ORNL/TM-11556

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

Characteristics Data Base: Programmer's Guide to the LWR Serial Numbers Data Base

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report [illegible]

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.
NTIS price codes—Printed Copy: A06; Microfiche: A01.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-11556

Chemical Technology Division

CHARACTERISTICS DATA BASE: PROGRAMMER'S GUIDE TO THE LWR SERIAL NUMBERS DATA BASE

E. D. Lewis
R. S. Moore

Automated Sciences Group, Inc.

Date Published - August 1990

Report prepared by
Automated Sciences Group, Inc.
Oak Ridge, TN
under Contract DE-AC05-86OR21642

for

Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6285
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
Office of Civilian Radioactive Waste Management
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0315928 5

TABLE OF CONTENTS

1.0 INTRODUCTION 1

2.0 SYSTEM DEVELOPMENT 2
- 2.1 PROGRAM START-UP 2
- 2.2 RESOURCE REQUIREMENTS 2
- 2.3 DEVELOPMENT ENVIRONMENT AND TOOLS 2
  - 2.3.1 dBASE III Plus 2
  - 2.3.2 Clipper Compiler 3

3.0 MENU FLOW CHART 4

4.0 FILE STRUCTURES 5

5.0 PROGRAMS 12
- 5.1 SOURCE 12
- 5.2 INSTALLATION 15

APPENDIX A PROGRAM LISTINGS A-1

PREFACE

The LWR Serial Numbers Data Base is one of the important elements of the Characteristics Data Base (CDB), which provides the detailed technical characteristics of potential repository wastes.* The User's Guide to the LWR Serial Numbers Data Base will also be published (as part of Revision 1 to the reference cited below), and this report provides the corresponding Programmer's Guide. Other PC data bases and guides available through this task program are:

LWR Radiological Data Base

LWR Assemblies Data Base

LWR NFA Hardware Data Base

High-Level Waste Data Base

LWR Quantities Data Base

Karl J. Notz
Task Manager
CDB Program

*"Characteristics of Spent Fuel, High-Level Waste, and Other Radioactive Wastes Which May Require Long-Term Isolation," DOE/RW-0184, Volumes 1-6 (December 1987) and Volumes 7-8 (June 1988).

Abstract

The LWR Serial Numbers Data Base is a menu-driven PC data base developed as part of OCRWM's waste, technical data base on the characteristics of potential repository wastes, which also includes non-LWR spent fuel, high-level and other materials. This programmer's guide completes the documentation for the LWR Serial Numbers Data Base. The PC data base itself may be requested from the Oak Ridge National Laboratory, using the order form provided in Volume 1 of publication DOE/RW-0184-R1.

1.0 INTRODUCTION

The LWR Serial Numbers Database System (SNDB) contains detailed data about individual, historically discharged LWR spent fuel assemblies. This data includes the reactor where used, the year the assemblies were discharged, the pool where they are currently stored, assembly type, burnup, weight, enrichment, and an estimate of their radiological properties. This information is distributed on floppy disks to users in the nuclear industry to assist in planning for the permanent nuclear waste repository.

The SNDB is menu-driven and very easy to use. People with little or no computer expertise should find it easy to produce reports on the data of interest. For instructions on using the system, please see "User's Guide to the LWR Serial Numbers Database System". Those knowledgeable in the use of PC databases can also take the data files and perform additional manipulations to obtain other reports and/or graphs.

This document describes the design and development of the SNDB. It provides a complete description of the file structures and an outline of the major code modules. It serves as a reference for a programmer maintaining the system, or for others interested in the technical detail of this database. The file structures, in particular, will be useful for those interested in personalized applications programming using the data.

This is the initial version of the SNDB. It contains historical data through Dec. 31, 1987, obtained from the Energy Information Administration (EIA). EIA obtains the data from the utility companies via the RW-859 Survey Form. It evaluates and standardizes the data and distributes the resulting batch level database as a large file on magnetic tape. The Characteristics Data Base obtains this database for use in the LWR Quantities Data Base. Additionally, the CDB obtains the individual assembly level detail from EIA for use in the SNDB. While the Quantities Data Base retains only the level of detail necessary for its reporting, the SNDB does retain and use the batch level data to assist in the identification of a particular assembly serial number. We expect to update the SNDB on an annual basis, as new historical data becomes available.

2.0 SYSTEM DEVELOPMENT

2.1 PROGRAM START-UP

The program is started by typing the command SNDB. This executes a DOS batch file that performs these operations:

1. Changes to the directory where the data and programs are stored.

2. Runs the executable module, SNDB.EXE. This starts the program, and the user can select various reports until he chooses to exit.

3. Returns to the directory that was the default at step 1.

2.2 RESOURCE REQUIREMENTS

The SNDB system requires the following computer resources:

1. 200K of available memory (RAM).

2. 5 Megabytes of disk space.

3. DOS 2.1 or above.

The system is distributed as an executable module with all necessary auxiliary programs. The user does not need to have dBase or any other additional software to run the system.

2.3 DEVELOPMENT ENVIRONMENT AND TOOLS

This section describes briefly the different tools used to produce the SNDB.

2.3.1 dBASE III Plus[1]

The data files used in the SNDB were created using dBase III Plus. The use of dBase has been continued for maintenance and modifications of the file structures as needed.

---

[1]dBase III+ is a product of Ashton Tate Corporation.

2.3.2 Clipper Compiler[2]

The code for the menu system of the SNDB was written in Clipper, Summer 1987 version. Clipper is the compiler for a language similar to dBase. Clipper was chosen because it provides the following advantages over the interpreted dBase language:

1. Enhanced performance

2. The ability to distribute an executable module that will run independently. This avoids requiring users to purchase any additional software.

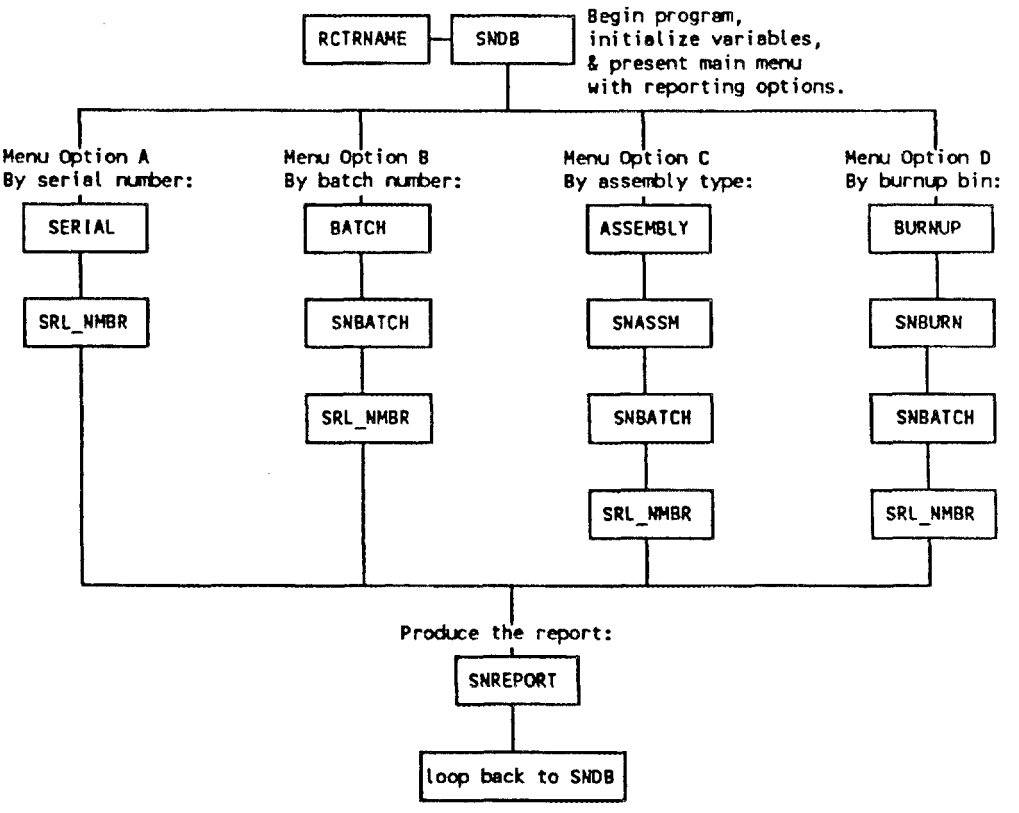
3. Some language extensions that cut programming time.

---

[2]Clipper is a product of Nantucket Corporation.

3.0 MENU FLOW CHART

```
 SSSS   NN   NN DDDDD   BBBBBB
SS  SS  NNN  NN  DD DD   BB  BB
SSS     NNNN NN  DD  DD  BB  BB
 SSS    NN NNNN  DD  DD  BBBBB
   SSS  NN  NNN  DD  DD  BB  BB
SS  SS  NN   NN  DD DD   BB  BB
 SSS    NN   NN DDDDD   BBBBBB
```



sndb.frm

4.0 FILE STRUCTURES

Standard abbreviations were used for Type/Size of Field Names ("C" for character, "N" for numeric, etc).

BURN.DBF

Lookup table for burnup range represented by a particular burnup bin.

FIELD NAME	TYPE/SIZE	DEFINITION
BINN	C2	A number representing a collection of burnups
BURNN	C13	Burnup range

INDEX NAME	INDEXED ON
(none)	

EIA.DBF

Lookup table for the names of the fuel assembly types.

FIELD NAME	TYPE/SIZE	DEFINITION
ATCODE	C4	CDB Assembly Type code
MANUFACT	C22	Manufacturer of assembly
ARRAY	C7	Fuel rod array size
VERSION	C13	Discriminator for more than one assembly w/same name
ASSM_CLASS	C3	CDB Assembly Class code
EIACODE	C5	EIA Assembly Type code
PWR_BWR	C3	'Pressurized' or 'Boiling' Water Reactor

INDEX NAME	INDEXED ON
EIA_CODE	EIACODE

NEUTOTAL.DBF

Number of (alpha,n) and (fission,n) neutrons at various decay times and burnups per metric ton of initial heavy metal.

FIELD NAME	TYPE/SIZE	DEFINITION
REACTOR	C10	'Pressurized' or 'Boiling' Water Reactor and the burnup bin
SOURCE	C1	Source of neutrons: (alpha,n), fission, or total
YR0	C9	Value at discharge
YR1	C9	Value 1 year from discharge
YR2	C9	Value 2 years from discharge
YR3	C9	Value 3 years from discharge
YR4	C9	Value 4 years from discharge
YR5	C9	Value 5 years from discharge
YR6	C9	Value 6 years from discharge
YR7	C9	Value 7 years from discharge
YR8	C9	Value 8 years from discharge
YR9	C9	Value 9 years from discharge
YR10	C9	Value 10 years from discharge
YR15	C9	Value 15 years from discharge
YR16	C9	Value 16 years from discharge
YR18	C9	Value 18 years from discharge
YR20	C9	Value 20 years from discharge
YR25	C9	Value 25 years from discharge
YR30	C9	Value 30 years from discharge
YR40	C9	Value 40 years from discharge
YR50	C9	Value 50 years from discharge
YR60	C9	Value 60 years from discharge
YR70	C9	Value 70 years from discharge
YR80	C9	Value 80 years from discharge
YR90	C9	Value 90 years from discharge
YR100	C9	Value 100 years from discharge
YR200	C9	Value 200 years from discharge
YR300	C9	Value 300 years from discharge
YR350	C9	Value 350 years from discharge
YR400	C9	Value 400 years from discharge
YR500	C9	Value 500 years from discharge
YR1000	C9	Value 1,000 years from discharge
YR1050	C9	Value 1,050 years from discharge
YR2000	C9	Value 2,000 years from discharge

NEUTOTAL.DBF (continued)

FIELD NAME	TYPE/SIZE	DEFINITION
YR3000	C9	Value 3,000 years from discharge
YR5000	C9	Value 5,000 years from discharge
YR10K	C9	Value 10,000 years from discharge
YR30K	C9	Value 30,000 years from discharge
YR40K	C9	Value 40,000 years from discharge
YR50K	C9	Value 50,000 years from discharge
YR100K	C9	Value 100,000 years from discharge
YR200K	C9	Value 200,000 years from discharge
YR500K	C9	Value 500,000 years from discharge
YR1000K	C9	Value 1,000,000 years from discharge

INDEX NAME	INDEXED ON
(none)	

PHOTOTAL.DBF

Total number of photons at various decay times and burnups per metric ton of initial heavy metal.

