68 89 91 93 95 97 99 119 195
TREXE:	85 103
TRPASS:	84 102
VARFILE:	65 117 171 181 266
VARLEN:	70 121 227 229
VARNAME:	67 116 193 200 201 203 210 226 251 257
VARNUM:	66 118 261
YESNO:	71 122

```
 1 * FNSUTIL.PRG *
 2 * by R. Lieberman *
 3 * written on 17 March 1988
 4 * PROCEDURE FILE OF COMMONLY USED UTILITIES
 5
 6
 7 ************************************************************************;
 7 ************
 8
 9 * FNSSCRN.PRG *
10 * by R. Lieberman *
11 * written on 17 March 1988 *
12 * STANDARD SCREEN GENERATOR *
13
14 PROCEDURE FNSSCRN
15 *&Generates Standard Screen
16 PARAMETERS progname
17
18 @0,0 CLEAR
19 *@0,1 SAY '&progname'+' version 1.1A'
20 @0,1 SAY 'VERSION 2.0'
21 @0,70 SAY mdate
22 STORE "FOOD AND NUTRITION SERVICE" TO line1
23 STORE 80 TO length1
24 CALL trpass  WITH line1
25 CALL trpass  WITH length1
26 CALL trexe   WITH "CENTER"
27 @2,0 SAY tr_retc
28 STORE "FOOD STAMP PROGRAM OPERATIONS STUDY DATA ACCESS SYSTEM" TO line2
29 CALL trpass with line2
30 CALL trpass with length1
31 CALL trexe   WITH "CENTER"
32 @3,0 SAY tr_retc
33 @1,0 to 24,79 DOUBLE
34 @4,0 SAY CHR(199)
35 @4,1 TO 4,78
36 @4,79 SAY CHR(182)
37 @21,0 SAY CHR(199)
38 @21,1 TO 21,78
39 @21,79 SAY CHR(182)
40 *@24,5 SAY "[F1] Help"
41
42 RETURN
43
44
45 ************************************************************************;
45 ************
46
47 FNSBELL.PRG *                        && After RETURN never executed
48 * by R. Lieberman *
49 * written on 18 March 1988 *
50 * RINGS BELL *
51
52 PROCEDURE FNSBELL
53 *&Rings Bell
54
55 ?? CHR(7)
56
57 RETURN
58
59
60 ************************************************************************;
60 ************
61
```

```
* FNSSLEEP.PRG *
* by R. Lieberman *
* written on 11 April 1988 *
* HALTS PROGRAM EXECUTION UNTIL USER PROMPT *

PROCEDURE FNSSLEEP
*&Pauses for Input from User

DO FNSCENTR WITH "Press [Return] to Continue",23
@8,0  SAY CHR(186)
key = 0
DO WHILE key<>13
   key=INKEY()
ENDDO
*WAIT "" TO Y
@9,0  SAY CHR(186)

RETURN


*************************************************************************;
************

* FNSCENTR.PRG *
* by R. Lieberman *
* written on 11 April 1988 *
* CENTERS TEXT WITHIN BOX *

PROCEDURE FNSCENTR
*&Centers Text on Screen
PARAMETERS a,b

*** a = text to be centered
*** b = row to written to

STORE 80 TO length
CALL Trpass WITH a
CALL Trpass with length
CALL Trexe   WITH "CENTER"

@b,1 SAY SPACE(77)
@b,0 SAY tr_retc
@b,0 SAY CHR(186)
@b,79 SAY CHR(186)

RETURN

*************************************************************************;
************

* FNSDEBUG.PRG *
* by R. Lieberman *
* written on 27 April 1988 *
* ACTIVATES DEBUG CODE *


PROCEDURE FNSDEBUG
*&Activates Debug Code
PARAMETERS debug

IF debug
   WAIT
   SET TALK ON
   SET ECHO ON
   SET STEP ON
ELSE
   SET STEP OFF
```

```
127     SET TALK OFF
128     SET ECHO OFF
129 ENDIF
130
131 RETURN
132
133 ************************************************************************;
133 ***********
134
135 * PREPARE.PRG *
136 * written by R. Lieberman *
137 * written on 15 April 1988 *
138 * PREPARES FOR dBASE III ARRAY DEFINITION *
