

ORNL/TM-11554

**OAK RIDGE  
NATIONAL  
LABORATORY**

**MARTIN MARIETTA**

**Characteristics Data Base:  
Programmer's Guide to the  
High-Level Waste Data Base**

OAK RIDGE NATIONAL LABORATORY

CENTRAL RESEARCH LIBRARY

CIRCULATION SECTION

ROOM ROOM 173

**LIBRARY LOAN COPY**

DO NOT TRANSFER TO ANOTHER PERSON

If you wish someone else to use this report, send to area with report and the library will arrange a loan.

ORNL-11554-1

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

This report has been read, used directly, from the best available copy.

Available to DOE and DOC contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831, prices available from (615) 576-8401, FTS 635-6401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5255 Port Royal Rd., Springfield, VA 22161  
NTIS price codes—Printed Copy: A04 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.

Chemical Technology Division

CHARACTERISTICS DATA BASE: PROGRAMMER'S GUIDE  
TO THE HIGH-LEVEL WASTE DATA BASE

K. E. Jones

DataPhile, Inc.  
Knoxville, Tennessee

R. Salmon

Chemical Technology Division

**NOTICE** This document contains information of a preliminary nature.  
It is subject to revision or correction and therefore does not represent a  
final report.

Date Published - August 1990

Report prepared by  
DataPhile, Inc., Knoxville, TN  
and  
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 0315110 3



## 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 and Clipper Compiler .....	3
2.3.2 MICROSOFT Linker .....	4
2.3.3 SAYWHAT Screen Generator .....	4
2.3.4 Microsoft C Compiler .....	5
2.4 MISCELLANEOUS .....	5
2.4.1 HDISPLAY.EXE .....	5
2.4.2 DBPRINT.EXE .....	6
3.0 MENU FLOW CHART .....	7
4.0 DATA FILE STRUCTURES .....	9
4.1 TEXT FILE TABLES .....	9
4.2 DBASE III FORMAT DATA FILES .....	10
5.0 PROGRAMS .....	14
5.1 MAJOR CODE FILES .....	14
5.2 MENU PROCEDURES .....	14
5.3 REPORT PROCEDURES .....	16
5.4 OTHER CODE CONSIDERATIONS .....	18
5.4.1 FIXNDX.PRG .....	18
5.4.2 WRITEIT Procedure .....	18
5.4.3 Operation on a Two Floppy System .....	18
5.5 INSTALLATION PROGRAMS .....	19
APPENDIX A -- PROGRAM LISTINGS .....	A-1



## PREFACE

The High-Level Waste 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 High-Level Waste Data Base has been published (see reference 2 on page 1), 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

LWR Quantities Data Base

LWR Serial Numbers 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 High-Level Waste Data Base is a menu-driven PC data base developed as part of OCRWM's technical data base on the characteristics of potential repository wastes, which also includes spent fuel and other materials. This programmer's guide completes the documentation for the High-Level Waste Data Base, the user's guide having been published previously. 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.



## 1.0 INTRODUCTION

The High Level Waste Database System (HLW System) describes the physical, chemical, and radiological properties of the nation's high level waste inventories, both current and projected. Much of the information in this system comes directly from the report Characteristics of Spent Nuclear Fuel, High-Level Waste, and other Radioactive Waste Which May Require Long-Term Isolation<sup>1</sup>, Chapter 3 and Appendix 3B (Characteristics Report). This system also provides additional data on the decay and photon emissions of immobilized waste forms from each high level waste site, at various decay times. This data was calculated using the ORIGEN2 computer code. Data is included for the four high level waste sites: West Valley Demonstration Plant, Savannah River Plant, Hanford, and Idaho National Engineering Laboratory.

The HLW system is completely menu-driven and very easy to use. Please see "User's Guide to the High Level Waste Database System"<sup>2</sup> for instructions on using the system. The User's Guide was prepared to serve as a complete user's reference for the system; However, the document is generally not needed to use the database, because the menus are self-explanatory.

This Programmer's Guide will explain the technical design and development of the HLW system. It will provide a detailed outline of the data files, as well as an outline of the major code modules. It will explain briefly how the programs operate, and, where appropriate, will explain the system design. This should provide some insight into the underlying motivations, making the code easier to understand. This document will serve as a reference for a programmer who needs to understand the technical details of the system in order to modify or extend its capabilities. Additionally, the data file explanations may be useful to those who would like to do some further study of the data beyond what the menu options provide.

---

<sup>1</sup>U.S. Department of Energy, Characteristics of Spent Nuclear Fuel, High-Level Waste, and Other Radioactive Wastes Which May Require Long-Term Isolation, DOE/RW-0184, December 1987.

<sup>2</sup>User's Guide to the High Level Waste Database System, ORNL/TM-10213, Appendix 3D, Oak Ridge National Laboratory, Oak Ridge, TN, September 1986.

## 2.0 SYSTEM DEVELOPMENT

### 2.1 PROGRAM START-UP

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

1. Changes to the directory where the data and programs are stored.
2. Loads VIDPOP.COM, a memory resident screen manager<sup>3</sup>. More will be said about the screen generator in Section 2.3, Development Environment and Tools.
3. Runs the executable module, START.EXE. This starts the program, and the user can select various reports until he chooses to exit.
4. Removes VIDPOP.COM from memory.
5. Returns to the directory that was the default at step 1.

### 2.2 RESOURCE REQUIREMENTS

The Quantities Database system requires the following computer resources.

1. 300K of available memory (RAM).
2. 2 Floppy disk drives or approximately 1 Megabyte of fixed disk space.
3. DOS 2.1 or above.

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

### 2.3 DEVELOPMENT ENVIRONMENT AND TOOLS

This section will describe briefly the different tools used to produce the HLW system. It will also include some tips useful in reconstructing the programs. The primary software used in development was dBase III Plus and the Clipper compiler, but several other tools were used to provide additional features.

---

<sup>3</sup>VIDPOP.COM is a part of the screen generator SAYWHAT?!, produced by The Research Group.

### 2.3.1 dBASE III Plus<sup>4</sup> and Clipper Compiler<sup>5</sup>

The HLW System was developed using both dBase III Plus and the Clipper compiler, Summer 1987 version. Much of the work was done using the interpreted dBase language, since the interpreter speeds up development. After the program was almost completed, it was compiled in Clipper. The compiled version offers 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.

In order to take advantage of the interpreted dBase environment for ease of development, no Clipper-provided extensions to the language were utilized.

A few things must be done differently depending on whether dBase or Clipper is being used. We used the logical variable "Clipper" to control this. When this variable is declared PUBLIC, the compiler will initialize it to true. dBase, however, will initialize it to false. Then, the Clipper specific items are included within an IF clause, and are only seen if the variable "Clipper" is true. This way, the program will execute in the dBase environment because the interpreter never encounters the Clipper specific code. The Clipper specific code will be executed in the compiled version, since the logical variable will be true.