FIELD NAME	TYPE/SIZE	DEFINITION
REACTOR	C10	'Pressurized' or 'Boiling' Water Reactor and the burnup bin
SPECTRUM	C10	Total number
YR0	C9	Value at discharge
YR1	C9	Value 1 year from discharge
YR2	C9	Value 2 years from discharge
YR3	C9	Value 3 years from discharge
YR4	C9	Value 4 years from discharge
YR5	C9	Value 5 years from discharge
YR6	C9	Value 6 years from discharge
YR7	C9	Value 7 years from discharge
YR8	C9	Value 8 years from discharge
YR9	C9	Value 9 years from discharge
YR10	C9	Value 10 years from discharge

PHOTOTAL.DBF (Continued)

FIELD NAME	TYPE/SIZE	DEFINITION
YR15	C9	Value 15 years from discharge
YR16	C9	Value 16 years from discharge
YR18	C9	Value 18 years from discharge
YR20	C9	Value 20 years from discharge
YR25	C9	Value 25 years from discharge
YR30	C9	Value 30 years from discharge
YR40	C9	Value 40 years from discharge
YR50	C9	Value 50 years from discharge
YR60	C9	Value 60 years from discharge
YR70	C9	Value 70 years from discharge
YR80	C9	Value 80 years from discharge
YR90	C9	Value 90 years from discharge
YR100	C9	Value 100 years from discharge
YR200	C9	Value 200 years from discharge
YR300	C9	Value 300 years from discharge
YR350	C9	Value 350 years from discharge
YR400	C9	Value 400 years from discharge
YR500	C9	Value 500 years from discharge
YR1000	C9	Value 1,000 years from discharge
YR1050	C9	Value 1,050 years from discharge
YR2000	C9	Value 2,000 years from discharge
YR3000	C9	Value 3,000 years from discharge
YR5000	C9	Value 5,000 years from discharge
YR10K	C9	Value 10,000 years from discharge
YR30K	C9	Value 30,000 years from discharge
YR40K	C9	Value 40,000 years from discharge
YR50K	C9	Value 50,000 years from discharge
YR100K	C9	Value 100,000 years from discharge
YR200K	C9	Value 200,000 years from discharge
YR500K	C9	Value 500,000 years from discharge
YR1000K	C9	Value 1,000,000 years from discharge

INDEX NAME	INDEXED ON
(none)	

POOL.DBF

Lookup table giving the name of each storage pool.

FIELD NAME	TYPE/SIZE	DEFINITION
POOL	C5	INIS storage pool identifier
NAME	C20	Name of the storage pool

INDEX NAME	INDEXED ON
POOL	POOL

RCTNAMES.DBF

Lookup table for reactor names.

FIELD NAME	TYPE/SIZE	DEFINITION
INISCODE	C4	INIS reactor identifier
REACTOR	C20	Reactor name
DOCKET	C6	Reactor identifier in Federal Docket
PWR_BWR	C1	'Pressurized' or 'Boiling' Water Reactor
ASSM_CLASS	C5	CDB Assembly Class
BWR_CLASS	C1	G.E. product line
FILESTHERE	L	T/F availability check for reactor & serial number files

INDEX NAME	INDEXED ON
RCTNMS	UPPER(REACTOR)

RCTR0101.DBF, RCTR0102.DBF, RCTR0103.DBF, etc.

Data relative to the batches of assemblies used at a specific reactor represented by an INIS_ID.

FIELD NAME	TYPE/SIZE	DEFINITION
INIS_ID	C4	INIS reactor identifier
BATCH	C6	Group of assemblies w/similar characteristics (Assembly Type, burnup, enrichment, etc.)
POOL_ID	C5	INIS storage pool identifier
ASSAY	C5	Initial enrichment of fuel
NUMBER	C3	Number of assemblies in batch
WEIGHT	C9	Average initial weight of heavy metal (grams)
RODARRAY	C5	EIA Assembly Type code
DIS_DATE	C4	Year of final discharge
BURNUP	C5	Final burnup of assembly
CYCLE	C3	Last cycle of irradiation
BURNUP_BIN	C2	Grouping by burnup range

INDEX NAME	INDEXED ON
BA_BURN	BATCH+BURNUP_BIN
BA_RCTR	BATCH
BA_RDDRY	BATCH+RODARRAY

S0101.DBF, S0102.DBF, S0102.DBF, etc.

This file contains the assembly serial numbers and individual assembly weights and burnups.

FIELD NAME	TYPE/SIZE	DEFINITION
INIS_ID	C4	INIS reactor identifier
ASSERNO	C14	Assembly serial number (vendor)
SERIAL	C14	Assembly serial number (ANSI)
BATID	C14	EIA Batch identifier (Group of assemblies w/similar characteristics)
POOLID	C4	INIS storage pool identifier
INILOD	N6.0	Initial heavy metal (grams)

INDEX NAME	INDEXED ON
ANSI	SERIAL
VENDOR	ASSERNO

TOTALS.DBF

Radiological totals for a particular set of conditions (reactor type+burnup+decay time).

FIELD NAME	TYPE/SIZE	DEFINITION
REACTOR	C3	'Pressurized' or 'Boiling' Water Reactor
BURNUP	C6	Final burnup exposure for this case
DECAY	C7	Length of time since discharge
CURIES	C8	Total curies of radioactive materials from 1 metric ton of initial heavy metal of fuel
WATTS	C8	Total watts of radioactive materials from 1 metric ton of initial heavy metal of fuel
GRAMS	C8	Total grams of all materials from 1 metric ton of initial heavy metal of fuel

INDEX NAME	INDEXED ON
TOTALST	PWR_BWR+DECAY
TOTALSB	PWR_BWR+BURNUP
TOTALS	PWR_BWR+DECAY+BURNUP

5.0 PROGRAMS

5.1 SOURCE

The names of the system programs with brief descriptions of how they work follows:

1. ASSEMBLY.PRG - Prompts user to select a report by burnup bin or assembly type and then select the desired batch number.

   After user has selected the assembly type (snassm.prg),the routine prompts the user to select a batch number (snbatch.prg). The serial numbers which match those requirements are listed for the user to pick from (srl_nmbr.prg) and a report is produced (snreport.prg).

2. BATCH.PRG - Prompts user to select a batch and serial number for the report.

   After the batch is selected (snbatch.prg), the serial numbers which match those requirements are listed for the user to pick from (srlnmbr.prg), and the report is produced (snreport.prg).

3. BURNUP.PRG - Prompts user to select a report by burnup bin and select the desired batch number.

   After user has selected a burnup bin (snburn.prg), the routine prompts the user to select a batch number (snbatch.prg). The serial numbers which match those requirements are listed for the user to pick from (srl_nmbr.prg) and a report is produced (snreport.prg).

4. CALC.PRG - Interpolation routine to calculate curies orwatts from stored values based on burnup and decay time. The formula was obtained from Appendix 1C of DOE/RW-0184.

5. CALC1.PRG - Interpolation routine to calculate photon spectra from stored values based on burnup and decay time. The photon formula was obtained from Appendix 1C of DOE/RW-0184.

6. CALC2.PRG - Interpolation routine to calculate number of (alpha,n), fission, and total neutrons from stored values based on burnup and decay time. The neutron formula was obtained from Appendix 1C of DOE/RW-0184.

7. CONVSCI.PRG - Converts a decimal number to a mantissa and exponent and returns it as a character string.

8. EXISTS.PRG - Checks to see if a selected output file already exists. If it does, the routine checks that the user wants to overwrite it.

9. FILTERS.PRG - Establishes a filter for srl_nmbr.prg if a batch number, an assembly type, or a burnup bin have been chosen so only the appropriate serial numbers will print.

10. GETTIME.PRG - Prompts user to select a specific decay time for report if time other than current time is desired.

    The current time is displayed for use by the calculations of radiological values. This decay time is used by the interpolation routines for curies, watts, photons, and neutrons and may be changed here to alter the calculations.

11. INTERPOL.PRG - Performs interpolation calculations for curies, watts, photon values, or neutron values using two time and two burnup values. The interpolation formulas were obtained from Appendix 1C of DOE/RW-0184.

12. PRNT.PRG - Prompts user to decide whether to send the report to the screen, the printer, or a file. If the user selects a file, the routine will prompt for a file name.

13. RCTRNAME.PRG - Allows user to enter reactor names for report.

    The program lists all available reactor names upon entry to screen. User enters a reactor site name, number or if name is unknown, enters a few letters of name for a filtered list. If the program is run repetitively, the F10 key saves the reactor site name entered previously.

14. RTRN.PRG - Prompts user with a "Hit any key" message before going on to next screen.

15. SERIAL.PRG - Prompts user to select either an ANSI- or Vendor-type serial number for report.

16. SNASSM.PRG - Prompts user to select an assembly type for a report.

    The available assembly types are displayed and the user makes a selection from the list.

17. SNBATCH.PRG - User can select from a list of available batch numbers or enter an actual batch number. The batch numbers are filtered by previous selections of burnup bin or assembly type.

18. SNBURN.PRG - Prompts user to select a burnup bin for a report.

    The available burnup bins are displayed and the user makes a selection from the list.

19. SNCDB.PRG - Presents the initial screen of information for a user accessing the Serial Number data base programs.

20. SNDB.PRG - Main menu for the Serial Number Data Base routines.

    The user is presented with a menu suggesting a report by serial number, batch, assembly type, burnup bin, or exiting the database program. The program loops through this routine repeatedly until the user exits, prompting for selection of a reactor name each time.

21. SNINIT.PRG - Initializes variables for the Serial Number Data Base routines.

22. SNREPORT.PRG - Produces the report using the parameters the user has selected from previous routines.

    Routine checks for the serial number which was selected. If it's not found the program starts over; otherwise it prompts for where to send the report, i.e. to a file, the printer, or the screen and cycles through once for the screen, twice for a file or the printer. Once the headers are printed, the routine prompts for a change in the time frame (gettime.prg). Once a time frame is
    established, the interpolation calculations are performed for curies, watts, photons and neutrons (calc.prg, calc1.prg, calc2.prg, interpol.prg, convsci.prg) and the rest of the report prints out.

23. SRL_NMBR.PRG - User can select from the list of available serial numbers or enter an actual serial number. The serial numbers are filtered by previous selections of batch, burnup bin, or assembly type.