139
140
141 PROCEDURE PREPARE
142 *&Prepares for dBASE III Array Definition
143 PUBLIC arraytable
144
145 STORE 300 TO numarray
146 CALL Trpass  WITH numarray
147 CALL Trexe   WITH "ALLOCATE"
148 *DO ALLOCATE WITH 300 && intialize array to store array descriptors
149 arraytable = tr_retc   && save address allocated
150
151 CALL trpass WITH arraytable
152 CAll trexe  WITH "SEPARATE"
153 segment = tr_retc
154 offset = tr_retn
155 STORE SPACE(254) TO m_space
156 arrayend = offset + 301
157
158 DO WHILE offset < arrayend
159     CALL trpass WITH m_space
160     CAll trexe  WITH "retc"
161     CALL Trpass WITH segment
162     CALL Trpass WITH offset
163     CALL Trexe WITH TYPE("offset")
164     CALL Trpass WITH tr_retc
165     CALL Trexe WITH "POKESTR"
166     offset = offset + 254
167 ENDDO
168
169 RETURN
170 ************************************************************************;
170 ***********
171
172 * DEF_ARR.PRG *
173 * written by R. Lieberman *
174 * written on 15 April 1988 *
175 * DEFINE dBASE III ARRAYS *
176
177
178 PROCEDURE DEF_ARR
179 *&Defines dBASE III Arrays
180 PARAMETERS name,arrtype,numelt,length
181
182 DO FIND_ARR WITH  name
183
184 IF rtcode
185     RELEASE ALL
186     RETURN
187 ENDIF
188
189 DO CASE
190 CASE arrtype = "BYTE"
191     CALL Trpass WITH numelt
```

```
192     CALL Trexe WITH "ALLOCATE"
193  CASE arrtype = "CHAR"
194     CALL Trpass WITH numelt
195     CALL Trexe WITH "ALLOCATE"
196  CASE arrtype = "INT"
197     y = numelt * 2
198     CALL Trpass WITH y
199     CALL Trexe WITH "ALLOCATE"
200  CASE arrtype = "LONG"
201     y = numelt * 2
202     CALL Trpass WITH y
203     CALL Trexe WITH "ALLOCATE"
204  CASE arrtype = "DBL"
205     y = numelt * 8
206     CALL Trpass WITH y
207     CALL Trexe WITH "ALLOCATE"
208  CASE arrtype = "SCR"
209     screen = 4096
210     CALL Trpass WITH screen
211     CALL Trexe WITH "ALLOCATE"
212  CASE arrtype = "STR"
213     IF TYPE("length") <> "N"
214        length = VAL(length)
215     ENDIF
216     length = IIF(length > 200,200,length)
217     y = numelt * length
218     CALL Trpass WITH y
219     CALL Trexe WITH "ALLOCATE"
220  ENDCASE
221
222  IF "" = tr_retc   && null means an error occurred
223     * DO DEALLOC WITH location
224     RETURN
225  ELSE
226     location = tr_retc
227     CALL trpass WITH arraytable
228     CAll trexe WITH "SEPARATE"
229     segment = tr_retc
230     offset  = tr_retn + (num_array * 30)
231     STORE SPACE(29) TO array_id
232     array_id = STUFF(array_id,1,8,location)
233     array_id = STUFF(array_id,9,10,name)
234     array_id = STUFF(array_id,19,4,arrtype)
235     array_id = STUFF(array_id,23,3,STR(numelt,3,0))
236     array_id = STUFF(array_id,26,4,STR(length,4,0))
237     CALL trpass WITH array_id
238     CAll trexe  WITH "retc"
239     CALL Trpass WITH segment
240     CALL Trpass WITH offset
241     CALL Trexe WITH TYPE("offset")
242     CALL Trpass WITH tr_retc
243     CALL Trexe WITH "POKESTR"
244     num_array = num_array + 1
245  ENDIF
246
247
248  RETURN
249
250
251  ************************************************************************;