For compiling, the Make utility that is distributed with Clipper was used to keep track of system dependencies. This Make utility works like other such utilities, where you create a file defining the system dependencies, and then the utility handles re-compiling only such modules as necessary, after changes have been made to the source code. A file called HLW.MAK contains the information needed by the Make utility. After changing 1 or more modules, you can use this command to ensure that the program is updated:

```
MAKE HLW.MAK
```

This will re-compile all necessary files, and link them together to form START.EXE, the main executable module.

---

<sup>4</sup>dBase III Plus is a product of Ashton-Tate Corporation.

<sup>5</sup>Clipper is a product of Nantucket Corporation.

### 2.3.2 MICROSOFT Linker<sup>6</sup>

The program modules were linked using the Microsoft Object Linker, Version 3.05. A linker response file, HLW.ARF, contains the required parameters to link the system. Therefore, you can link the system using this command:

```
LINK @HLW.ARF
```

This will pull in all the necessary object modules and libraries needed to create the executable modules.

### 2.3.3 SAYWHAT Screen Generator

The screens were produced using the SAYWHAT?! screen generator. Using this program, screens are easily designed and then saved as files with a .SQZ extension. These screen files are distributed as part of the HLW system. The SAYWHAT?! memory resident utility VIDPOP.COM must be installed in memory before the HLW system main program (START.EXE) is loaded. As mentioned in Section 2.1, the batch file HLW.BAT, distributed with the system, loads VIDPOP.COM before START.EXE. It also removes VIDPOP.COM from memory when the user exits the system, so that the memory will be freed for use by other programs.

When resident in memory, VIDPOP.COM traps interrupt 10H (which normally writes a character on the screen) and examines the output. To send commands to VIDPOP, the program first sends two signal characters of 255H. VIDPOP will recognize these characters and then interpret subsequent characters as instructions for itself. Therefore, this dBase statement tells VIDPOP to display the screen MAIN (line 44 in CONTROL.PRG).

```
?? chr(255) + chr(255) + "MAIN/"
```

The two output bytes chr(255) signal VIDPOP that this is a command. The '/' at the end of the screen filename is a terminator. The assembly language program CLIPPOP accomplishes the same purpose in the compiled Clipper environment. CLIPPOP outputs the 2 chr(255) bytes, then the rest of the test string that was passed to it as a parameter. CLIPPOP (based on similar modules distributed with SAYWHAT?!) is written and assembled with the Microsoft Assembler. The object module is linked with the other object modules to form START.EXE.

---

<sup>6</sup>Product of Microsoft Corporation.

One other assembly language program, GETCHOIC.ASM, is needed to work with SAYWHAT?!. GETCHOIC is distributed with SAYWHAT?!. Both the assembly language code and the object module are distributed. GETCHOIC handles obtaining the user's choice from a moving bar menu. This object module is linked in with the other compiled modules to form START.EXE.

#### **2.3.4 Microsoft C Compiler<sup>7</sup>**

Some auxiliary programs used by the system were written in the C computer language and compiled using the Microsoft C Compiler Version 5.0. The C source code for these programs is included with the Clipper source code in the disks that accompany this document. Two of the programs, HDISPLAY.EXE and DBPRINT.EXE are described in the following sections, since they are used extensively in the Quantities Database System.

### **2.4 MISCELLANEOUS**

This section describes other miscellaneous tools and utilities used in the Quantities Database.

#### **2.4.1 HDISPLAY.EXE**

This program is used to display a text file on the PC screen. It allows the user to browse through the file using the arrow keys. The HLW program runs this program as an external program to display your reports on the screen, using Clipper's RUN command. HDISPLAY allows you to browse through the data, rather than scrolling it off the screen where you would be unable to review parts you had already seen. The program was written in Microsoft C, and the source code is included with the source code for the Quantities system.

To use this program, you would issue this command:

**HDISPLAY FILENAME.EXT [options]**

where FILENAME.EXT is any valid DOS filename and extension. HDISPLAY first looks for a second file with the same name and an extension of .HDG. It uses this file as a heading, and the contents of this file are displayed across the top portion of the screen. The heading file can have at most 10 lines. Then the contents of the primary text file are displayed underneath the heading lines, filling up the screen. When the user presses the arrow keys to scroll the text file, only the bottom portion of the file is scrolled; the top

---

<sup>7</sup>Product of Microsoft Corporation.

heading portion remains fixed. Creating a separate heading file avoids having to count lines and include page breaks during report generations, because it is all done after the database program has formatted the data lines for the report.

The options for the HDISPLAY program are listed below. Each option must be followed by a space, and then a number, **n**. You must use lower case for the option letters.

- h** Heading option. The next **n** lines will be considered the heading; they will be displayed on the top of the screen, with the remainder of the file in a window below it. The heading part is not scrolled with the rest of the file.
- l** Line length option. The line length will be **n**. If no line length is given, it will default to 80.
- t** Trash option. The first **n** lines will be discarded.

#### 2.4.2 DBPRINT.EXE

This utility program is very similar to HDISPLAY.EXE, described in the previous section. It is used to send text files to the printer rather than the screen. Like HDISPLAY, it looks for a heading file first, and displays the heading lines at the top of the page. It also numbers the pages. The program was written in Microsoft C, and the source code is included with the source code for the Quantities system.

DBPRINT has the following options. Each option must be followed by a space, and then a number, **n**. You must use lower case for the option letters.

- l** Line length option. The line length will be **n**. If no line length is given, it will default to 80.
- p** Page length option. The number of lines on a page will be **n**. If no page length is given, it will default to 57. This option is useful if you want to print reports in landscape mode. You can set the page length to be 44, and the heading will be correctly printed at the top of each page.

### 3.0 MENU FLOW CHART

Figures 1 and 2 outline the major procedures that correspond to menu options. Because there are so many options, the menu options are divided into two parts: the first figure contains the basic menu selections that can always be chosen. The second figure shows the query-specific menu options, which can only be chosen if the user has selected certain queries for menu option C that require additional user selections.