24. STARTUP.PRG - Determines column numbers and increments for position of data printing in report.

5.2 INSTALLATION

INSTALL.BAT is included with the system files and programs on the floppies. To install the system, the user places the first diskette in a floppy drive and enters the following command: INSTALL C where "C" is the letter of the hard disk where the system should be installed. The batch file will accept either "C" or "C:" with checking done by the routine CHKARG (from K.E. Jones of Dataphile). Available space on the hard drive is also checked by the routine DISKSP.EXE (from K.E. Jones of Dataphile). If there is not enough available space, installation will stop.

```
1  *=====================================================================
2  *  Program:  assembly.prg
3  *  Purpose:  Selects type of serial number report by assembly, then batch.
4  *  Notes:    Called by sndb.prg.
5  *  Date:     03/89
6  *  Author:   B. Lewis
7  *=====================================================================
8  serialno="S"+minis
9  *
10 *  First, pick out an assembly type:
11 *
12 DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
13 *
14    DO snassm
15 *
16 *  Now pick out a batch number:
17 *
18    IF .NOT. rtrnprev .AND. .NOT. logoff
19       DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
20          DO snbatch
21 *
22 *  Pick out a serial number for the report:
23 *
24          IF .NOT. rtrnprev .AND. .NOT. logoff
25 *
26             DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
27                SELECT 9
28                USE &serialno INDEX vendor,ansi
29 *
30                SET FILTER TO minis+batch=key
31                GO TOP
32 *
33                DO sri_nmbr
34                SET FILTER TO
35 *
36 *  Now, let's print the report:
37 *
38                IF .NOT. rtrnprev .AND. .NOT. logoff
39                   DO snreport
40                ENDIF
41 *
42             ENDDO
43             rtrnprev=.F.
44 *
45          ENDIF
46       ENDDO
47       rtrnprev=.F.
48 *
49    ENDIF
50 ENDDO
51 *
52 RETURN
```

```
1  *=====================================================================
2  *  Program:  batch.prg
3  *  Purpose:  Selects type of serial number report by batch.
4  *  Notes:    Called by sndb.prg.
5  *  Date:     03/89
6  *  Author:   B. Lewis
7  *=====================================================================
8  serialno="S"+minis
9  *
10 *  Get a batch number for the report:
11 *
12 DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
13 *
14    DO snbatch
15 *
16 *  Pick out a serial number for the report:
17 *
18    IF .NOT. rtrnprev .AND. .NOT. logoff
19 *
20       DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
21          SELECT 9
22          USE &serialno INDEX vendor,ansi
23 *
24          SET FILTER TO  minis+batch=key
25          GO TOP
26 *
27          DO srl_nmbr
28          SET FILTER TO
29 *
30 *  Now, let's print the report:
31 *
32          IF .NOT. rtrnprev .AND. .NOT. logoff
33             DO snreport
34          ENDIF
35 *
36       ENDDO
37       rtrnprev=.F.
38 *
39    ENDIF
40 *
41 ENDDO
42 *
43 RETURN
44
```

```
1  *=====================================================================
2  *  Program:  burnup.prg
3  *  Purpose:  Selects type of serial number report by burnup bin, then batch.
4  *  Notes:    Called by sndb.prg.
5  *  Date:     03/89
6  *  Author:   B. Lewis
7  *=====================================================================
8  serialno="S"+minis
9  *
10 *  First, pick out a burnup bin:
11 *
12 DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
13 *
14    DO snburn
15 *
16 *  Now pick out a batch number:
17 *
18    IF .NOT. rtrnprev .AND. .NOT. logoff
19       DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
20          DO snbatch
21 *
22 *  Pick out a serial number for the report:
23 *
24          IF .NOT. rtrnprev .AND. .NOT. logoff
25 *
26             DO WHILE .NOT. rtrnprev .AND. .NOT. logoff
27                SELECT 9
28                USE &serialno INDEX vendor,ansi
29 *
30                SET FILTER TO minis+batch=key
31                GO TOP
32 *
33                DO srl_nmbr
34                SET FILTER TO
35 *
36 *  Now, let's print the report:
37 *
38                IF .NOT. rtrnprev .AND. .NOT. logoff
39                   DO snreport
40                ENDIF
41 *
42             ENDDO
43             rtrnprev=.F.
44 *
45          ENDIF
46       ENDDO
47       rtrnprev=.F.
48 *
49    ENDIF
50 ENDDO
51 *
52 RETURN
```

```
1  *==========================================================================
2  *  Program:   calc.prg
3  *  Purpose:   Calculates curies or watts based on decay time in years and
4  *             burnup.  (Formulas obtained from Appendix 1c of DOE/RW-0184.)
5  *             This is Barry McCloud's interpolation function.  Note that
6  *             interpol.prg does the actual calculation.
7  *  Notes:     Called by snreport.prg
8  *  Date:      04/89
9  *  Author:    B. Lewis
10 *==========================================================================
11 PARAMETERS formula,t,b,h,p_or_b,string
12 ******************************************************************
13 *  formula - tells program to look for curies, watts, or neutrons
14 *  t - decay time to search for
15 *  t0 - time variable for calculations.  Is <= t.
16 *  t1 - time variable for calculations.  Is >= t.
17 *  b - burnup to search for
18 *  b0 - burnup variable for calculations.  Is <= b.
19 *  b1 - burnup variable for calculations.  Is >= b.
20 *  h - weight of assembly
21 *  p_or_b - Look at PWR or BWR data?
22 *  string - return value with answer
23 ******************************************************************
24 *
25 SET DECIMALS TO 10
26 SELECT 1
27 USE totals INDEX totalst,totalsb,totals
28 *
29 string=SPACE(10)
30 thyme=STR(t,7)
31 burn=STR(b,6)
32 bhold=b
33 thold=t
34 STORE 0 TO b0,b1,t0,t1,c_t0b0,c_t0b1,c_t1b0,c_t1b1,c_tb,c_t0b,c_t1b
35 *
36 *  Find times that bracket actual time
37 *
38 SET ORDER TO 1
39 SEEK p_or_b+thyme
40 *
41 IF FOUND()
42    t0=VAL(decay)
43    t1=VAL(decay)
44 *
45 ELSE
46    GO TOP
47    LOCATE FOR decay > thyme .AND. pwr_bwr=p_or_b
48 *
49    t1=VAL(decay)
50    SKIP-1
51    t0=VAL(decay)
52 ENDIF
53 *
54 *  Find burnups that bracket actual burnup
55 *
```

```
SET ORDER TO 2
SEEK p_or_b+burn
*
IF FOUND()
   b0=VAL(burnup)
   b1=VAL(burnup)
*
ELSE
   GO TOP
   LOCATE FOR burnup > burn .AND. pwr_bwr=p_or_b
*
   b1=VAL(burnup)
   SKIP-1
   b0=VAL(burnup)
ENDIF
*
*  Find values for all combinations
*
   SET ORDER TO 3
*
   hold=STR(t0,7)+STR(b0,6)
   SEEK p_or_b+hold
*
   DO CASE
      CASE formula="C"
           value=curies
      CASE formula="W"
           value=watts
   ENDCASE
*
   p=VAL(SUBSTR(value,6,3))
   c_t0b0=VAL(value) * (10 ** p)
*
   hold=STR(t0,7)+STR(b1,6)
   SEEK p_or_b+hold
*
   DO CASE
      CASE formula="C"
           value=curies
      CASE formula="W"
           value=watts
   ENDCASE
*
   p=VAL(SUBSTR(value,6,3))
   c_t0b1=VAL(value) * (10 ** p)
*
   hold=STR(t1,7)+STR(b0,6)
   SEEK p_or_b+hold
*
   DO CASE
      CASE formula="C"
           value=curies
      CASE formula="W"
           value=watts
   ENDCASE
```

```
111 *
112    p=VAL(SUBSTR(value,6,3))
113    c_t1b0=VAL(value) * (10 ** p)
114 *
115    hold=STR(t1,7)+STR(b1,6)
116    SEEK p_or_b+hold
117 *
118    DO CASE
119       CASE formula="C"
120            value=curies
121       CASE formula="W"
122            value=watts
123    ENDCASE
124 *
125    p=VAL(SUBSTR(value,6,3))
126    c_t1b1=VAL(value) * (10 ** p)
127 *
128 *  Do interpolations:
129 *
130    DO interpol WITH thold,bhold,b0,b1,t0,t1,c_t0b0,c_t0b1,c_t1b1,c_t1b0,c_tb
131 *
132 *  Convert answer to per assembly by multiplying by the weight in grams:
133 *
134    c_tb=c_tb * (h / 1000000)
135 *
136 *  Build the string which will return the answer:
137 *
138    DO convsci WITH c_tb,string
139 *
140 USE
141 *
142 SET DECIMALS TO 4
143 RETURN
```

```
*==========================================================================
* Program:   calc1.prg
* Purpose:   Calculates photons based on decay time in years and
*            burnup.  (Formulas obtained from Appendix 1c of DOE/RW-0184.)
*            Note that interpol.prg does actual calculation.
* Notes:     Called by snreport.prg
* Date:      06/89
* Author:    B. Lewis
*==========================================================================
PARAMETERS formula,t,b,h,p_or_b,string
*****************************************************************
* formula - tells program to look for curies, watts, or neutrons
* t - decay time to search for
* t0 - time variable for calculations.  Is <= t.
* t1 - time variable for calculations.  Is >= t.
* b - burnup to search for
* b0 - burnup variable for calculations.  Is <= b.
* b1 - burnup variable for calculations.  Is >= b.
* h - weight of assembly
* p_or_b - Look at PWR or BWR data?
* string - return value with answer
*****************************************************************
*
SET DECIMALS TO 10
*
string=SPACE(10)
burn=STR(b,6)
bhold=b
thold=t
STORE 0 TO b0,b1,t0,t1,c_t0b0,c_t0b1,c_t1b0,c_t1b1,c_tb,c_t0b,c_t1b
*
* Get the time variable names based on the decay time value by
* searching for a match or years that bracket the given year:
*
SELECT 1
USE yrfile INDEX yrfile
*
SEEK t
IF FOUND()
   t0=t
   t1=t
   timevar1=yr
   timevar0=yr
ELSE
   GO TOP
   LOCATE FOR year > t
   t1=year
   timevar1=yr
*
   SKIP-1
   t0=year
   timevar0=yr
ENDIF
*
* Set a filter so we're looking at either PWR or BWR data:
```

```
*
SELECT 1
USE phototal INDEX phototal
*
*  Find burnups that bracket actual burnup
*
hold=SUBSTR(burn,1,3)+","+SUBSTR(burn,4,3)
SEEK p_or_b+hold
*
IF FOUND()
   burnrec0=RECNO()
   burnrec1=RECNO()
   b0=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
   b1=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
*
ELSE
   GO TOP
   LOCATE FOR VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3)) > b .AND. ;
              p_or_b=SUBSTR(reactor,1,3)
*
   burnrec1=RECNO()
   b1=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
   SKIP-1
   burnrec0=RECNO()
   b0=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
ENDIF
*
*  Find values for all combinations
*
   GOTO burnrec0
   p=VAL(SUBSTR(&timevar0,7,3))
   c_t0b0=VAL(&timevar0) * (10 ** p)   && time0 & burnup0
*
   p=VAL(SUBSTR(&timevar1,7,3))
   c_t1b0=VAL(&timevar1) * (10 ** p)       && time1 & burnup0
*
   GOTO burnrec1
   p=VAL(SUBSTR(&timevar0,7,3))
   c_t0b1=VAL(&timevar0) * (10 ** p)       && time0 & burnup1
*
   p=VAL(SUBSTR(&timevar1,7,3))
   c_t1b1=VAL(&timevar1) * (10 ** p)       && time1 & burnup1
*
*  Do interpolations:
*
   DO interpol WITH thold,bhold,b0,b1,t0,t1,c_t0b0,c_t0b1,c_t1b1,c_t1b0,c_tb
*
*  Convert answer to per assembly by multiplying by the weight in grams:
*
   c_tb=c_tb * (h / 1000000)
*
*  Build the string which will return the answer:
*
   DO convsci WITH c_tb,string
*
```

```
111 USE
112 SET DECIMALS TO 4
113 RETURN
```

```
1  *==========================================================================
2  *  Program:   calc2.prg
3  *  Purpose:   Calculates neutrons for (alpha,n), fission, and totals based
4  *             on decay time in years and burnup.
5  *             (Formulas obtained from Appendix 1c of DOE/RW-0184.)
6  *             Note that interpol.prg does actual calculation.
7  *  Notes:     Called by snreport.prg
8  *  Date:      06/89
9  *  Author:    B. Lewis
10 *==========================================================================
11 PARAMETERS formula,t,b,h,p_or_b,string
12 ***************************************************************
13 *  formula - tells program to look for curies, watts, or, if neutrons
14 *            for alpha, fission, or total data.
15 *  t - decay time to search for
16 *  t0 - time variable for calculations.  Is <= t.
17 *  t1 - time variable for calculations.  Is >= t.
18 *  b - burnup to search for
19 *  b0 - burnup variable for calculations.  Is <= b.
20 *  b1 - burnup variable for calculations.  Is >= b.
21 *  h - weight of assembly
22 *  p_or_b - Look at PWR or BWR data?
23 *  string - return value with answer
24 ***************************************************************
25 *
26 SET DECIMALS TO 10
27 *
28 string=SPACE(10)
29 burn=STR(b,6)
30 bhold=b
31 thold=t
32 STORE 0 TO b0,b1,t0,t1,c_t0b0,c_t0b1,c_t1b0,c_t1b1,c_tb,c_t0b,c_t1b
33 *
34 *  Get the time variable names based on the decay time value by
35 *  searching for a match or years that bracket the given year:
36 *
37 SELECT 1
38 USE yrfile INDEX yrfile
39 *
40 SEEK t
41 IF FOUND()
42    t0=t
43    t1=t
44    timevar1=yr
45    timevar0=yr
46 ELSE
47    GO TOP
48    LOCATE FOR year > t
49    t1=year
50    timevar1=yr
51 *
52    SKIP-1
53    t0=year
54    timevar0=yr
55 ENDIF
```

```
*