251  ************
252
253  * TO_ARRAY.PRG *
254  * written by R. Lieberman *
255  * written on 22 April 1988 *
256  * STORE DATA TO DBASE III ARRAYS *
257
```

```
258
259 PROCEDURE TO_ARRAY
260 *&Stores Data in dBASE III Arrays
261 PARAMETERS x, y, z
262
263 PUBLIC name,element,value
264
265 STORE x to name
266 STORE y to element
267 STORE z to value
268
269 STORE SPACE(10) TO arrayname
270
271 CALL trpass WITH arraytable
272 CAll trexe  WITH "SEPARATE"
273 segment = tr_retc
274 offset = tr_retn
275
276
277 PUBLIC ARRTYPE,NUMELT,LOCATION,LENGTH
278
279 DO WHILE .T.
280    CALL Trpass WITH segment
281    CALL Trpass WITH offset
282    CALL Trexe WITH "PEEKSTR"
283    IF TRIM(SUBSTR(tr_retc,9,10)) = name
284       arrtype = SUBSTR(tr_retc,19,4)
285       numelt  = SUBSTR(tr_retc,23,3)
286       location = SUBSTR(tr_retc,1,8)
287       length = VAL(SUBSTR(tr_retc,24,3))
288       EXIT
289    ELSE
290       IF SUBSTR(tr_retc,1,8) <> SPACE(8)
291          offset = offset + 30
292          LOOP
293       ELSE
294          WAIT "ARRAY NAME NOT FOUND - PERFORM ERROR PROCESSING"
295          EXIT
296       ENDIF
297    ENDIF
298 ENDDO
299
300 IF element > VAL(numelt) .OR. element < 1
301    WAIT "ARRAY ELEMENT OUT OF BOUNDS"
302 ENDIF
303
304 CALL trpass WITH location
305 CAll trexe  WITH "SEPARATE"
306 segment = tr_retc
307 DO CASE
308 CASE arrtype = "BYTE"
309    IF TYPE("value") = "C"
310       IF VAL(value) > 15
311          WAIT "ARRAY DEFINED FOR 1-BYTE HEX STRINGS ONLY"
312       ENDIF
313    ELSE
314       IF value > 15
315          WAIT "ARRAY DEFINED FOR 1-BYTE HEX STRINGS ONLY"
316       ENDIF
317    ENDIF
318    offset = tr_retn + (element - 1)
319 CASE arrtype = "CHAR"
320    IF TYPE("value") = "N"
321       IF LEN(LTRIM(STR(value,5,0))) > 1
322          WAIT "ARRAY DEFINED FOR 1-BYTE ASCII STRINGS ONLY"
323       ENDIF
324    ENDIF
```

```
325    offset = tr_retn + (element - 1)
326    tr_retc = value
327 CASE arrtype = "INT"
328    IF TYPE("value") = "C"
329       WAIT "INVALID ARRAY FORMAT"
330    ENDIF
331    offset = tr_retn + ((element - 1)*2)
332 CASE arrtype = "LONG"
333    IF TYPE("value") = "C"
334       WAIT "INVALID ARRAY FORMAT"
335    ENDIF
336    offset = tr_retn + ((element - 1)*4)
337 CASE arrtype = "DBL"
338    IF TYPE("value") = "C"
339       WAIT "INVALID ARRAY FORMAT"
340    ENDIF
341    offset = tr_retn + ((element - 1)*8)
342 CASE arrtype = "SCR"
343    offset = tr_retn + ((element - 1)*4096)
344 CASE arrtype = "STR"
345    IF TYPE("value") = "C"
346       IF LEN(value) > (length - 1)
347          WAIT "ARRAY ELEMENT WIDTH INSUFFICIENT FOR PASSED ARGUMENT"
348       ENDIF
349    ELSE
350       IF LEN(LTRIM(STR(value,15,2))) > (length - 1)
351          WAIT "ARRAY ELEMENT WIDTH INSUFFICIENT FOR PASSED ARGUMENT"
352       ENDIF
353    ENDIF
354    offset = tr_retn + ((element - 1) * length)
355    CALL trpass WITH value
356    CAll trexe WITH "retc"
357 ENDCASE
358
359 CALL Trpass WITH segment
360 CALL Trpass WITH offset
361 CALL Trexe WITH TYPE("offset")
362 CALL Trpass WITH tr_retc