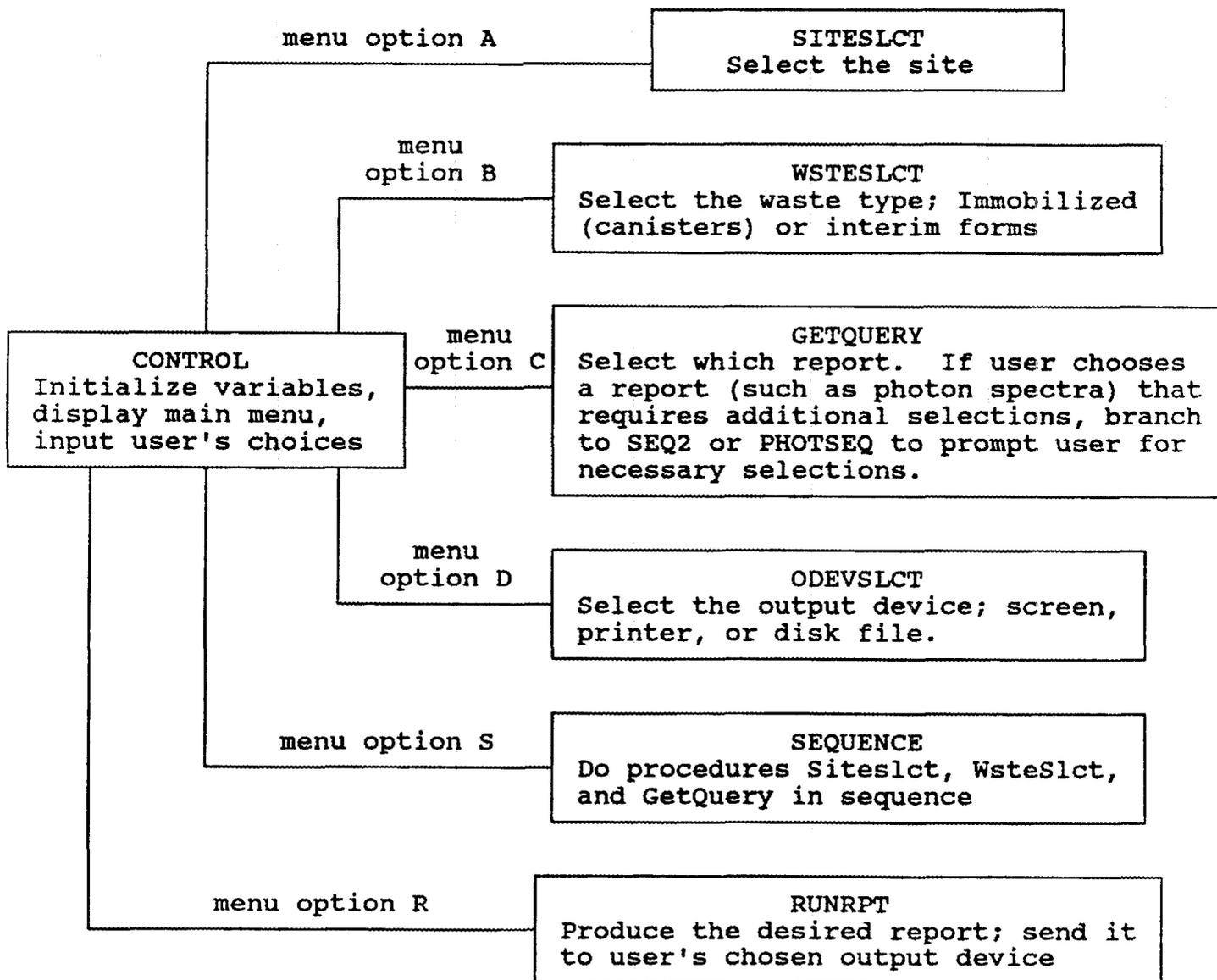


Figure 1. Procedures for Basic Menu Options

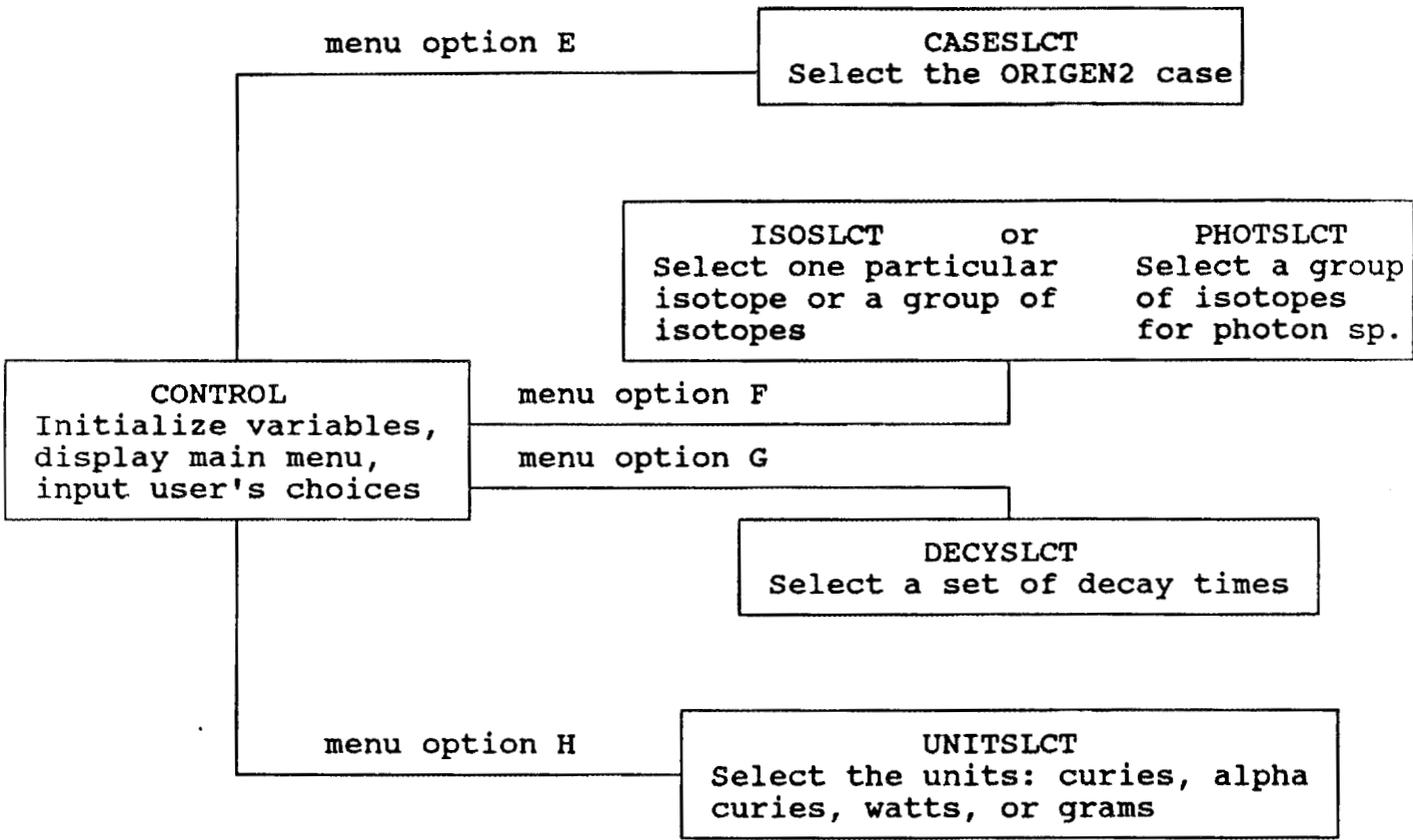


Figure 2. Procedures for Query-Specific Menu Options

## 4.0 DATA FILE STRUCTURES

This section will explain how the data files are organized in the HLW system. There are basically two different arrangements for the data included in this system.

1. Text files that are similar to a printed table on disk.
2. dBase III format data files (extension of DBF) that contain the ORIGEN2 generated data.

The next 2 sections explain each of these organizations.