*  Set a filter so we're looking at either PWR or BWR data and
*  either alpha, fission, or total neutrons:
*
SELECT 1
USE neutotal INDEX neutotal
*
*  Find burnups that bracket actual burnup
*
hold=SUBSTR(burn,1,3)+","+SUBSTR(burn,4,3)
SEEK p_or_b+hold+formula
*
IF FOUND()
   burnrec0=RECNO()
   burnrec1=RECNO()
   b0=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
   b1=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
*
ELSE
   GO TOP
   LOCATE FOR VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3)) > b .AND. ;
              p_or_b=SUBSTR(reactor,1,3) .AND. formula=source
*
   burnrec1=RECNO()
   b1=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
   SKIP-1
   burnrec0=RECNO()
   b0=VAL(SUBSTR(reactor,4,3) + SUBSTR(reactor,8,3))
ENDIF
*
*  Find values for all combinations
*
   GOTO burnrec0
   p=VAL(SUBSTR(&timevar0,7,3))
   c_t0b0=VAL(&timevar0) * (10 ** p)   && time0 & burnup0
*
   p=VAL(SUBSTR(&timevar1,7,3))
   c_t1b0=VAL(&timevar1) * (10 ** p)       && time1 & burnup0
*
   GOTO burnrec1
   p=VAL(SUBSTR(&timevar0,7,3))
   c_t0b1=VAL(&timevar0) * (10 ** p)       && time0 & burnup1
*
   p=VAL(SUBSTR(&timevar1,7,3))
   c_t1b1=VAL(&timevar1) * (10 ** p)       && time1 & burnup1
*
*  Do interpolations:
*
   DO interpol WITH thold,bhold,b0,b1,t0,t1,c_t0b0,c_t0b1,c_t1b1,c_t1b0,c_tb
*
*  Convert answer to per assembly by multiplying by the weight in grams:
*
   c_tb=c_tb * (h / 1000000)
*
*  Build the string which will return the answer:
```

```
111 *
112    DO convsci WITH c_tb,string
113 *
114 USE
115 SET DECIMALS TO 4
116 RETURN
```

```
 1  *=====================================================================
 2  *  Program:  convsci.prg
 3  *  Purpose:  Takes an incoming number and converts it to a mantissa
 4  *            and exponent (scientific notation).  Value returned is
 5  *            a character string.
 6  *  Date:     04/89
 7  *  Author:   B. Lewis
 8  *=====================================================================
 9  PARAMETERS number,newnumber
10  *
11  mantissa=number
12  exponent=0
13  *=====================================================================
14  *  Determine the exponent & fractional value of the number:
15  *
16  DO WHILE number >= 10.0 .OR. number < 1.0
17  *
18     DO CASE
19        CASE number >= 10.0
20             number = number / 10.0
21             exponent = exponent + 1
22  *
23        CASE number < 1.0
24             number = number * 10.0
25             exponent = exponent - 1
26     ENDCASE
27  *
28  ENDDO
29  *=====================================================================
30  *  Let's add the "E" and turn everything into a string.
31  *
32  IF exponent >= 0
33     newnumber = "E+"
34  *
35  ELSE
36     newnumber = "E-"
37  ENDIF
38  *
39  SET DECIMALS TO 2
40  newnumber = STR(number,4,2) + newnumber
41  SET DECIMALS TO 5
42  *=====================================================================
43  *  Place a zero in the string with the exponent if it's less than 10:
44  *
45  IF exponent < 10
46     newnumber = newnumber + "0" + STR(ABS(exponent),1)
47  ELSE
48     newnumber = newnumber + STR(ABS(exponent),2)
49  ENDIF
50  *
51  RETURN
```

```
 1 *==============================================================
 2 *  Program:  filters.prg
 3 *  Purpose:  Establishes a filter string for the listing of serial
 4 *            numbers if batch, burnup bin, or assembly types have
 5 *            already been chosen.
 6 *  Notes:    Called by srl_nmbr.prg.
 7 *  Date:     12/89
 8 *  Author:   B. Lewis
 9 *==============================================================
10 filterstring=SPACE(1)
11 yes=.F.
12 *--------------------------------------------------------------
13 *  Establish a filter by batch number:
14 *
15 IF (EMPTY(assm) .AND. EMPTY(bin)) .AND. .NOT. EMPTY(key)
16    filterstring="batch='&key'"
17    yes=.T.
18 ENDIF
19 *--------------------------------------------------------------
20 *  Establish a filter by rod array and batch number:
21 *
22 IF .NOT. EMPTY(assm)
23    filterstring="rodarray='&assm' .AND. batch='&key'"
24    yes=.T.
25 ENDIF
26 *--------------------------------------------------------------
27 *  Establish a filter by burnup bin and batch number:
28 *
29 IF .NOT. EMPTY(bin)
30 DO CASE
31    CASE "1" $ bin
32         filterstring="batch='&key' .AND. maxburn>=1 .AND. maxburn<=5000"
33 *
34    CASE "2" $ bin
35         filterstring="batch='&key' .AND. maxburn>=5001 .AND. maxburn<=10000"
36 *
37    CASE "3" $ bin
38         filterstring="batch='&key' .AND. maxburn>=10001 .AND. maxburn<=15000"
39 *
40    CASE "4" $ bin
41         filterstring="batch='&key' .AND. maxburn>=15001 .AND. maxburn<=20000"
42 *
43    CASE "5" $ bin
44         filterstring="batch='&key' .AND. maxburn>=20001 .AND. maxburn<=25000"
45 *
46    CASE "6" $ bin
47         filterstring="batch='&key' .AND. maxburn>=25001 .AND. maxburn<=30000"
48 *
49    CASE "7" $ bin
50         filterstring="batch='&key' .AND. maxburn>=30001 .AND. maxburn<=35000"
51 *
52    CASE "8" $ bin
53         filterstring="batch='&key' .AND. maxburn>=35001 .AND. maxburn<=40000"
54 *
55    CASE "9" $ bin
```

```
56          filterstring="batch='&key' .AND. maxburn>=40001 .AND. maxburn<=45000"
57  *
58     CASE "10" $ bin
59          filterstring="batch='&key' .AND. maxburn>=45001 .AND. maxburn<=50000"
60  *
61     CASE "11" $ bin
62          filterstring="batch='&key' .AND. maxburn>=50001 .AND. maxburn<=55000"
63  *
64     CASE "12" $ bin
65          filterstring="batch='&key' .AND. maxburn>=55001 .AND. maxburn<=60000"
66  *
67     CASE "13" $ bin
68          filterstring="batch='&key' .AND. maxburn<60001"
69  *
70  ENDCASE
71  *
72     yes=.T.
73  ENDIF
74  *
75  IF yes
76    SET FILTER TO &filterstring
77  ENDIF
78  *
79  RETURN
```

```
1  *==========================================================================
2  *  PROCEDURE:  gettime.prg
3  *  PURPOSE:    Calculates a time frame for the CURIES calculations.
4  *  DATE:       04/89
5  *  AUTHOR:     B. Lewis
6  *==========================================================================
7  *  Set up a round-up factor if month>6:
8  *
9  years=0
10 IF MONTH(DATE()) > 6
11    years=1
12 ENDIF
13 *==========================================================================
14 *  Determine # of elapsed years since date of discharge:
15 *
16 years = years + YEAR(DATE()) - discharged
17 *==========================================================================
18 *  Display current values based on system time (and user's time):
19 *
20 DO WHILE .T.
21 @ 18,2 SAY "Current Time: " + ;
22            ALLTRIM(STR(MONTH(DATE()))) + "/" + STR(YEAR(DATE()),4)
23 *
24 @ 19,2 SAY "Time Frame:  " + ALLTRIM(STR(years,4)) + " years     "
25 *==========================================================================
26 *  Does user want to use a different time frame?
27 *
28 answer="N"
29 @ 17,33 TO 21,78 DOUBLE
30 @ 18,35 SAY "This time frame will be used to calculate"
31 @ 19,35 SAY "the radiological values.  Do you want to"
32 @ 20,35 SAY "change the time frame (Y/N)? - "
33 @ 20,COL() GET answer PICTURE "Y"
34 READ
35 *==========================================================================
36 *  Get new value for elapsed time since discharge:
37 *
38 IF answer="Y"
39    @ 19,15 GET years PICTURE "9999"
40    READ
41 *
42 ELSE
43    EXIT
44 ENDIF
45 *
46 ENDDO
47 *
48 @ 17,1 CLEAR TO 21,79
49 *
50 RETURN
```

```
1  *==========================================================================
2  *  Program:   interpol.prg
3  *  Purpose:   Does interpolation with two time and two burnup values.
4  *             (Formulas obtained from Appendix 1c of DOE/RW-0184.)
5  *  Notes:     Called by calc.prg, calc1.prg, and calc2.prg (from
6  *             snreport.prg).
7  *  Date:      04/89
8  *  Author:    B. Lewis
9  *==========================================================================
10 PARAMETERS thold,bhold,b0,b1,t0,t1,c_t0b0,c_t0b1,c_t1b1,c_t1b0,c_tb
11 ******************************************************************
12 *
13 SET DECIMALS TO 10
14 *
15 *  Do 1st of 3 interpolations:
16 *
17    IF c_t0b0=0 .OR. b0=0
18       c_t0b = 0
19    ELSE
20 *
21       IF b1=b0
22          p=0
23       ELSE
24          p = (LOG(c_t0b1 / c_t0b0)) / (LOG(b1 / b0))
25       ENDIF
26       c_t0b = c_t0b0 * ((bhold / b0) ** p)
27       ENDIF
28 *
29 *  Do 2nd of 3 interpolations:
30 *
31    IF c_t1b0=0 .OR. b0=0
32       c_t1b=0
33    ELSE
34 *
35       IF b1=b0
36          p=0
37       ELSE
38          p = (LOG(c_t1b1 / c_t1b0)) / (LOG(b1 / b0))
39       ENDIF
40       c_t1b = c_t1b0 * ((bhold / b0) ** p)
41    ENDIF
42 *
43 *  And finally, the 3rd:
44 *
45    IF c_t1b=0 .OR. t0=0
46       c_tb=0
47    ELSE
48 *
49       IF t1=t0
50          m=0
51       ELSE
52          m = (LOG( c_t0b / c_t1b)) / (LOG(t1 / t0))
53       ENDIF
54       c_tb = c_t0b * ((thold / t0 ) ** (-1 * m))
55    ENDIF
```

```
*
SET DECIMALS TO 4
RETURN
```

```
1  *==========================================================================
2  *  Program:  prnt.prg
3  *  Purpose:  Checks to see if report should go to screen, printer or a file.
4  *  Note:     Called by snreport.prg.
5  *  Date:     06/89
6  *  Author:   B. Lewis
7  *==========================================================================
8  PARAMETERS prntwhere,outfile
9  *
10 printwhere=" "
11 outfile="OUTFILE"
12 *
13 *  Get user's selection for output method:
14 *
15 CLEAR
16 @ 10,5 TO 13,60 DOUBLE
17 DO WHILE .NOT. printwhere $ "SPFZ"
18    printwhere=" "
19    @ 11,10 SAY "Send report to screen, printer, or"
20    @ 12,10 SAY "file (S/P/F)? - " ;
21            GET printwhere PICTURE "!"
22    READ
23 ENDDO
24 *
25 IF printwhere="Z"
26    QUIT
27 ENDIF
28 *
29 *  Send output to a file (default file is OUTFILE.PRN)?
30 *
31 IF printwhere="F"
32         number=MEMORY(0)
33         CLEAR
34         @ 11,10 CLEAR TO 11,59
35         @ 12,10 CLEAR TO 12,59
36         @ 11,10 SAY "Printout will go to file OUTFILE.PRN "
37 *
38         ans="N"
39         @ 12,10 SAY "Change the file name (Y/N)? - " ;
40                 GET ans PICTURE "Y"
41         READ
42 *
43 *  If user isn't changing the file name, see if they've output
44 *  to OUFILE before and if so, make sure they want to overwrite:
45 *
46         IF ans="N"
47            newfile=.F.
48            DO exists WITH outfile,newfile
49 *
50            IF newfile                  && change the output file:
51               DO newone WITH outfile
52            ENDIF
53 *
54 *  Get a new file name for output:
55 *
```

```
56          ELSE
57             DO newone WITH outfile
58          ENDIF
59  *
60  ENDIF         && output to a file??
61  *
62  CLEAR
63  RETURN
64
65  PROCEDURE exists
66  *
67  *  Routine will check if file name already exists.  If it does, then
68  *  whether or not to overwrite it
69  *
70  PARAMETER outfile,newfile
71  *
72                IF FILE(ALLTRIM(outfile) + ".PRN")
73                   overwrite="N"
74                   @ 11,10 CLEAR TO 12,59
75                   @ 11,10 SAY "File name already exists, overwrite (Y/N)? - " ;
76                           GET overwrite PICTURE "Y"
77                   READ
78  *
79  *  IF user wants to overwrite his existing file, don't get another name:
80  *
81                   IF overwrite="N"
82                      newfile=.F.
83                   ELSE
84                      newfile=.T.
85                   ENDIF
86  *
87                ELSE
88  *
89  *  Before accepting the new file name, make sure the user really entered
90  *  something other than a blank name:
91  *
92                   IF EMPTY(outfile)
93                      newfile=.T.
94                   ELSE
95                      newfile=.F.
96                   ENDIF
97  *
98                ENDIF
99  *
100 RETURN
101
102 PROCEDURE newone
103 *
104 *  Routine to get a new file name for report output
105 *
106 PARAMETERS outfile
107 *
108            newfile=.T.
109            DO WHILE newfile
110               @ 11,10 CLEAR TO 12,59
```

```
111                 @ 12,10 SAY "(File name may be up to 8 characters long.)"
112  *
113                 outfile=SPACE(8)
114                 @ 11,10 SAY "Enter new file name- " GET outfile PICTURE "!!!!!!!!"
115                 READ
116  *
117  *  Check to see if new file name exists; if so, check if it's okay to
118  *  overwrite it; if not, get another file name.
119  *
120                 DO exists WITH outfile,newfile
121              ENDDO
122  *
123  RETURN
```

```
 1  *=========================================================================
 2  *  Program:  rctrname.prg
 3  *  Purpose:  Gets reactor name/site user wants to report on.  Will give
 4  *            list of available names.
 5  *  Notes:    Called by sndb.prg.
 6  *  Date:     03/89
 7  *  Author:   B. Lewis