363 CALL Trexe WITH "POKESTR"
364
365 RETURN
366
367
368 ************************************************************************;
368 ************
369
370 * FIND_ARR.PRG *
371 * written by R. Lieberman *
372 * written on 18 May 1988 *
373 * FIND ABSOLUTE ADDRESS OF ARRAY *
374
375
376 PROCEDURE FIND_ARR
377 *&Finds Absolute Address of Array
378 PARAMETERS name
379
380 rtcode = .F.
381
382 CALL trpass WITH arraytable
383 CAll trexe  WITH "SEPARATE"
384 segment = tr_retc
385 offset      = tr_retn
386
387 IF num_array > 0
388    arrayend = (offset * num_array)+1
389 ELSE
390    arrayend = offset + 1
```

```
391 ENDIF
392
393 DO WHILE offset < arrayend
394    CALL Trpass WITH segment
395    CALL Trpass WITH offset
396    CALL Trexe WITH "PEEKSTR"
397    IF TRIM(SUBSTR(tr_retc,9,10)) <> "&name"
398       offset = offset + 30
399       LOOP
400    ELSE
401       segment = SUBSTR(tr_retc,1,4)
402       offset = SUBSTR(tr_retc,5,4)
403       array_id = SUBSTR(tr_retc,1,29)
404       arr_len = VAL(SUBSTR(tr_retc,26,4))
405       CALL trpass WITH offset
406       CALL trexe WITH "DEC"
407       offset = tr_retn
408       rtcode = .T.
409       EXIT
410    ENDIF
411 ENDDO
412
413 RETURN
414
415
416 ************************************************************************;
416 ***********
417
418 * STO_SCRN.PRG *
419 * written by R. Lieberman *
420 * written on 17 May 1988 *
421 * STORE SCREEN TO DBASE III ARRAY *
422
423
424 PROCEDURE STO_SCRN
425 *&Stores Screen to dBASE III ARRAY
426 PARAMETERS name
427
428
429 CALL trpass WITH arraytable
430 CAll trexe  WITH "SEPARATE"
431 segment = tr_retc
432 offset      = tr_retn
433
434 DO WHILE .T.
435    CALL Trpass WITH segment
436    CALL Trpass WITH offset
437    CALL Trexe WITH "PEEKSTR"
438    IF TRIM(SUBSTR(tr_retc,9,10)) <> name
439       offset = offset + 30
440       LOOP
441    ELSE
442       segment = SUBSTR(tr_retc,1,4)
443       offset  = SUBSTR(tr_retc,5,4)
444       Call Trpass WITH segment
445       CALL Trpass WITH offset
446       CALL Trexe WITH TYPE( "offset" )
447       CALL Trexe WITH "POKESCR"
448       EXIT
449    ENDIF
450 ENDDO
451
452
453 RETURN
454
455 ************************************************************************;
455 ***********
```

```
456
457 * RST_SCRN.PRG *
458 * written by R. Lieberman *
459 * written on 17 May 1988 *
460 * RESTORES SCREEN FROM DBASE III ARRAY *
461
462
463 PROCEDURE RST_SCRN
464 *&Restores Screen from dBASE III Array
465 PARAMETERS name
466
467
468 CALL trpass WITH arraytable
469 CAll trexe  WITH "SEPARATE"
470 segment = tr_retc
471 offset      = tr_retn
472
473 DO WHILE .T.
474    CALL Trpass WITH segment
475    CALL Trpass WITH offset
476    CALL Trexe WITH "PEEKSTR"
477    IF TRIM(SUBSTR(tr_retc,9,10)) <> name
478       offset = offset + 30
479       LOOP
480    ELSE
481       segment = SUBSTR(tr_retc,1,4)
482       offset  = SUBSTR(tr_retc,5,4)
483       Call Trpass WITH segment
484       CALL Trpass WITH offset
485       CALL Trexe WITH TYPE( "offset" )
486       CALL Trexe WITH "PEEKSCR"
487       EXIT
488    ENDIF
489 ENDDO
490 RETURN
```