### 4.1 TEXT FILE TABLES

The text files are tables taken directly from the Characteristics Report. When the user chooses a particular menu option, the program locates the corresponding text file and displays it on the screen or sends it to the printer or disk file, according to the user's selections (See section 5.3, REPORT PROCEDURES, for more information about how the programs work). These text files cover the Data Requested menu options 1-7 and 10 for immobilized waste forms, and all Data Requested menu options for interim waste forms.

This is a slightly unusual procedure for a database system, but we felt it was the simplest and best way to include the data. The high level waste data in the report was already well organized in these formatted tables. For the most part, it was small amounts of different types of data. This did not easily lend itself to the classic relational database model. We felt that it would not be cost effective to break down the already formatted tables into data base files, and then write programs to re-construct the tables. It seemed better to just incorporate the tables as text files. Since the tables had already been typed on a word processor for the DOE report, we were able to convert these tables into DOS format standard ASCII text files. It was then fairly simple to write programs that would select the correct tables according to the user's selection for Site, Waste Form, and Data Requested.

A naming convention was used to identify these text files.

1. The first 2 characters identify the waste form. MB --> Immobilized waste form, NT --> Interim waste form.
2. The third character identifies the number of the data requested menu option.

3. The next 3 or 4 characters identify the high level waste site.

WVDP--> West Valley Demonstration Plant  
 SRP --> Savannah River Plant  
 HANF--> Hanford  
 INEL --> Idaho National Engineering Laboratory  
 DEF --> All defense sites (SRP, HANF, INEL)  
 ALL --> All sites

4. An extension of TAB.

For example, a file name **MB1SRP.TAB** contains the table for Immobilized Waste, Data Requested menu option 1 (Description of Canisters and Waste Forms) for the site Savannah River.

#### 4.2 dBASE III FORMAT DATA FILES

dBase III format data files are used for the Data Requested menu options 8 and 9 for Immobilized Forms waste. These files contain the data calculated by ORIGEN2 for the curies, watts, alpha curies, grams, and photon emission spectra for the different isotopes contained in the canisters of high level waste. This data fits very readily into standard data files. The program selects the pertinent data and formats it into an output table according to the user's choices for Site, Isotopes, Decay Time, and Units.

The basic organization of the main files will explain much of the data contained in the system, and why the files are organized as they are. The detailed listing of fields following this explanation will serve as a complete reference. Three of the files, ORIGEN.DBF, ORITOT.DBF, and PHOTON.DBF are very similar.

Each record in the ORIGEN file contains the values for one particular isotope from one site, for one output case, for one unit (curies, watts, alpha curies, or grams), at 24 different decay times. A sample record looks like this:

```
SITEID:    HANF
OCASE:
UNITS:     C
ACTIVATION: A
ISOTOPE:   U234
ZNUM:      92
YR0:       1.750E-05
YR1:       1.877E-05
YR2:       2.010E-05
YR5:       2.416E-05
```

This shows the calculated curies (units = 'C') for the isotope U234 from the site Hanford at all the different decay times. We have only included four of the decay times, to keep things short. There is only one ORIGEN case for Hanford, the maximum radioactivity case, so the field OCASE is blank (There are two ORIGEN cases for the West Valley site, and only one case for all the others). Similar records would exist for all the other isotopes for all the other sites. In addition, similar records exist for the same isotopes with different units, such as grams or watts. The isotopes are included in this data only if the value for that unit is above a certain cutoff value, that is, if at any time it contributes ??% or greater of the total for that decay time. Therefore, it is possible to have records for certain isotopes with some units and not for others. For example, we might have an isotope present with the units of curies, since it is highly radioactive, but this isotope is not present with the units of grams, since it does not contribute significantly to the total weight.

The ACTIVATION field identifies to which activation group this isotope belongs: activation products, actinides and daughters, fission products, or all isotopes. This is used when the program needs to list all the isotopes in this activation group. The ZNUM field gives the atomic number for this isotope. This is included so that, by indexing the file on ZNUM, the output tables will list the isotopes by atomic number rather than alphabetically.

The ORITOT file is very similar. It contains only the totals for the different sites, units and activation groups, so there are only a few records per site. The totals are pre-calculated and stored in this file so that the system will be able to show the data to the user more quickly.

The PHOTON file is also very similar. In place of the ISOTOPE field, it has the SPECTRUM field, which contains the value of the average photon energy. For each site, ORIGEN2 case, and activation group, there are 18 records, giving the 18 different average photon energies. The UNITS field is set to 'P' for photons.

The following paragraphs describe each data file used in the HLW system.

#### ISOTOPE.DBF

Contains the different isotopes for each site. It is used to show a list of available isotopes when the user wants to see 1 specific isotope.

FIELD NAME	TYPE/SIZE	COMMENT
SITEID	C4	Identifies which site
OCASE	C1	Identifies which ORIGEN case
ISOTOPE	C7	Isotope name
ZNUM	N3	Z number for isotope

INDEX NAME                    INDEXED ON

ISOTOPE                        SITEID + OCASE + STR(ZNUM,3) + ISOTOPE

**ORIGEN.DBF**

Contains the curies, watts, grams, and alpha curies for all isotopes for each of the 4 high level sites.

FIELD NAME	TYPE/SIZE	DEFINITION
SITEID	C4	Identifies which site
OCASE	C1	Identifies which ORIGEN case
UNITS	C1	C--> Curies, G--> grams, W--> Watts, A-->alpha curies
ACTIVATION	C1	Which group of isotopes. V--> Activation products, A--> Actinides and daughters, F--> Fission products
ISOTOPE	C7	Isotope name
ZNUM	N3	Z number for isotope
YR0	C9	Value at year 0 -- time of immobilization
YR1	C9	Value at 1 year after immobilization
YR2	C9	Value at year 2
YR5	C9	Value at year 5
YR10	C9	Value at year 10
YR15	C9	Value at year 15
YR20	C9	Value at year 20
YR30	C9	Value at year 30
YR50	C9	Value at year 50
YR100	C9	Value at year 100
YR200	C9	Value at year 200
YR300	C9	Value at year 300
YR350	C9	Value at year 350
YR500	C9	Value at year 500
YR1000	C9	Value at year 1,000
YR1050	C9	Value at year 1,050
YR2K	C9	Value at year 2,000
YR5K	C9	Value at year 5,000
YR10K	C9	Value at year 10,000
YR20K	C9	Value at year 20,000

YR50K	C9	Value at year 50,000
YR100K	C9	Value at year 100,000
YR500K	C9	Value at year 500,000
YR1000K	C9	Value at year 1,000,000

INDEX NAME	INDEXED ON
------------	------------

ORIGEN	SITEID + OCASE + UNITS + ACTIVATION + STR(ZNUM,3) + ISOTOPE
--------	--

#### ORITOT.DBF

Contains total values of curies, watts, grams, and alpha curies for each site for each different activation groups. The structure is identical to ORIGEN.DBF, except that there is no Znum field. The isotope field is always set to "TOTAL".