 8  *=========================================================================
 9  SELECT 9
10  USE rctnames INDEX rctnms
11  *
12  SET FUNCTION 10 TO holdname
13  *
14  DECLARE recordno[200]
15  filterset=.F.
16  line=4
17  count=0
18  col1=15
19  col2=45
20  column=col1
21  *
22  DO WHILE .NOT. logoff
23     CLEAR
24     @ 1,1 TO 20,79 DOUBLE
25     @ 2,15 SAY "S E L E C T   R E A C T O R   S I T E   N A M E:"
26     @ 3,2 TO 3,78 DOUBLE
27     @ 18,74 SAY "SN04"
28  *-----------------------------------------------
29  *  List the reactor names:
30  *
31     column=col1
32     line=4
33     DO WHILE .T.
34        IF .NOT. EOF()
35           IF line=17
36  *
37              IF column=col1
38                 column=col2
39                 line=4
40  *
41              ELSE
42                 EXIT
43              ENDIF
44           ENDIF
45  *
46        ELSE
47           SET FILTER TO
48           EXIT
49        ENDIF
50  *
51        IF filesthere
52           line=line+1
53           count=count+1
54           recordno[count]=RECNO()
55           @ line,column SAY STR(count,3) + "-" + ALLTRIM(reactor)
```

```
56          ENDIF
57   *
58          SKIP
59       ENDDO
60   *
61       IF count > 1
62          string = "Enter reactor site name, 1-"+ALLTRIM(STR(count))+ ;
63                   ", or X to Exit- "
64          length=LEN(string)
65          centered=40 - length/2
66       ELSE
67          string = "Enter the reactor site name, 1, or X to Exit- "
68          length=LEN(string)
69          centered=40 - length/2
70       ENDIF
71   *
72       holdcol1=centered-11
73       holdcol2=centered+length+10
74       @ 19,centered-12 CLEAR TO 21,centered+length+11
75       @ 19,centered-12 TO 21,centered+length+11 DOUBLE
76   *
77       @ 20,centered-10 SAY string
78   *
79       IF .NOT. EOF()
80          string="< <F10> for previous selection OR <ENTER> for more selections>"
81          @ 22,40-LEN(string)/2 SAY string
82   *
83       ELSE
84          string="< <F10> for previous selection>"
85          @ 22,40-LEN(string)/2 SAY string
86       ENDIF
87   *-----------------------------------------------
88   *  Get user's selection:
89   *
90       sn04_choice=SPACE(20)
91       @ 20,centered+length-10 GET sn04_choice PICTURE "!!!!!!!!!!!!!!!!!!!!"
92       READ
93   *-----------------------------------------------
94   *  Act on user's selection:
95   *
96       reactrname=SPACE(20)
97       DO CASE
98          CASE EMPTY(sn04_choice)                  && nothing entered??
99               IF EOF()
100                 count=0
101                 GO TOP
102              ENDIF
103  *
104         CASE ALLTRIM(sn04_choice)="X"            && ready to exit??
105              logoff=.T.
106              EXIT
107  *
108         CASE VAL(sn04_choice) >= 1 .AND. ;
109              VAL(sn04_choice) <= count           && number entered??
110              GOTO recordno[VAL(sn04_choice)]
```

```
111            minis=unitid
112            reactrname=reactor
113            EXIT
114  *
115        OTHERWISE                          && reactor name entered??
116               SEEK sn04_choice
117  *-----------------------------------------------------
118  *  If value is unavailable, see if we can locate anything
119  *  that matches the partial string the user gave us:
120  *
121            IF .NOT. FOUND()                && no match on user's entry
122               length=LEN(ALLTRIM(sn04_choice))
123               LOCATE FOR ALLTRIM(sn04_choice)=UPPER(SUBSTR(reactor,1,length)) ;
124                          .AND. filesthere
125  *
126               filterset=.F.
127               IF FOUND()                   && match on user's partial entry
128                  filterset=.T.
129                  SET FILTER TO ALLTRIM(sn04_choice)=UPPER(SUBSTR(reactor,1,length))
130                  GO TOP
131                  line=4
132                  count=0
133                  column=col1
134  *
135               ELSE                         && no match on user's partial entry
136                  string= "< Invalid choice, try again...>"
137                  centered=40-(LEN(string)/2)
138                  @ 20,holdcol1 CLEAR TO 20,holdcol2
139                  @ 20,centered SAY string
140                  INKEY(0)
141               ENDIF
142  *
143            ELSE                            && match on user's entry
144               minis=unitid
145               reactrname=sn04_choice
146               EXIT
147            ENDIF
148  *
149     ENDCASE
150  *
151     IF EOF()
152        SET FILTER TO
153        GO TOP
154        count=0
155     ENDIF
156  ENDDO
157  *
158  IF filterset
159     SET FILTER TO
160  ENDIF
161  *-------------------------------------------------------
162  *  If a reactor was chosen, make sure there's data available
163  *  to process before we go on and do the rest of the routines:
164  *
165  IF .NOT. EMPTY(reactrname) .AND. ;
```

```
   .NOT. (holdname=reactrname)
*
   holdname=reactrname
*-----------------------------------------------------------
*  Find the serial number file and reactor file.  If not there,
*  don't proceed:
*
   serialno="S"+minis+".dbf"
   rctrno="RCTR"+minis+".dbf"
*
   IF .NOT. FILE(rctrno) .OR. .NOT. FILE(serialno)
      string="No data available for " + ALLTRIM("&reactrname")
      centered=40-LEN(string)/2
*
      CLEAR
      @ 8,centered-2 TO 13,centered+length+2 DOUBLE
      @ 10,centered SAY string
      @ 11,centered SAY "Try again..."
*
      reactrname=SPACE(20)
      rownum=22
      DO rtrn WITH rownum
*
   ELSE
      SET SAFETY OFF
      USE &serialno
      INDEX ON asserno TO vendor
      INDEX ON asserno TO ansi        && no ansi data available,
*     INDEX ON serial TO ansi         && so will only look at
      SET SAFETY ON                   && vendor data for now
*
      USE &rctrno
      SET SAFETY OFF
      SET UNIQUE ON
      INDEX ON batch to ba_rctr
      INDEX ON batch+burnup_bin to ba_burn
      INDEX ON batch+rodarray TO ba_rdrry
      SET UNIQUE OFF
      SET SAFETY ON
*
   ENDIF
ENDIF
*
CLOSE DATABASE
*
RETURN
```

```
1     *==================================================================
2     *  Program:     rtrn.prg
3     *  Purpose:     Gives user a prompt to hit RETURN on a particular
4     *               line number.
5     *  Date:        01/89
6     *  Programmer:  Betty Lewis
7     *==================================================================
8     PARAMETERS rowval
9     *
10    string="<  Press any key to continue ...  >"
11    centered=40 - (LEN(string) / 2)
12 *
13    @ rowval,centered SAY string
14    inkey(0)
15 *
16    RETURN
```

```
 1 *=====================================================================
 2 *  Program:  serial.prg
 3 *  Purpose:  Selects a serial number to report on.
 4 *  Notes:    Called by sndb.prg.
 5 *  Date:     03/89
 6 *  Author:   B. Lewis
 7 *=====================================================================
 8 serialno="S"+minis
 9 SELECT 9
10 USE &serialno INDEX vendor,ansi
11 *
12 *  Present initial menu to user:
13 *
14 DO WHILE .NOT. logoff
15    CLEAR
16    @ 1,1 TO 22,79 DOUBLE
17    string="S E L E C T   A   S E R I A L   N U M B E R"
18    centered=40-LEN(string)/2
19    @ 2,centered SAY string
20 *
21    string="for " + ALLTRIM("&reactrname") + ":"
22    centered=40-LEN(string)/2
23    @ 3,centered SAY string
24    @ 4,2 TO 4,78 DOUBLE
25 *
26    @ 8,25 SAY "A - List by ANSI Serial Number"
27    @ 10,25 SAY "B - List by Vendor Serial Number"
28    @ 14,25 SAY "R - RETURN to Previous Menu"
29    @ 15,25 SAY "X - EXIT Data Base System"
30    @ 21,74 SAY "SN02"
31 *
32    string="Select A, B, R, or X - "
33    length=LEN(string)
34    centered=40-length/2
35 *
36    @ 21,centered-2 CLEAR TO 23,centered+length+2
37    @ 21,centered-2 TO 23,centered+length+2 DOUBLE
38    @ 22,centered SAY string
39 *
40 *  Get user's selection:
41 *
42    sn02_choice=" "
43    DO WHILE .NOT. sn02_choice $ "ABRX"
44       sn02_choice=" "
45       @ 22,centered+length GET sn02_choice PICTURE "!"
46       READ
47    ENDDO
48 *
49 *  Act on user's selection:
50 *
51    serialnumber=SPACE(14)
52    rtrnprev=.F.
53    DO CASE
54       CASE sn02_choice="R"
55            rtrnprev=.T.
```

```
56            EXIT
57  *
58        CASE sn02_choice="X"
59            logoff=.T.
60            EXIT
61  *
62        CASE sn02_choice="A"
63            ansi=.T.
64            DO srl_nmbr
65  *
66        CASE sn02_choice="B"
67            ansi=.F.
68            DO srl_nmbr
69     ENDCASE
70  *
71  *  Are we doing a report with serial number data only?
72  *
73  IF EMPTY(key) .AND. .NOT. logoff .AND. .NOT. rtrnprev
74     DO snreport
75  ENDIF
76  *
77  ENDDO
78  *
79  RETURN
80
```

```
1  *=========================================================================
2  *  Program:  snassm.prg
3  *  Purpose:  Provides a list of assembly types.
4  *  Notes:    Called by storage.prg.
5  *  Date:     03/89
6  *  Author:   B. Lewis
7  *=========================================================================
8  *
9  rctrno="RCTR"+minis
10 *-------------------------------------------------------
11 *  Set up the index on unique arrays for listing purposes
12 *
13    SELECT 8
14    USE eia INDEX eia_code
15 *
16    SELECT 7
17    USE &rctrno
18 *
19    SET SAFETY OFF
20    SET UNIQUE ON
21    INDEX ON rodarray TO ua_rctr
22    SET UNIQUE OFF
23    SET SAFETY ON
24 *
25    SET RELATION TO rodarray INTO eia
26 *
27    DECLARE recordno[200]
28    key=SPACE(10)
29    line=7
30    count=0
31 *-------------------------------------------------------
32 *  Present menu to user:
33 *
34    CLEAR
35    @ 2,0 TO 22,79 DOUBLE
36    string="S E L E C T   A N   A S S E M B L Y   T Y P E"
37    centered=40-LEN(string)/2
38    @ 3,centered SAY string
39 *
40    string="for " + ALLTRIM("&reactrname") + ":"
41    centered=40-LEN(string)/2
42    @ 4,centered SAY string
43 *
44    @ 5,1 TO 5,78 DOUBLE
45    @ 21,74 SAY "SND8"
46 *-------------------------------------------------------
47 *  List out available assembly types from the file:
48 *
49    DO WHILE .T.
50       IF EOF()
51          GO TOP
52          count=0
53          line=7
54          @ 6,5 CLEAR TO 20,70
55       ENDIF
```

```
*
      count=count+1
      recordno[count]=RECNO()
      place=LEN(STR(count))-1
      @ line,16 SAY SUBSTR(STR(count),place,2) + " - " + ;
                    ALLTRIM(eia->manufact) + ", " + ;
                    ALLTRIM(eia->array) + ", " + ;
                    ALLTRIM(eia->version)
*
      line=line+1
      SKIP
      IF line=20 .OR. EOF()
         line=7
*-------------------------------------------------------
*  Put together an input string based on the value of COUNT:
*
         IF count=1
            string="Select 1, R to RETURN, or X to EXIT - "
            length=LEN(string)
            centered=40-length/2
         ELSE
            string="Select 1-"+ALLTRIM(STR(count))+", R to RETURN or X to EXIT - "
            length=LEN(string)
            centered=40-length/2
         ENDIF
*-------------------------------------------------------
*  Get user's selection:
*
         valid=.F.
         valnum=.F.
         vallet=.F.
*
         DO WHILE .NOT. valid
            sn08_choice="  "
*
            @ 21,centered-2 CLEAR TO 23,centered+length+4
            @ 21,centered-2 TO 23,centered+length+4 DOUBLE
*
            IF .NOT. EOF()
               string="<< <ENTER> to continue listing... >>"
               @ 23,(40-LEN(string)/2) SAY string
            ENDIF
*
            @ 22,centered SAY string
            @ 22,centered+length GET sn08_choice PICTURE "!!"
            READ
*
            vallet=ALLTRIM(sn08_choice) $ "RX"
            valnum=VAL(sn08_choice) >= 1 .AND. VAL(sn08_choice) <= count
            valid=vallet .OR. valnum
            IF EMPTY(sn08_choice)
               valid=.T.
            ENDIF
         ENDDO
*-------------------------------------------------------
```

```
111  *  Get out of the loop if user has made a valid selection:
112  *
113           IF valid
114              IF .NOT. EMPTY(sn08_choice)
115                 EXIT
116              ENDIF
117           ENDIF
118  *
119           @ 6,5 CLEAR TO 20,70
120        ENDIF
121     ENDDO
122  *-------------------------------------------------------
123  *  Act on user's selection:
124  *
125     DO CASE
126        CASE ALLTRIM(sn08_choice)="R"
127             rtrnprev=.T.
128  *
129        CASE ALLTRIM(sn08_choice)="X"
130             logoff=.T.
131  *
132        OTHERWISE
133             GOTO recordno[VAL(sn08_choice)]
134             assm=eia->rodarray
135     ENDCASE
136  *
137  SELECT 7
138  USE
139  SELECT 8
140  USE
141  RETURN
```

```
1  *==========================================================================
2  *  Program:  snbatch.prg
3  *  Purpose:  Allows selection of a reactor batch for reports.