VARIABLES CROSS-REF FOR FNSSCRN

Variable	Line number in file
FNSBELL.PR:	47
LENGTH1:	23 25 30
LINE1:	22 24
LINE2:	28 29
MDATE:	21
PROGNAME:	16
TREXE:	26 31
TRPASS:	24 25 29 30
TR_RETC:	27 32

VARIABLES CROSS-REF FOR FNSSLEEP

Variable	Line number in file
FNSCENTR:	70
KEY:	72 74

VARIABLES CROSS-REF FOR FNSCENTR

Variable	Line number in file
A:	91 97
B:	91 101 102 103 104
LENGTH:	96 98
TREXE:	99
TRPASS:	97 98
TR_RETC:	102

VARIABLES CROSS-REF FOR FNSDEBUG

Variable	Line number in file
DEBUG:	118 120

VARIABLES CROSS-REF FOR PREPARE

Variable	Line number in file
ARRAYEND:	156 158
ARRAYTABLE:	143 149 151
M_SPACE:	155 159
NUMARRAY:	145 146
OFFSET:	154 156 158 162 163 166
SEGMENT:	153 161
TREXE:	147 152 160 163 165
TRPASS:	146 151 159 161 162 164
TR_RETC:	149 153 164
TR_RETN:	154

VARIABLES CROSS-REF FOR DEF_ARR

Variable	Line number in file
ARRAYTABLE:	227
ARRAY_ID:	231 232 233 234 235 236 237
ARRTYPE:	180 190 193 196 200 204 208 212 234
FIND_ARR:	182
LENGTH:	180 213 214 216 217
LOCATION:	226 232
NAME:	180 182 233
NUMELT:	180 191 194 197 201 205 217
NUM_ARRAY:	244
OFFSET:	230 240 241
RTCODE:	184
SCREEN:	209
SEGMENT:	229 239
TREXE:	192 195 199 203 207 211 219 228 238 241 243
TRPASS:	191 194 198 202 206 210 218 227 237 239 240 242
TR_RETC:	222 226 229 242
TR_RETN:	230
Y:	197 198 201 202 205 206 217 218

VARIABLES CROSS-REF FOR TO_ARRAY

Variable	Line number in file
ARRAYNAME:	269
ARRAYTABLE:	271
ARRTYPE:	277 284 308 319 327 332 337 342 344
ELEMENT:	263 266 300 331 336 341 343 354
LENGTH:	277 287 354
LOCATION:	277 286 304
NAME:	263 265 283
NUMELT:	277 285
OFFSET:	274 281 291 318 325 331 336 341 343 354 360 361
SEGMENT:	273 280 306 359
TREXE:	272 282 305 356 361 363
TRPASS:	271 280 281 304 355 359 360 362
TR_RETC:	273 283 287 306 326 362
TR_RETN:	274 318 325 331 336 341 343 354
VALUE:	263 267 309 314 320 326 328 333 338 345 355
X:	261 265
Y:	261 266
Z:	261 267

VARIABLES CROSS-REF FOR FIND_ARR

Variable	Line number in file
ARRAYEND:	388 390 393
ARRAYTABLE:	382
ARRAY_ID:	403
ARR_LEN:	404
NAME:	378
NUM_ARRAY:	387 388
OFFSET:	385 390 393 395 398 402 405 407
RTCODE:	380 408
SEGMENT:	384 394 401
TREXE:	383 396 406
TRPASS:	382 394 395 405
TR_RETC:	384 397 404
TR_RETN:	385 407

VARIABLES CROSS-REF FOR STO_SCRN

Variable	Line number in file
ARRAYTABLE:	429
NAME:	426 438
OFFSET:	432 436 439 443 445
SEGMENT:	431 435 442 444
TREXE:	430 437 446 447
TRPASS:	429 435 436 444 445
TR_RETC:	431 438
TR_RETN:	432