INDEX NAME	INDEXED ON
------------	------------

ORITOT	SITEID + OCASE + UNITS + ACTIVATION
--------	-------------------------------------

#### PHOTON.DBF

Contains the photon spectra for each site for each different activation groups: activation products, Actinides and daughters, fission products, and total of all isotopes. The structure is identical to ORIGEN.DBF, except that there is no Znum field or Isotope field. Instead of Isotope, there is a field called Spectrum, which contains the mean value.

INDEX NAME	INDEXED ON
------------	------------

PHOTON	SITEID + OCASE + ACTIVATION + SPECTRUM
PHOTOTAL	SITEID + OCASE + SPECTRUM + ACTIVATION

## 5.0 PROGRAMS

This section will provide a high level explanation of the code in the HLW system. The first four sub-sections will describe the Clipper source code. The final sub-section, 5.5, will cover the batch file and programs used to install the system. A complete listing of all code is given in Appendix A.

### 5.1 MAJOR CODE FILES

The source code for the HLW system is organized into 3 files: START.PRG, CONTROL.PRG, and NOTZLIB.PRG. The files START and CONTROL are fairly small and contain only one subroutine, or procedure. Execution begins in the module START. START initializes global variables and branches to CONTROL. CONTROL opens all the database files. Then it sets up a loop which displays the main menu and inputs the user's selections, branching to the appropriate procedures based on the user's selection. This continues until the user enters an 'X' to exit the system.

NOTZLIB contains all the other procedures of the system. The procedures were combined into one file for convenience in keeping track of the code. The file was named after Karl Notz, head of the Characteristics Data Base program. The procedures can be roughly divided into two groups: procedures that handle the menus and procedures that produce the reports. These two activities will be covered in the following sections.

### 5.2 MENU PROCEDURES

The basic menu procedures are shown in the diagram in section 3.0. For the most part, these procedures are short and simple. They display a list of choices on the bottom half of the screen, and input the user's choice. Then they call the procedure WRITEIT to display the new choices on the main menu screen.

The menu procedures ensure that the user has entered valid responses, so that a valid query, or report, is always defined. Each menu routine checks the user's input and requires a valid response. There are two advantages to this procedure:

1. The user is prompted for a valid response immediately after a change in selections requires changes in other selections. This lets the user see instantly what needs to be changed and why it requires changes, as a result of his previous action.
2. The reporting procedures do not need to check for valid selections, since that has already been done.

To define a report, the user has to select a Site, a Waste Type (Immobilized or Interim Forms), and a Data Requested (a query). The available queries vary according to the values for Site and Waste Type. For example, there are four choices for Savannah River interim waste forms. There are nine choices for immobilized waste.

Suppose the user had chosen the Savannah River site, an Immobilized Waste Form, and Data Request Number 6 (Cumulative average radioactivity and thermal power per canister). Then, if he changed the site to West Valley Demonstration Plant, the Data Requested number 6 would no longer be valid, since that query is not available for West Valley. He would then be prompted for a new value for Data Requested, so that the defined report would be valid.

Keeping a valid query definition is accomplished by checking the query after any change in Site or Waste Type. If either of these items are changed, logic in the routine CONTROL will check to insure Data Requested value is available for the current values of Site and Waste Type. If the query is not available for that Site and Waste Type, the program will call the GETQUERY routine, and force the user to select a valid query.

Also, if the user chooses Data Requested option 8 (Radionuclide content) or 9 (Photon spectrum), he will be prompted for the necessary additional selections, so that a valid query will be completely defined. This is done by the routine GETQUERY. For Data Requested option 8, GETQUERY will call the routine SEQ2, which will in turn call CASESLCT (select the ORIGEN2 case, if the site is West Valley), ISOSLCT (select the isotope), DECYSLCT (select a set of decay times), and UNITSLCT (select the units). For Data Requested option 9, GETQUERY will call the routine PHOTSEQ, which will call PHOTSLCT (select the group of isotopes) and DECYSLCT (select a set of decay times).

### 5.3 REPORT PROCEDURES

Figure 3 shows the major procedures used to produce the reports.

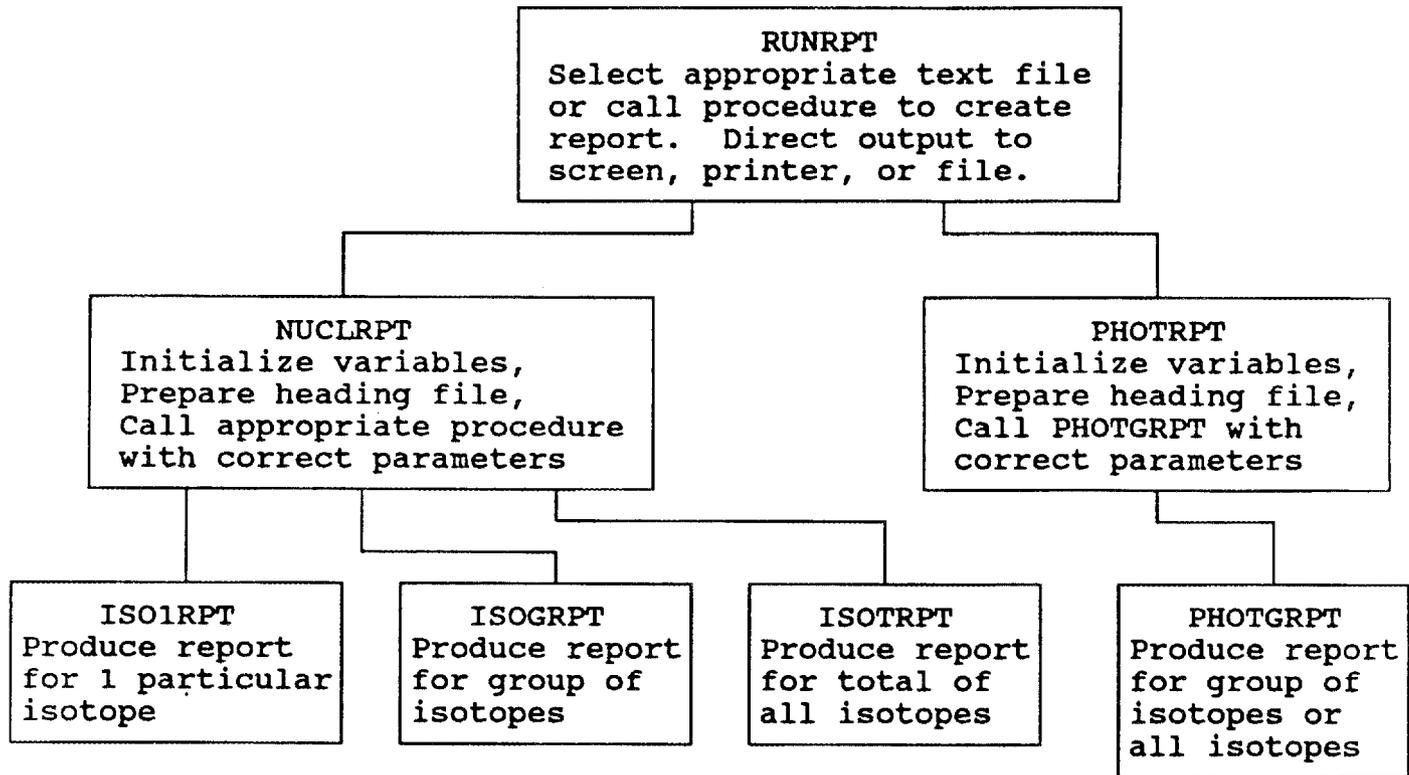


Figure 3. Major Reports Procedures

The procedure **RUNRPT** is the central procedure called to produce the reports. It calls the appropriate routines to produce the reports (if necessary) and then sends the formatted report file to the chosen output destination as follows:

1. If the output destination is the screen, the utility **HDISPLAY** (explained in Section 2.4.1) is used to display the report on the screen.
2. If the output destination is the printer, the utility **DBPRINT** (explained in Section 2.4.2) is used to send the report to the printer.
3. If the output destination is a disk file, the report is copied to a file of that name.