4  *  Notes:    Called by storage.prg
5  *  Date:     03/89
6  *  Author:   B. Lewis
7  *==========================================================================
8  rctrno="RCTR" + minis
9  *
10 SELECT 8
11 USE &rctrno INDEX ba_rctr,ba_burn,ba_rdrry
12 *
13 DECLARE recordno[300]
14 rtrnprev=.F.
15 line=6
16 count=1
17 *-------------------------------------------------------------------
18 *  Determine column numbers for data:
19 *
20 COUNT TO amount
21 DO CASE
22    CASE amount <= 15
23         startcolumn=34
24         inc=0
25 *
26    CASE amount <= 30
27         startcolumn=16
28         inc=27
29 *
30    CASE amount <= 45
31         startcolumn=11
32         inc=22
33 *
34    CASE amount <= 60
35         startcolumn=6
36         inc=16
37 *
38    CASE amount <= 75
39         column=3
40         inc=14
41 *
42    OTHERWISE
43         startcolumn=3
44         inc=14
45 ENDCASE
46 *-------------------------------------------------------------------
47 *  Check if burnup & bin or an assembly type have been previously selected.
48 *  If so, set a filter so we can stay within those bounds.
49 *
50    IF .NOT. EMPTY(bin)
51       SET ORDER TO 2
52       filterstring="burnup_bin='&bin'"
53       SET FILTER TO &filterstring
54       GO TOP
55    ENDIF
```

```
 56 *
 57    IF .NOT. EMPTY(assm)
 58       SET ORDER TO 3
 59       filterstring="rodarray='&assm'"
 60       SET FILTER TO &filterstring
 61       GO TOP
 62    ENDIF
 63 *-----------------------------
 64 *  Headers:
 65 *
 66    CLEAR
 67    @ 1,0 TO 22,79 DOUBLE
 68    string="S E L E C T   A N   A S S E M B L Y   B A T C H"
 69    centered=40-LEN(string)/2
 70    @ 2,centered SAY string
 71 *
 72    string="for "+ALLTRIM("&reactrname")+":"
 73    centered=40-LEN(string)/2
 74    @ 3,centered SAY string
 75 *
 76    @ 4,1 TO 4,78 DOUBLE
 77    @ 21,74 SAY "SN06"
 78 *-----------------------------------------
 79 *  List out available batches from the file:
 80 *
 81    GO TOP
 82    column=startcolumn
 83    DO WHILE .T.
 84 *
 85       IF EOF()                        && reset and start again
 86          GO TOP
 87          count=1
 88          line=6
 89          column=startcolumn
 90          @ 5,2 CLEAR TO 20,75
 91       ENDIF
 92 *
 93       @ line,column SAY STR(count,3) + " - " + ALLTRIM(batch)
 94 *
 95       recordno[count]=RECNO()
 96       count=count+1
 97       line=line+1
 98       SKIP
 99 *
100       getdata=.F.
101       IF .NOT. EOF()
102          IF line=21
103             line=6
104             column=column+inc
105 *
106             IF column>70
107                column=startcolumn
108                getdata=.T.
109             ENDIF
110 *
```

```
111          ENDIF
112       ELSE
113          getdata=.T.
114       ENDIF
115 *-----------------------------------------------------------
116 *  Put together an input string based on the value of COUNT:
117 *
118    IF getdata
119       IF count=1
120          string="Select 1, R to RETURN, or X to EXIT - "
121          length=LEN(string)
122          centered=40-length/2
123       ELSE
124          string="Select 1-"+ALLTRIM(STR(count-1))+", R to RETURN, or X to EXIT - "
125          length=LEN(string)
126          centered=40-length/2
127       ENDIF
128 *-----------------------------------------------------------
129 *  Get user's selection:
130 *
131       @ 22,2 TO 22,77 DOUBLE
132       @ 21,centered-4 CLEAR TO 23,centered+length+2
133       @ 21,centered-4 TO 23,centered+length+2 DOUBLE
134 *
135       IF .NOT. EOF()
136          string1="<<  <ENTER> to continue listing...  >>"
137          centered1=40-LEN(string1)/2
138          @ 23,centered1 SAY string1
139       ENDIF
140 *
141       valid=.F.
142       valnum=.F.
143       vallet=.F.
144       DO WHILE .NOT. valid
145          sn06_choice=SPACE(3)
146          @ 22,centered-2 SAY string
147          @ 22,centered+length-2 GET sn06_choice PICTURE "!!!"
148          READ
149 *
150          IF EMPTY(sn06_choice)
151             EXIT
152          ENDIF
153 *
154          vallet=ALLTRIM(sn06_choice) $ "RX"
155          valnum=VAL(sn06_choice) >= 1 .AND. VAL(sn06_choice) <= count
156          valid=vallet .OR. valnum
157       ENDDO
158 *-----------------------------------------------------------
159 *  Act on user's selection:
160 *
161       rtrnprev=.F.
162       DO CASE
163          CASE EMPTY(sn06_choice)
164 *             Just go around again
165 *
```

```
166          CASE "R" $ sn06_choice
167               rtrnprev=.T.
168               EXIT
169 *
170          CASE "X" $ sn06_choice
171               logoff=.T.
172               EXIT
173 *
174          OTHERWISE
175               GOTO recordno[VAL(sn06_choice)]
176               key=batch
177               EXIT
178       ENDCASE
179 *
180       @ 5,2 CLEAR TO 20,75
181    ENDIF     && endif for GETDATA
182 *
183 ENDDO
184 *
185 SET FILTER TO
186 SELECT 8
187 USE
188 *
189 RETURN
```

```
1  *==============================================================================
2  * Program...: snburn.prg
3  * Purpose...: Lists available burnup categories for snreport.prg.
4  * Notes.....: Called by storage.prg.
5  * Date......: 03/89
6  * Author....: B. Lewis
7  *==============================================================================
8  *
9  SELECT 8
10 USE burn
11 COUNT TO number
12 *
13 DO WHILE .T.
14    SELECT 8
15    GO TOP
16 *
17 *  Display menu:
18 *
19 CLEAR
20 @  3,0  TO 22,79 DOUBLE
21 string='S E L E C T   A   B U R N U P   B I N'
22 centered=40-LEN(string)/2
23 @ 4,centered SAY string
24 *
25 string="for " + ALLTRIM("&reactrname") + ":"
26 centered=40-LEN(string)/2
27 @ 5,centered SAY string
28 *
29 @  6,1  TO 6,78 DOUBLE
30 *
31 line=9
32 column=16
33 DO WHILE .NOT. EOF()
34    @ line,column SAY STR(RECNO(),2,0) + ". " + burnn
35 *
36    line=line+1
37    IF line=16
38       line=9
39       column=40
40    ENDIF
41 *
42    SKIP
43 ENDDO
44 *
45 @ 18,26 SAY 'R. RETURN to previous menu'
46 @ 19,26 SAY 'X. EXIT the data base system'
47 @ 21,74 SAY 'SN07'
48 *
49 string="Select 1-" + ALLTRIM(STR(number)) + ", R, or X - "
50 length=LEN(string)
51 centered=40-length/2
52 *
53 @ 21,centered-2 CLEAR TO 23,centered+length+4
54 @ 21,centered-2 TO 23,centered+length+4 DOUBLE
55 @ 22,centered SAY string
```

```
 56 *
 57 *  Get user's selection:
 58 *
 59 vallet=.F.
 60 valnum=.F.
 61 valid=.F.
 62 *
 63 DO WHILE .NOT. valid
 64    sn07_choice = '  '
 65    @ 22,centered+length GET sn07_choice PICTURE "!!"
 66    READ
 67 *
 68    vallet=ALLTRIM(sn07_choice) $ "RX"
 69    valnum=VAL(sn07_choice) >= 1 .AND. VAL(sn07_choice) <= 13
 70    valid=vallet .OR. valnum
 71 ENDDO
 72 *
 73 *  Act on user's selection:
 74 *
 75 bin = SPACE(2)
 76 burn = SPACE(13)
 77 rtrnprev=.F.
 78 *
 79 DO CASE
 80    CASE ALLTRIM(sn07_choice) = 'R'
 81         rtrnprev=.T.
 82         EXIT
 83 *
 84    CASE ALLTRIM(sn07_choice) = 'X'
 85         logoff=.T.
 86         EXIT
 87 *
 88    OTHERWISE
 89         GOTO VAL(sn07_choice)
 90         bin=binn
 91         burn=burnn
 92 *-------------------------------
 93 *  Check for data availability:
 94 *
 95         rctrno="RCTR"+minis
 96         SELECT 7
 97         USE &rctrno
 98 *
 99         SET SAFETY OFF
100         INDEX ON burnup_bin TO bin_rctr
101         SET SAFETY ON
102 *
103         SEEK bin
104         IF .NOT. FOUND()
105            SAVE SCREEN
106            @ 10,20 CLEAR TO 13,60
107            @ 10,20 TO 13,60 DOUBLE
108            @ 11,25 SAY "No data available for report..."
109            DO rtrn WITH ROW()+1
110            RESTORE SCREEN
```

```
111          ELSE
112             EXIT
113          ENDIF
114  ENDCASE
115  *
116  ENDDO
117  SELECT 7
118  USE
119  SELECT 8
120  USE
121  RETURN
```

```
*==========================================================================
*  Program:  sncdb.prg
*  Purpose:  Presents intial entry screen into CDB Program for the Serial
*            Number Data Base system.
*  Date:     03/89
*  Author:   B. Lewis
*==========================================================================
*
CLEAR
*
@ 0,0 TO 23,79 DOUBLE
@ 2,12 SAY "        CCCCC         DDDDDDD         BBBBBBB"
@ 3,12 SAY "       CC   CC         DD   DD         BB   BB"
@ 4,12 SAY "      CC               DD   DD         BB   BB"
@ 5,12 SAY "      CC               DD   DD         BBBBBB"
@ 6,12 SAY "      CC               DD   DD         BB   BB"
@ 7,12 SAY "       CC   CC         DD   DD         BB   BB"
@ 8,12 SAY "        CCCCC         DDDDDDD         BBBBBBB"
*
@ 10,12 SAY "              Characteristics Data Base "
*
@ 11,1 TO 11,78 DOUBLE
*
string= "SERIAL NUMBER DATA BASE (SNDB):"
centered=40-(LEN(string)/2)
@ 12,centered SAY string
*
@ 14,1 SAY "       Program provides a link between an individual assembly's"
@ 15,1 SAY "       serial number (ANSI-Standard or vendor-specific) and the "
@ 16,1 SAY "       burnup/enrichment/discharge information in the LWR Quantities"
@ 17,1 SAY "       Data Base."
*
@ 18,1 TO 18,78 DOUBLE
@ 19,1 SAY "     For Assistance       Karl J. Notz        or      R. Scott Moore"
@ 20,1 SAY "        Contact:          (615) 574-6632              (615) 482-6601"
@ 21,1 SAY "                          FTS 624-6632                FTS 482-2688"
*
num=23
DO rtrn WITH num
*
RETURN
```

```
 1 *==========================================================================
 2 *  Program:  sndb.prg
 3 *  Purpose:  Main menu for serial number data base routines.
 4 *  Date:     03/89
 5 *  Author:   B. Lewis
 6 *==========================================================================
 7 *  Intro and variable initialization:
 8 *
 9 DO sninit
10 DO sncdb
11 holdname=SPACE(20)
12 *-----------------------------------
13 *  Get a reactor name:
14 *
15 DO WHILE .NOT. logoff
16    minis=SPACE(4)
17    reactrname=SPACE(20)
18    serialnumber=SPACE(14)
19    DO rctrname
20 *-----------------------------------
21 *  Present menu of options to user
22 *  If a reactor name was chosen:
23 *
24 IF .NOT. EMPTY(reactrname)
25    serialno="S"+minis
26    SELECT 9
27    USE &serialno INDEX vendor,ansi
28 *
29    rtrnprev=.F.
30    DO WHILE .NOT. logoff .AND. .NOT. rtrnprev
31 *
32       bin=SPACE(2)
33       burn=SPACE(13)
34       key=SPACE(10)
35       assm=SPACE(10)
36 *
37       CLEAR
38       @ 1,1 TO 22,79 DOUBLE
39 *
40       string="S E R I A L   N U M B E R   D A T A   B A S E   R E P O R T"
41       centered=40-(LEN(string)/2)
42       @ 2,centered SAY string
43 *
44       string="for " + ALLTRIM("&reactrname")+":"
45       centered=40-(LEN(string)/2)
46       @ 3,centered SAY string
47 *
48       @ 4,2 TO 4,78 DOUBLE
49       @ 21,74 SAY "SN01"
50       @ 8, 25 SAY "A - Report by Serial Number"
51       @ 10,25 SAY "B - Report by Batch Number"
52       @ 12,25 SAY "C - Report by Burnup Bin"
53       @ 14,25 SAY "D - Report by Assembly Type"
54       @ 17,25 SAY "R - RETURN to Previous Menu"
55       @ 18,25 SAY "X - EXIT Data Base System"
```

```
 56 *
 57       string="Select an A-D, R, or X - "
 58       length=LEN(string)
 59       centered=40-length/2
 60 *
 61       @ 21,centered-2 CLEAR TO 23,centered+length+1
 62       @ 21,centered-2 TO 23,centered+length+1 DOUBLE
 63       @ 22,centered SAY string
 64 *----------------------------------
 65 *  Get user's selection:
 66 *
 67       sn01choice=" "
 68       DO WHILE .NOT. sn01choice $ "ABCDRX"
 69          sn01choice=" "
 70          @ 22,centered+length GET sn01choice PICTURE "!"
 71          READ
 72       ENDDO
 73 *----------------------------------
 74 *  Act on user's selection:
 75 *
 76       DO CASE
 77          CASE sn01choice="R"
 78               EXIT
 79 *
 80          CASE sn01choice="X"
 81               logoff=.T.
 82               EXIT
 83 *
 84          CASE sn01choice="A"
 85               DO serial
 86 *
 87          CASE sn01choice="B"
 88               DO batch
 89 *
 90          CASE sn01choice="C"
 91               DO burnup
 92 *
 93          CASE sn01choice="D"
 94               DO assembly
 95       ENDCASE
 96 *
 97       rtrnprev=.F.
 98    ENDDO
 99 *
100 ENDIF
101 ENDDO
102 *
103 CLOSE ALL
104 CLEAR
105 RETURN
```

```
1  *==========================================================================
2  *  Program:  sninit.prg
3  *  Purpose:  Initializes database routines & variables for serial number
4  *            data base programs.
5  *  Date:     03/89
6  *  Author:   B. Lewis
7  *==========================================================================
8  *
9  SET TALK OFF
10 SET ECHO OFF
11 SET ESCAPE OFF
12 SET CONFIRM ON
13 SET COLOR TO W/B+
14 *
15 minis=SPACE(4)
16 reactrname=SPACE(20)
17 holdname=SPACE(20)
18 serialnumber=SPACE(14)
19 key=SPACE(10)
20 assm=SPACE(5)
21 bin=SPACE(2)
22 storage=.F.
23 ansi=.F.
24 discharged=0
25 years=0
26 *
27 PUBLIC rtrnprev,logoff,minis,reactrname,serialnumber