VARIABLES CROSS-REF FOR RST_SCRN

Variable	Line number in file
ARRAYTABLE:	468
NAME:	465 477
OFFSET:	471 475 478 482 484
SEGMENT:	470 474 481 483
TREXE:	469 476 485 486
TRPASS:	468 474 475 483 484
TR_RETC:	470 477
TR_RETN:	471

```
 1 * FNSVAR.PRG *
 2 * by Richard Lieberman *
 3 * written on 8 May 1988 *
 4 *&Initializes Public Variables
 5
 6 elements = 0 && counter of elements selected
 7 num_array = 0
 8 state_1 = '  '  && initialize return codes from SELECTST.PRG
 9 state_2 = '  '  && initialize return codes from SELECTST.PRG
10 state_3 = '  '  && initialize return codes from SELECTST.PRG
11 state_4 = '  '  && initialize return codes from SELECTST.PRG
12 state_5 = '  '  && initialize return codes from SELECTST.PRG
13 state_6 = '  '  && initialize return codes from SELECTST.PRG
14 state_7 = '  '  && initialize return codes from SELECTST.PRG
15 state_8 = '  '  && initialize return codes from SELECTST.PRG
16 state_9 = '  '  && initialize return codes from SELECTST.PRG
17 state_10= '  '  && initialize return codes from SELECTST.PRG
18 disp_choice = 0 && initialize return codes from SELECTST.PRG
19 disp_cat = 0
20 RETURN
```

VARIABLES CROSS-REF FOR FNSVAR.PRG

Variable	Line number in file
DISP_CAT:	19
DISP_CHOIC:	18
ELEMENTS:	6
NUM_ARRAY:	7
STATE_1:	8
STATE_10:	17
STATE_2:	9
STATE_3:	10
STATE_4:	11
STATE_5:	12
STATE_6:	13
STATE_7:	14
STATE_8:	15
STATE_9:	16

```
 1 *********************************
 2 *  makedict.prg
 3 *  creates dictionary from cross-reference dbf
 4 *********************************
 5 SET ESCAPE ON
 6 SET STATUS OFF
 7 SET SCOREBOARD OFF
 8 SET TALK ON
 9 @0,0 CLEAR
10 @8,1 SAY '**** PLEASE BE PATIENT ****'
11 @9,1 SAY '****************************'
12 @10,1 SAY 'Making backup of data dictionary....'
13 USE DATADICT
14 COPY TO DATADOLD
15 @10,0 CLEAR
16 @10,1 SAY 'Backup of old data dictionary stored as DATADOLD.dbf.'
17 @11,1 SAY 'Making new data dictionary...'
18 USE XREF
19 COPY TO DATADICT FIELDS FIELD_NAME,FIELD_TYPE,FIELD_LEN,FIELD_DEC,DES;
19 CRIP,KEYWORD,MODULE,SUBJECT,FILENAME,IDNUM,CONSTRUCT,YESNOFOR EXTRACT
20 CLOSE ALL
21 @11,0 CLEAR
22 @11,1 SAY 'New data dictionary created.                          '
23 @12,1 SAY 'Creating index files....'
24 USE DATADICT
25 INDEX ON SUBJECT-FIELD_NAME TO DATADICT.NDX
26 INDEX ON IDNUM TO IDNUM.NDX
27 INDEX ON FIELD_NAME TO NAME.NDX
28 @12,0 CLEAR
29 @12,1 SAY 'New index files have been created.          '
30 @13,1 SAY 'Creating new standard report definition file...'
31 close all
32 USE XREF
33 COPY TO RPTDEF FIELDS IDNUM,FIELD_NAME,SUBJECT,FILENAME,FIELD_TYPE,FI;
33 ELD_LEN,FIELD_DEC,DESCRIP,MODULE,REPORTNUM FOR EXTRACT .AND.;
33  REPORTNUM > 0
34 CLOSE ALL
35 @13,0 CLEAR
36 @13,1 SAY 'New standard report definition file has been created.'
37 @14,1 SAY 'Processing completed successfully.'
38
```