As explained in Section 4.1, many of the reports are preformatted text files. In these cases, it is not necessary to produce a report, only to select the correct file and send it to the appropriate output device. For these reports, **RUNRPT** constructs the filename from

the user's selections according to the naming convention described in Section 4.1. Then, using macros, this filename is directed to the selected output destination. No additional routines need to be called.

The HLW system must format reports from data in the database files for two queries: Radionuclide Content and Photon Spectra. RUNRPT calls the procedure NUCLRPT for the first query, and calls PHOTRPT for the second one. These procedures are very similar. They produce a formatted text report in a file by using the alternate file feature of Clipper. Then, after the report is completely generated, RUNRPT directs the file to the selected output device, just as it does with the pre-formatted text files.

For the query on Radionuclide Content (Data Requested option number 8 for Immobilized Waste Type) RUNRPT calls the procedure NUCLRPT. NUCLRPT initializes some variables, formats the page heading (calls the procedure HEADING), and then calls one of these procedures, according to the user's selection for Isotopes.

ISO1RPT -- report on 1 particular isotope

ISOGRPT -- report on a group of isotopes, such as Activation Products

ISOTRPT -- report on the total of all isotopes

These 3 procedures are quite straightforward. They locate the data in either the ORIGEN datafile or the ORITOT datafile, and format output lines with the values at the chosen decay times.

An interesting technique is used to accommodate the decay times. Users can choose a set of decay times. For example, one set is 0, 1, 2, 5, and 10 years. The corresponding data fields for the values at these years are YR0, YR1, YR2, YR5, and YR10. We set variables equal to these field names, such as this.

VAL1 = "YR0"

VAL2 = "YR2"

Then we can obtain the value of the field by using the macro operator (&). For example, suppose the field YR0 contains the value "1.50E-01", and VAL1 is equal to "YR0". Then &VAL1 is equal to "1.502E-01". Therefore, we set up these variables equal to the appropriate field names according to the currently chosen set of decay times. Then the following code statement will select the appropriate values (line 1067 in NOTZLIB.PRG).

```
line = Origen->isotope + " " + &val1 + " " + &val2 + " " + &val3 + ;
      " " + &val4 + " " + &val5 + " " + &val6
```

The procedure PHOTRPT is very similar to NUCLRPT. It produces the report on Photon Spectra. It initializes variables, formats the page heading (calls HEADING) and

then calls the procedure PHOTGRPT to produce the report on the user's selected group of isotopes. PHOTGRPT is similar to ISOGRPT. It locates the data in the photon datafile and formats output lines, using the same macro technique for the decay times.

## 5.4 OTHER CODE CONSIDERATIONS

### 5.4.1 FIXNDX.PRG

Fixndx.prg is a short program that contains the code used to create all indices used in the HLW system. It can be compiled and linked to form a separate executable module, FIXNDX.EXE. You can run this program at the DOS prompt to re-set all the indices. This is useful in system development if you are using different test database files and need to quickly set up indices for a different test system.

### 5.4.2 WRITEIT Procedure

This procedure is in the file NOTZLIB.PRG. It is used to update the main menu screen, by displaying the current selections. All updates to the menu screen are done through this procedure. The other menu selection procedures call WRITEIT with the new text to display, but not the screen location where it is to be displayed. WRITEIT alone contains that information. This organization is very useful because it is then easy to change the screen layout, and we only have to update the screen coordinates for the displayed items in one procedure, rather than having to change screen coordinates in several different procedures.

### 5.4.3 Operation on a Two Floppy System

The HLW system will work on either a computer with only two floppy disk drives or a computer with a fixed disk. In the case of the fixed disk, the entire system is copied into a subdirectory and the system operates from that subdirectory (see Section 5.5 for information about installation). Working on a two floppy disk system is more difficult, since the program and data requires three diskettes (360K each). We need some way to find the data files on another disk drive, and some way to check that the appropriate disk is in the second disk drive. If the appropriate disk is not in the drive, the user is prompted to switch disks. The problem is solved by including special "mark" files on the different disks.

The HLW system is distributed on three disks that contain the following files.

DISK 0 (PROGRAMS) -- Executable module, screen files, flopdisk.mrk

DISK 1 (DATA DISK 1) -- Origen.dbf, Oritot.dbf, data1.mrk

**DISK 2 (DATA DISK 2) -- Table text files, data2.mrk**

(This includes only the major files; it is not a complete listing.)

To use the HLW system on two floppy disks, put Disk 0 into the A: floppy drive and Disk 1 into the B: floppy drive. When the program begins, it checks the default drive (in this case A:) for the file FLOPDISK.MRK. If it finds this file, it assumes the use of a two floppy disk system, since the file FLOPDISK.MRK is deleted when you install the program onto a fixed disk using the INSTALL.BAT file. The program then sets the global variable "DRIVE" equal to "B:", and the global logical variable "FLOPPY" to true. The program will look for all data files on the B: drive by concatenating the variable "DRIVE" as a macro to data references. For example, when the program wants to open ORIGEN.DBF, it uses this command.

```
use &DRIVE.ORIGEN
```

which will expand to this command.

```
use B:ORIGEN
```

If the program cannot find the file FLOPDISK.MRK on the default drive, it sets the variable "DRIVE" equal to a null string, and the logical variable "FLOPPY" to false. Then, the macro expression &DRIVE has no value, and the data references will evaluate to the current default drive.

The two files DATA1.MRK and DATA2.MRK solve the second problem, that is, making sure that the correct data disk is in the B: drive. These two files are on Disk 1 and Disk 2, respectively. They are very small files that serve to "mark" which data diskette. It is immaterial what they contain. When the program needs to refer to some data file on the disk and the variable "FLOPPY" is true, it will check for the presence of the appropriate mark file, either DATA1.MRK or DATA2.MRK. If the file is not found, the program will prompt the user to change data diskettes and then press any key. After the user's keystroke, the program will check for the mark file again, in case the user did not actually insert the correct disk. This process continues until the mark file is found. Then the program will continue its operation.