28 PUBLIC key,assm,bin,holdname,storage,ansi,discharged,years
29 *
30 minis=SPACE(4)
31 reactrname=SPACE(20)
32 holdname=SPACE(20)
33 serialnumber=SPACE(14)
34 key=SPACE(10)
35 assm=SPACE(5)
36 bin=SPACE(2)
37 burn=SPACE(13)
38 storage=.F.
39 *
40 RETURN
```

```
1  *=========================================================================
2  *  Program:  snreport.prg
3  *  Purpose:  Prints report matching parameters user specifies.
4  *  Notes:    Called by serial.prg and storage.prg.
5  *  Date:     03/89
6  *  Author:   B. Lewis
7  *=========================================================================
8  *
9  serialno="S"+minis
10 rctrno="RCTR" + minis
11 printwhere=" "
12 outfile=SPACE(12)
13 *
14 col1=1
15 col2=13
16 col3=29
17 col4=44
18 col5=56
19 col6=68
20 *
21 SELECT 9
22 USE &serialno INDEX vendor,ansi
23 *
24 SELECT 6
25 USE &rctrno INDEX ba_rdrry
26 *
27 SELECT 7
28 USE pool INDEX pool
29 *
30 SELECT 8
31 USE eia INDEX eia_code
32 *
33 *  Establish the relationships between the serial number file and
34 *  the other data base files.
35 *
36 SELECT 9
37 SET RELATION TO SUBSTR(rodarray,1,5) INTO eia, ;
38              TO poolid+" " INTO pool,  ;
39              TO SUBSTR(batch,1,6)+SUBSTR(rodarray,1,5) INTO &rctrno
40 *
41 SEEK serialnumber
42 IF .NOT. FOUND()
43    string="No data available."
44    length=LEN(string)
45    centered=40-length/2
46 *
47    CLEAR
48    @ 9,centered-2 TO 12,centered+length+2 DOUBLE
49    @ 10,centered SAY string
50    @ 11,centered SAY "Try again..."
51    INKEY(0)
52 *
53 *  Got some data!  Let's print a report!!
54 *
55 ELSE
```

```
 56 *
 57 *  Establish where to send the report and whether or not to print
 58 *  the report to the screen as well as a file or the printer:
 59 *
 60    DO prnt WITH printwhere,outfile
 61 *
 62    IF printwhere="S"
 63       number=1
 64    ELSE
 65       number=2
 66    ENDIF
 67 *
 68 *  Cycle thru at least once for a screen printout, twice for both
 69 *  screen and either printer or file
 70 *
 71    FOR i = 1 TO number
 72       SELECT 9
 73 *
 74 *  Set up the report's destination:
 75 *
 76       DO CASE
 77          CASE i=2 .AND. printwhere="P"
 78               SET DEVICE TO PRINT
 79               SET PRINT ON
 80 *
 81          CASE i=2 .AND. printwhere="F"
 82               SET PRINTER TO &outfile
 83               SET DEVICE TO PRINT
 84               SET PRINT ON
 85 *
 86          OTHERWISE
 87               SET DEVICE TO SCREEN
 88               SET PRINT OFF
 89               SET PRINTER TO
 90               CLEAR
 91       ENDCASE
 92 *
 93 *  Print a header
 94 *
 95          line=REPLICATE("_",70)
 96          @ 1,4 SAY line
 97 *
 98          @ 2,5 SAY "Report for Reactor:  " + ALLTRIM(UPPER(reactrname))
 99          @ 2,65 SAY DATE()
100 *
101          IF EMPTY(serial)
102             string="Vendor Serial# " + ALLTRIM(serialnumber) + ;
103                    ", ANSI Serial# (Not Available)"
104          ELSE
105             string="Vendor Serial# " + ALLTRIM(serialnumber) + ;
106                                    ", ANSI Serial# " + ALLTRIM(serial)
107          ENDIF
108 *
109       @ 3,5 SAY string
110       @ 4,65 SAY "Batch No.:"
```

```
111  *
112        @ 5,5 SAY "Assembly Type: " + ALLTRIM(eia->manufact) + ", " + ;
113                  ALLTRIM(eia->array) + ", " + ;
114                  ALLTRIM(eia->version) + "-" + ;
115                  ALLTRIM(eia->pwr_bwr)
116  *
117        @ 5,65 SAY ALLTRIM(batch)
118  *
119        @ 6,4 SAY line
120  *
121        @ 8,col3 SAY "Initial"
122  *
123        @ 9,col1 SAY "Storage"
124        @ 9,col2 SAY "Burnup:"
125        @ 9,col3 SAY "Enrichment:"
126        @ 9,col4 SAY "Weight:"
127        @ 9,col5 SAY "Discharge"
128        @ 9,col6 SAY "Time"
129  *
130        @ 10,col1 SAY "Location:"
131        @ 10,col2 SAY "(MWD/MTIHM)"
132        @ 10,col3 SAY "(% U-235)"
133        @ 10,col4 SAY "(Grams)"
134        @ 10,col5 SAY "Date:"
135        @ 10,col6 SAY "Frame:"
136  *
137        @ 12,col1 SAY ALLTRIM(pool->name)
138  *
139        length=LEN(ALLTRIM(&rctrno->burnup))
140        length=11-length
141        @ 12,col2+length SAY  ALLTRIM(&rctrno->burnup)
142        temp=VAL(&rctrno->assay) / 1000
143        @ 12,col3 SAY  ALLTRIM(STR(temp))
144        @ 12,col4 SAY  ALLTRIM(&rctrno->weight)
145        @ 12,col5 SAY  ALLTRIM(&rctrno->dis_date)
146  *
147  *  Get time frame for calculations first time thru the loop:
148  *  Then do the calculations themselves:
149  *
150        IF i=1
151           heavy=VAL(&rctrno->weight)
152           burnups=VAL(&rctrno->burnup)
153           discharged=VAL(&rctrno->dis_date)
154           pwr_or_bwr=eia->pwr_bwr
155           years=0
156  *
157           DO gettime
158  *
159           string1=SPACE(10)
160           formula="C"                          && Do calculations for curies
161           DO calc WITH formula,years,burnups,heavy,pwr_or_bwr,string1
162  *
163           string2=SPACE(10)
164           formula="W"                          && Do calculations for watts
165           DO calc WITH formula,years,burnups,heavy,pwr_or_bwr,string2
```

```
166 *
167          string3=SPACE(10)
168          formula=" "                              && Do calculations for photons
169          DO calc1 WITH formula,years,burnups,heavy,pwr_or_bwr,string3
170 *
171          string4=SPACE(10)
172          formula="A"                         && Do calculations for neutrons-alpha
173          DO calc2 WITH formula,years,burnups,heavy,pwr_or_bwr,string4
174 *
175          string5=SPACE(10)
176          formula="F"                        && Do calculations for neutrons-fission
177          DO calc2 WITH formula,years,burnups,heavy,pwr_or_bwr,string5
178 *
179          string6=SPACE(10)
180          formula="O"                         && Do calculations for neutrons-total
181          DO calc2 WITH formula,years,burnups,heavy,pwr_or_bwr,string6
182       ENDIF
183 *
184       @ 12,col6 SAY ALLTRIM(STR(years)) + " years"
185 *
186       @ 13,1 SAY REPLICATE("_",78)
187       string="Estimated Radiological Characteristics for this assembly:"
188       centered=40 - (LEN(string) / 2)
189       @ 14,centered SAY string
190 *
191       @ 16,col1 SAY "Heat Load:"
192       @ 16,col2 SAY "Activity:"
193       @ 16,col4 SAY "(alpha,n)"
194       @ 16,col5 SAY "S. Fission"
195       @ 16,col6 SAY "Total"
196 *
197       @ 17,col1 SAY "(watts)"
198       @ 17,col2 SAY "(curies)"
199       @ 17,col3 SAY "Photons:"
200       @ 17,col4 SAY "Neutrons:"
201       @ 17,col5 SAY "Neutrons:"
202       @ 17,col6 SAY "Neutrons:"
203 *
204       @ 19,col1 SAY string2
205       @ 19,col2 SAY string1
206       @ 19,col3 SAY string3
207       @ 19,col4 SAY string4
208       @ 19,col5 SAY string5
209       @ 19,col6 SAY string6
210 *
211 *  Turn off the output to file or printer if that's what the user chose:
212 *
213       DO CASE
214          CASE printwhere="P"
215               EJECT
216               SET PRINT OFF
217               SET DEVICE TO SCREEN
218 *
219          CASE printwhere="F"
220               SET DEVICE TO SCREEN
```

```
221                 SET PRINT OFF
222                 SET PRINTER TO
223                 @ 22,0 CLEAR TO 22,79
224                 mess="< See " + ALLTRIM(outfile) + ".PRN for copy of report... >"
225                 centered=40-LEN(mess)/2
226                 @ 22,centered SAY mess
227 *
228            OTHERWISE
229         ENDCASE
230 *
231         IF i=number
232            rownum=23
233            DO rtrn WITH rownum
234         ENDIF
235 *
236      NEXT
237 *
238 ENDIF
239 *
240 RETURN
```

```
 1 *=========================================================================
 2 *  Program:  srl_nmbr.prg
 3 *  Purpose:  Provides a list of all known serial numbers if user needs
 4 *            assistance in making a selection.
 5 *  Notes:    Called by serial.prg and snbatch.prg.
 6 *  Date:     03/89
 7 *  Author:   B. Lewis
 8 *=========================================================================
 9 SELECT 9
10 serialno="S" + minis
11 USE &serialno INDEX vendor,ansi
12 *
13 DECLARE recordno[2000]
14 serialnumber=SPACE(14)
15 line=7
16 rec_count=1
17 col1=15
18 col2=45
19 column=col1
20 *-------------------------------------------------------
21 *  Set up filters if program is coming here from either
22 *  SNASSM or SNBURN or SNBATCH:
23 *
24 DO filters
25 *-------------------------------------------------------
26 *  List out available serial numbers from the file:
27 *
28 nodata=.F.
29 GO TOP
30 IF EOF()
31   nodata=.T.
32 ENDIF
33 *
34 IF .NOT. nodata
35 *-------------------------------------------------------
36 * Headers:
37 *
38 CLEAR
39 @ 2,1 TO 22,79 DOUBLE
40 string="S E L E C T   A   S E R I A L   N U M B E R"
41 centered=40-LEN(string)/2
42 @ 3,centered SAY string
43 *
44 string="for "+ALLTRIM("&reactrname")+":"
45 centered=40-LEN(string)/2
46 @ 4,centered SAY string
47 *
48 @ 5,2 TO 5,78 DOUBLE
49 @ 6,col1+6 SAY "VENDOR# (ANSI#)"
50 @ 6,col2+6 SAY "VENDOR# (ANSI#)"
51 @ 21,74 SAY "SN11"
52 *
53 DO WHILE .T.
54 *
55    IF EOF()                  && reset and start again
```

```
56        line=7
57        rec_count=1
58        column=col1
59        GO TOP
60     ENDIF
61  *
62     recordno[rec_count]=RECNO()
63  *
64     IF EMPTY(serial)
65        ansi_num=" (NO ANSI)"
66     ELSE
67        ansi_num=" ("+ALLTRIM(serial)+")"
68     ENDIF
69  *
70     @ line,column SAY STR(rec_count,4) + " - " + ALLTRIM(asserno)+ansi_num
71  *
72     SKIP
73     line=line+1
74     rec_count=rec_count+1
75  *
76     getdata=.F.
77     IF .NOT. EOF()
78        IF line=20
79           line=7
80  *
81           IF column=col1
82              column=col2
83           ELSE
84              column=col1
85              getdata=.T.
86           ENDIF
87        ENDIF
88  *
89     ELSE
90        getdata=.T.
91        column=col1
92        line=7
93     ENDIF
94  *-----------------------------------------
95  *  If user is ready for data, let's get it:
96  *
97     IF getdata
98        IF (rec_count-1) = 1
99           string="Select a serial #, 1, R, or X to EXIT - "
100          length=LEN(string)
101          centered=40-length/2
102 *
103       ELSE
104          string="Select a serial #, 1-" + ALLTRIM(STR(rec_count-1)) + ;
105                 ", R, or X - "
106          length=LEN(string)
107          centered=40-length/2
108       ENDIF
109 *
110       @ 21,centered-9 CLEAR TO 23,centered+length+9
```

```
111        @ 21,centered-9 TO 23,centered+length+9 DOUBLE
112  *
113        IF .NOT. EOF()
114           @ 23,40-LEN(string)/2 SAY "<<  <ENTER> to continue listing...  >>"
115        ENDIF
116  *
117        sn11_choice=SPACE(14)
118        @ 22,centered-7 SAY string
119        @ 22,centered+length-7 GET sn11_choice PICTURE "!!!!!!!!!!!!!!"
120        READ
121  *-----------------------------------------
122  *  Act on user's selection:
123  *
124        rtrnprev=.F.
125        DO CASE
126           CASE ALLTRIM(sn11_choice)="R"                  && return
127                rtrnprev=.T.
128                EXIT
129  *
130           CASE ALLTRIM(sn11_choice)="X"                  && exit
131                logoff=.T.
132                EXIT
133  *
134           CASE EMPTY(sn11_choice)                        && no choice made
135  *             No choice made, just loop around again
136  *                                                       && invalid choice
137           CASE VAL(sn11_choice) < 0 .OR. VAL(sn11_choice) > rec_count-1
138                @ 23,1 CLEAR TO 23,78
139                @ 23,25 SAY "<< Invalid choice, try again... >>"
140                DO rtrn WITH ROW()+1
141  *                                                       && selection made
142           CASE VAL(sn11_choice) >= 1 .AND. VAL(sn11_choice) <= rec_count-1
143                GOTO recordno[VAL(sn11_choice)]
144                serialnumber=asserno
145                EXIT
146  *
147           OTHERWISE                                      && serial # entered
148                SEEK sn11_choice
149  *
150                IF .NOT. FOUND()
151                   @ 23,1 CLEAR TO 23,78
152                   @ 23,25 SAY "<< Invalid choice, try again... >>"
153                   DO rtrn WITH ROW()+1
154  *
155                ELSE
156                   serialnumber=asserno
157                ENDIF
158        ENDCASE
159  *
160        @ line,column CLEAR TO 20,77
161        getdata=.F.
162     ENDIF
163  *
164  ENDDO
165  *
```

```
ELSE
   string="No data available..."
   length=LEN(string)
   centered=40-length/2
   @ 12,centered-2 CLEAR TO 15,centered+length+2
   @ 12,centered-2 TO 15,centered+length+2 DOUBLE
   @ 13,centered SAY string
   @ 14,centered SAY "Hit any key..."
   INKEY(0)
   rtrnprev=.T.
ENDIF
*
   SET FILTER TO
RETURN
```