VARIABLES CROSS-REF FOR MAKEDICT.PRG

Variable	Line number in file			
CONSTRUCT:	19			
DATADICT:	13	19	24	
DATADICT.N:	25			
DATADOLD:	14			
DESCRIP:	19	33		
EXTRACT:	19	33		
FIELD_DEC:	19	33		
FIELD_LEN:	19	33		
FIELD_NAME:	19	25	27	33
FIELD_TYPE:	19	33		
FILENAME:	19	33		
IDNUM:	19	26	33	
IDNUM.NDX:	26			
KEYWORD:	19			
NAME.NDX:	27			
REPORTNUM:	33			
RPTDEF:	33			
SUBJECT:	19	25	33	
XREF:	18	32		
YESNOFOR:	19			

```
 1 *** PACKALL.PRG   ****
 2 *** CHANGES DATE TO CURRENT DATE
 3 use ACS1
 4 delete all for rownum=0
 5 pack
 6 use ACS2
 7 delete all for rownum=0
 8 pack
 9 use ACS3
10 delete all for rownum=0
11 pack
12 use ACS4
13 delete all for rownum=0
14 pack
15 use ACS5
16 delete all for rownum=0
17 pack
18 use ADHOC
19 delete all for STATE=0
20 pack
21 use BACKLIST
22 delete all for FILE_NAME = 'XX'
23 pack
24 use CLAIMS1
25 delete all for rownum=0
26 pack
27 use CLAIMS2
28 delete all for rownum=0
29 pack
30 use CLAIMS3
31 delete all for rownum=0
32 pack
33 use CLAIMS4
34 delete all for rownum=0
35 pack
36 use CLAIMS5
37 delete all for rownum=0
38 pack
39 use CLAIMS6
40 delete all for rownum=0
41 pack
42 use CLAIMS7
43 delete all for rownum=0
44 pack
45 use CLAIMS8
46 delete all for rownum=0
47 pack
48 use CLAIMS9
49 delete all for rownum=0
50 pack
51 use COM
52 delete all for rownum=0
53 pack
54 use DATADICT
55 delete all for FIELD_NAME = 'XX'
56 pack
57 use DATACS
58 delete all for rownum=0
59 pack
60 use DATCLA
61 delete all for rownum=0
62 pack
63 use DATCOM
64 delete all for rownum=0
```

```
 65 pack
 66 use DATMR
 67 delete all for rownum=0
 68 pack
 69 use DATQC
 70 delete all for rownum=0
 71 pack
 72 use DATSUR
 73 delete all for rownum=0
 74 pack
 75 use DISPLAY
 76 delete all for LINE = 'XX'
 77 pack
 78 use DISPLAY2
 79 delete all for LINE = 'XX'
 80 pack
 81 use EXTACS
 82 delete all for FIELD_NAME = 'XX'
 83 pack
 84 use EXTCLA
 85 delete all for FIELD_NAME = 'XX'
 86 pack
 87 use EXTCOM
 88 delete all for FIELD_NAME = 'XX'
 89 pack
 90 use EXTMR
 91 delete all for FIELD_NAME = 'XX'
 92 pack
 93 use EXTQC
 94 delete all for FIELD_NAME = 'XX'
 95 pack
 96 use EXTSUR
 97 delete all for FIELD_NAME = 'XX'
 98 pack
 99 use FNSLOG
100 delete all for USERID = 'XX'
101 pack
102 use FNSSTATE
103 delete all for STATECODE = 0
104 pack
105 use MR1
106 delete all for rownum=0
107 pack
108 use MR2
109 delete all for rownum=0
110 pack
111 use MR3
112 delete all for rownum=0
113 pack
114 use MR4
115 delete all for rownum=0
116 pack
117 use MR5
118 delete all for rownum=0
119 pack
120 use QC1
121 delete all for rownum=0
122 pack
123 use QC2
124 delete all for rownum=0
125 pack
126 use QC3
127 delete all for rownum=0
128 pack
129 use RESULT
130 delete all for STATE = 0
131 pack
```

```
132 use RESULT2
133 delete all for LINE = 'XX'
134 pack
135 use RPTDEF
136 delete all for IDNUM = 0
137 pack
138 use SHELLRE2
139 delete all for LINE = 'XX'
140 pack
141 use STDSTRUC
142 delete all for FIELD_NAME = 'XX'
143 pack
144 use STRUCT
145 delete all for FIELD_NAME = 'XX'
146 pack
147 use STRUCT2
148 delete all for FIELD_NAME = 'XX'
149 pack
150 use SURVEY1
151 delete all for rownum=0
152 pack
153 use SURVEY2
154 delete all for rownum=0
155 pack
156 use SURVEY3
157 delete all for rownum=0
158 pack
159 use SURVEY4
160 delete all for rownum=0
161 pack
162 use SURVEY5
163 delete all for rownum=0
164 pack
165 use SURVEY6
166 delete all for rownum=0
167 pack
168 use SURVEY7
169 delete all for rownum=0
170 pack
171 use VARINFO
172 delete all for FIELD_NAME = 'XX'
173 pack
174 use XREF
175 delete all for FIELD_NAME = 'XX'
176 pack
177 close all
178 run dir *.dbf/p
```