## 5.5 INSTALLATION PROGRAMS

INSTALL.BAT, a DOS batch file, is used to install the HLW system onto the user's hard disk. To install the system, the user places the first distribution diskette in one of his floppy drives and issues the following command.

## INSTALL C

where "C" is the letter of the hard disk where he wants to install the database. It can be any letter that denotes a disk drive. The batch file accepts the disk drive argument with a colon (such as "C:") or without it.

Basically, INSTALL.BAT is a straightforward batch file. It makes a subdirectory called HLWD on the specified hard disk, and copies the contents of the distribution diskettes into that subdirectory. A batch file to start up the system, HLW.BAT is installed in the directory above the HLWD subdirectory.

In addition to the above basic operations, INSTALL.BAT checks if the current directory on the specified disk drive is the root directory using the program CHKROOT.EXE (written in C). If the directory is not the root, then CHKROOT asks the user if he wants the new software appended into the current directory. If the user answers yes, then a subdirectory named HLWD will be created under the current directory, and the HLW system installed there. If the user answers no, then the subdirectory HLWD will be created under the root directory, and the HLW system installed there. This provides added flexibility in installing the system.

APPENDIX A -- PROGRAM LISTINGS  
CONTROL.PRG

1

```

1 *****
2 * CONTROL.PRG Controls user selections. Initially, runs the Sequence *
3 * program which sequences through all selections. Then sets up a loop *
4 * where user can indicate which item he wants to change, and the program *
5 * branches to the appropriate routine(s). Finally, if the user selects 'X'*
6 * the program will ask for confirmation, then exit. *
7 *****
8
9 set talk off
10
11 sitetxt = " "
12 wtxt = " "
13 qtxt = " "
14 decytxt = " "
15 oDevice = "Screen"
16 odtxt = "Screen "
17 querytxt = " "
18 casetxt = " "
19 tcase = " "
20 isotxt = " "
21 unitxt = " "
22 fName = " "
23
24 * Indicates if the origen screen, with additional choices, is displayed
25 orgflag = .f.
26 * Indicates if the photon screen, with additional choices, is displayed
27 photflag = .f.
28
29 select 1
30 use &drive.origen index &drive.origen
31 *select 2
32 *use photon index photon
33 select 3
34 use isotope index isotope
35 select 4
36 use &drive.oritot index &drive.oritot
37 select 1
38
39 @ 0,0 clear
40 *run loader main
41 if clipper
42 call ClipPop with "Main"
43 else
44 ?? chr(255) + chr(255) + "MAIN/"
45 endif
46
47 finalAns = " "
48
49 do Refresh && put defaults on screen
50 do sequence
51
52 do while .T.
53 FINALANS='R'
54 @ 17, 0 clear
55 @ 15,23 get FINALANS picture 'I'
56 READ
57
58 do case
59
60 * Site
61 case FINALANS='A'
62 oldSite = site
63 do siteSlct
64 do writeIt with "S", sitetxt
65
66 * If origen queries have been chosen, need to check if site has changed
67 * to or from WVDP
68 if (site = 1)
69 if (oldSite <> site .and. (squery = 8 .or. squery = 9))
70 do CaseSlct
71 do WriteIt with "C", casetxt

```

APPENDIX A -- PROGRAM LISTINGS  
CONTROL.PRG

2

```

72         endif
73     else
74         if (oldSite = 1 .and. (squery = 8 .or. squery = 9))
75             set color to w+/bg
76             casetxt = nocstxt
77             do WriteIt with "C", casetxt
78             tcase = " "
79             set color to n/bg
80         endif
81     endif
82
83     qerr = .f.
84     if (waste = 1)
85         do case
86             case site = 1
87                 if squery = 5 .or. squery = 6
88                     qerr = .t.
89                 endif
90             case site = 2
91                 if squery = 5
92                     qerr = .t.
93                 endif
94
95             case site = 5
96                 if squery <> 2
97                     qerr = .t.
98                 endif
99
100            case site = 6
101                if squery > 2
102                    qerr = .t.
103                endif
104            endcase
105        else          && Interim form waste type
106            if (site > 4 .and. squery >= 2)
107                qerr = .t.
108            endif
109        endif
110        if (qerr)
111            querytxt = "**** Must be re-defined for the Waste Type"
112            do WriteIt with "D",querytxt
113            do GetQuery
114        endif
115
116    * Waste type
117        case FINALANS='B'
118            prevWaste = waste
119            do wasteSlct
120            if ((orgflag .or. photflag) .and. waste = 2)
121
122                if (clipper)
123                    call ClipPop with "Display Page 1"
124                else
125                    ?? chr(255) + chr(255) + "DISPLAY PAGE 1/"
126                endif
127
128                ? " "
129                do refresh
130                orgflag = .f.
131                photflag = .f.
132            else
133                do writeIt with "W", wtxt
134            endif
135
136    * If the Waste Type has changed, need to also change Data Requested
137        if waste <> prevWaste
138            querytxt = "**** Must be re-defined for the Waste Type"
139            do WriteIt with "D",querytxt
140            do GetQuery
141        endif
142

```

APPENDIX A -- PROGRAM LISTINGS  
CONTROL.PRG

3

```

143 * Query -- different options if immobilized or interim waste type
144     case FINALANS='C'
145     do GetQuery
146 * Output device
147     case FINALANS='D'
148     do ODevSlct
149     do writeIt with "O", odtxt
150
151 * Case (Average or Max. Only appropriate for Immobilized waste, queries
152 * 8 and 9, if site="WVDP"
153
154     case FINALANS='E'
155     if (orgflag .and. site = 1)
156     do caseSlct
157     do writeIt with "C", casetxt
158     else
159     if (photflag .and. site = 1)
160     do caseSlct
161     do WriteIt with "C", casetxt
162     else
163     if (photflag .or. orgflag)
164     @ 19,10 say "There are no other choices for case for this site."
165     else
166     @19,10 say "Please select one of the options indicated."
167     endif
168     wait
169     endif
170     endif
171
172 * Isotope -- only appropriate for origen queries
173     case FINALANS='P'
174     if (orgflag)
175     do isoSlct
176     do writeIt with "I", isotxt
177     else
178     if (photflag)
179     do PhotSlct
180     do WriteIt with "I", photxt
181     else
182     @19,10 say "Please select one of the options indicated."
183     wait
184     endif
185     endif
186
187 * Decay time -- only appropriate for origen queries
188     case FINALANS='G'
189     if (orgflag .or. photflag)
190     do decySlct
191     do writeIt with "T", decytxt
192     else
193     @19,10 say "Please select one of the options indicated."
194     wait
195     endif
196
197 * Units -- only appropriate for origen queries
198     case FINALANS='H'
199     if (orgflag)
200     do unitSlct
201     do writeIt with "U", unitxt
202     else
203     @19,10 say "Please select one of the options indicated."
204     wait
205     endif
206
207 * Run the report
208     case FINALANS='R'
209     do RUMRPT with fName
210     if (orgflag)
211     if (clippar)
212     call ClipPop with "Main"
213     call ClipPop with "Fill Page 1"