INTERNAL DISTRIBUTION

1. R. J. Andermann
2. R. C. Ashline
3. J. M. Begovich
4. A. G. Croff
5. R. M. Gove
6. V. T. Hinkel
7. D. S. Joy
8. E. K. Johnson
9. P. E. Johnson
10. J. A. Klein
11. S. B. Ludwig
12. A. P. Malinauskas
13. W. C. McClain
14. J. W. Nehls

15 - 24. K. J. Notz

25. W. J. Reich
26. R. Salmon
27. S. N. Storch
28. M. G. Stewart
29. T. D. Welch
30. Central Research Library
31. Laboratory Records Department
32. Laboratory Records, ORNL RC
33. ORNL Patent Section
34. Y-12 Technical Library

EXTERNAL DISTRIBUTION

DOE - Office of Civilian Radioactive Waste Management, Forrestal Building, 1000 Independence Avenue, S. W., Washington, DC 20585

35. A. B. Brownstein
36. W. J. Danker
37. H. J. Hale
38. M. L. Payton
39. E. Svenson

DOE - Energy Information Administration, Washington, DC 20585

40. H. Chou
41. J. A. Disbrow
42. K. Gibbard

DOE - Albuquerque Operations Office, P. O. Box 5400, Albuquerque, NM 87115

43. K. Golliher
44. D. M. Lund

DOE - Idaho Operations Office, 785 DOE Place, Idaho Falls, ID 83402

45. M. Fisher
46. M. W. Shupe

DOE - Nevada Operations Office, P. O. Box 98518, Las Vegas, NV 89113

47. Carl Gertz
48. Ed Wilmot

DOE - Oak Ridge Operations Office, P. O. Box 2001, Oak Ridge, TN 37831

49. Office of Assistant Manager for Energy Research

DOE - Richland Operations Office, P. O. Box 550, Richland, WA 99352

50. D. C. Langstaff

Lawrence Livermore National Laboratory, P. O. Box 808, Livermore, CA 94550

51. Les Jardine
52. Ray Stout

Pacific Northwest Laboratory, P. O. Box 999, Richland, WA 99352

53. G. H. Beeman
54. M. E. Cunningham
55. A. T. Luksic
56. R. W. Walling

Sandia National Laboratories, P. O. Box 5800, Albuquerque, NM 87185

57. J. Cashwell
58. A. W. Dennis
59. E. Ryder

ASG, Inc., 800 Oak Ridge Turnpike, Oak Ridge, TN 37830

60. E. A. Lewis
61. R. S. Moore

DataPhile, Inc., 9121 Garrison Rd., Knoxville, TN 37919

62. K. E. Jones

David Andress Associates, Inc., 11008 Harriett Lane, Kensington, MD 20895

63. D. Andress

EG&G Idaho, Inc., P. O. Box 1625 Idaho Falls, ID 83415

64. H. Worle

Electric Power Research Institute, P. O. Box 10412, Palo Alto, CA 94303

65. R. W. Lambert
66. O. Ozer

E. R. Johnson Associates, Inc., 10461 White Granite Drive., Suite 204, Oakton, VA 22124

67. N. B. McLeod
68. M. White

General Atomics, P. O. Box 85608, San Diego, CA 92138

69. S. D. Su
70. J. Boshoven

H&R Technical Associates, 575 Oak Ridge Turnpike, Oak Ridge, TN 37830

71. W. R. Rhyne

Science Applications Int. Corp., 800 Oak Ridge Turnpike, Oak Ridge, TN 37830

72. R. Best
73. R. W. Peterson

Science Applications Int. Corp., 101 Convention Center Dr., Las Vegas, NV 89109

74. R. Morrisette

Westinghouse Hanford Operations, P. O. Box 800, Richland, WA 99352

75. R. A. Watrous

Westinghouse Idaho Nuclear Company, P. O. Box 4000, Idaho Falls, ID 83404

76. D. A. Knecht

Roy F. Weston, Inc., 955 L'Enfant Plaza, SW., Eighth Floor, Washington, DC 20024

77. M. Conroy
78. J. DiNunno

R. F. Weston, 1 Weston Way, West Chester, PA 19380

79. R. R. Macdonald

80-89. Office of Scientific and Technical Communication,
P. O. Box 62, Oak Ridge, TN 37831