VARIABLES CROSS-REF FOR PACKALL.PRG

Variable	Line number in file
ACS1:	3
ACS2:	6
ACS3:	9
ACS4:	12
ACS5:	15
ADHOC:	18
BACKLIST:	21
CLAIMS1:	24
CLAIMS2:	27
CLAIMS3:	30
CLAIMS4:	33
CLAIMS5:	36
CLAIMS6:	39
CLAIMS7:	42
CLAIMS8:	45
CLAIMS9:	48
COM:	51
DATACS:	57
DATADICT:	54
DATCLA:	60
DATCOM:	63
DATMR:	66
DATQC:	69
DATSUR:	72
DIR:	178
DISPLAY:	75
DISPLAY2:	78
EXTACS:	81
EXTCLA:	84
EXTCOM:	87
EXTMR:	90
EXTQC:	93
EXTSUR:	96
FIELD_NAME:	55 82 85 88 91 94 97 142 145 148 172 175
FILE_NAME:	22
FNSLOG:	99
FNSSTATE:	102
IDNUM:	136
LINE:	76 79 133 139
MR1:	105
MR2:	108
MR3:	111
MR4:	114
MR5:	117
P:	178
QC1:	120
QC2:	123
QC3:	126
RESULT:	129
RESULT2:	132
ROWNUM:	4 7 10 13 16 25 28 31 34 37 40 43 46
	49 52 58 61 64 67 70 73 106 109 112 115 118
	121 124 127 151 154 157 160 163 166 169
RPTDEF:	135
SHELLRE2:	138
STATE:	19 130
STATECODE:	103
STDSTRUC:	141
STRUCT2:	147
SURVEY1:	150
SURVEY2:	153

```
 1 * TRLIB.PRG
 2 * Authors: Alastair Dallas, Ralph Davis, Tom Rettig, Leonard Zerman
 3 * Copyright (c) 1986 Tom Rettig Associates, All Rights Reserved
 4 *
 5 * NOTE: This file is called automatically by TRLIB.EXE when
 6 *       loading dBASE.
 7 *
 8 *& Tom Rettig Library Load/Control Program
 9 SET TALK OFF
10
11 * Set up five public memvars (must be stored in this order).
12 PUBLIC tr_retc, tr_retd, tr_retl, tr_retn, tr_dfmt
13
14 * Return memvars must be initialized before storing anything else.
15 tr_retc = SPACE(254)
16 tr_retd = CTOD("  /  /  ")
17 tr_retl = .F.
18 tr_retn = 0.00
19
20 * Set return memvar addresses.
21 LOAD Trret
22 *                         Order must remain:
23 CALL trret WITH tr_retc     && first
24 CALL trret WITH tr_retd     && second
25 CALL trret WITH tr_retl     && third
26 CALL trret WITH tr_retn     && fourth
27 RELEASE MODULE trret
28
29 * Load trlib variable-passer and function-executor.
30 LOAD Trpass
31 LOAD Trexe
32
33 * Set the date format (must be uppercase).
34 tr_dfmt = "AMERICAN"
35
36 ************************************
37 * Do <your command file name> here:
38 DO FNSMAIN
39 ************************************
40
41 * Return to interactive dBASE if your program does not QUIT.
42 SET TALK ON
43 RETURN
44 * eof
```

VARIABLES CROSS-REF FOR TRLIB.PRG

Variable	Line number in file
FNSMAIN:	38
TREXE:	31
TRPASS:	30
TRRET:	21 23 24 25 26 27
TR_DFMT:	12 34
TR_RETC:	12 15 23
TR_RETD:	12 16 24
TR_RETL:	12 17 25
TR_RETN:	12 18 26