```

APPENDIX A -- PROGRAM LISTINGS  
CONTROL.PRG

4

```

214         call ClipPop with "Origen"
215     else
216         ?? chr(255) + chr(255) + "MAIN/"
217         ?? chr(255) + chr(255) + "FILL PAGE 1/"
218         ?? chr(255) + chr(255) + "ORIGEN/"
219     endif
220
221     do refresh
222     do refresh2
223     else
224     if (photflag)
225         if (clipper)
226             call ClipPop with "Main"
227             call ClipPop with "Fill Page 1"
228             call ClipPop with "Photon"
229         else
230             ?? chr(255) + chr(255) + "MAIN/"
231             ?? chr(255) + chr(255) + "FILL PAGE 1/"
232             ?? chr(255) + chr(255) + "PHOTON/"
233         endif
234
235         do refresh
236         @ 8, 43 say casetxt
237         @ 9,47 say photxt
238         @ 10,49 say decytxt
239     else
240         if (clipper)
241             call ClipPop with "Main"
242         else
243             ?? chr(255) + chr(255) + "MAIN/"
244         endif
245         do refresh
246     endif
247 endif
248
249 * Sequence through all choices
250     case FINALANS='S'
251         do sequence
252
253     case FINALANS='X'
254         theanswer = 'N'
255         @ 19,29 say 'EXIT SYSTEM SELECTED !'
256         @ 21,20 say 'Do you wish to end this session (Y/N)?' get theanswer
257 picture '! '
258         read
259         if theanswer='Y'
260             exit
261         endif
262     otherwise
263         @ 19,10 say 'Please select one of the options indicated.'
264         @ 21,15 say ''
265         wait
266     endcase
267 enddo
268
269 close databases
270 close alternate
271 return

```

APPENDIX A -- PROGRAM LISTINGS  
INSTALL.BAT

5

```
1 echo off
2 cls
3 if . == %1. goto Erri
4 echo                                OCRWM/ORNL
5 echo      High Level Radioactive Waste Personal Computer Database System      6 echo .
7 echo This installation procedure will transfer the High Level Waste
8 echo Database System from floppy disks to a hard disk.
9 echo .
10 echo The hard disk you have selected is drive- %1
11 echo .
12 echo If the drive letter is NOT correct, hold down Ctrl and press C to abort
13 echo the installation.
14 echo You will then be asked if you want to terminate the batch file, press Y.
15 echo .
16 echo If the drive letter is correct, then
17 pause
18
19 ChkRoot %1
20 if errorlevel 1 goto Done
21
22 chkArg %1
23 if errorlevel 1 goto Colon
24
25 copy HLW.bat %1:
26 md %1:HLWD
27 copy *.* %1:hld
28 del %1:hld\flopdisk.mrk
29 echo Remove the High Level Waste Database Programs Diskette, and insert
30 echo the High Level Waste Database Data #1 Diskette in the floppy drive.
31 pause
32 copy *.* %1:hld
33 echo Remove the High Level Waste Database Data #1 Diskette, and insert
34 echo the High Level Waste Database Data #2 Diskette in the floppy drive.
35 pause
36 copy *.* %1:hld
37 goto Next
38
39 rem Come here if argument had a colon -- same as above but leave off ':'
40 :Colon
41 copy HLW.bat %1
42 md %1HLWD
43 copy *.* %1hld
44 del %1hld\flopdisk.mrk
45 echo Remove the High Level Waste Database Programs Diskette, and insert
46 echo the High Level Waste Database Data #1 Diskette in the floppy drive.
47 pause
48 copy *.* %1hld
49 echo Remove the High Level Waste Database Data #1 Diskette, and insert
50 echo the High Level Waste Database Data #2 Diskette in the floppy drive.
51 pause
52 copy *.* %1hld
53
54 :Next
55 echo .
56 echo The installation is now complete. You may remove the floppy disk from
57 echo the drive. A batch file named HLW has been placed in the current
58 echo directory of your fixed disk. You may run the High Level Waste
```

APPENDIX A -- PROGRAM LISTINGS  
INSTALL.BAT

6

```
59 echo Database system by typing HLW when you are in that directory.
60 echo .
61 echo Good Luck!
62 goto Done
63
64 :Err1
65 echo You must indicate the letter of your hard disk. For example, if your
66 echo hard disk letter is C, you would enter the command
67 echo INSTALL C. Please start over.
68 goto Done
69
70 :Done
71 echo .
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

1 *****
2 * Sequence -- Sequences through all the choices on the screen. *
3 *****
4
5 procedure SEQUENCE
6
7 do siteSlct
8 do writeIt with "S", sitetxt
9 do wsteSlct
10 do writeIt with "W", wtxt
11 do GetQuery
12
13 return
14
15 *****
16 * Seq2 -- sequences through the additional choices for origen data; *
17 * isotope, decay time, and units. *
18 *****
19
20 procedure Seq2
21
22 if (clipper)
23     call ClipPop with "Main"
24     call ClipPop with "Fill Page 1"
25     call ClipPop with "Origen"
26 else
27     ?? chr(255) + chr(255) + "MAIN/"
28     ?? chr(255) + chr(255) + "FILL PAGE 1/"
29     ?? chr(255) + chr(255) + "ORIGEN/"
30 endif
31
32 *run loader origen
33 do refresh
34     orgflag = .t.
35     photflag = .f.
36     if (site = 1)                && West Valley
37         do CaseSlct
38     if (debug)
39         @ 21,0 clear
40         @ 21,10 say "After CaseSlct. casetxt: " + casetxt
41         wait
42     endif
43         do WriteIt with "C", casetxt
44     else
45         tcase = " "
46         casetxt = nocstxt        && Message saying only 1 case
47         do WriteIt with "C", nocstxt
48     endif
49     do isoslct
50     do writeIt with "I", isotxt
51     do decyslct
52     do writeIt with "T", decytxt
53     do unitSlct
54     do writeIt with "U", unitxt
55
56 return
57
58 *****
59 * PhotSeq -- sequences through the additional choices for the photon *
60 * data; isotope group, and time set. *
61 *****
62
63 procedure PhotSeq
64
65 if (debug)
66     @ 21,0 clear
67     @ 21,10 say "In PhotSeq. About to pop screens."
68     wait "" to dummy
69 endif
70

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

71 if (clipper)
72     call ClipPop with "Main"
73     call ClipPop with "Fill Page 1"
74     call ClipPop with "Photon"
75 else
76     ?? chr(255) + chr(255) + "MAIN/"
77     ?? chr(255) + chr(255) + "FILL PAGE 1/"
78     ?? chr(255) + chr(255) + "PHOTON/"
79 endif
80
81 *run loader photon
82 do refresh
83
84     photflag = .t.
85     orgflag = .f.
86
87     if (site = 1)           && West Valley
88 if (debug)
89     @ 21,0 clear
90     @ 21,10 say "In PhotSeq. Site = 1, about to call CaseSlct."
91     wait "" to dummy
92 endif
93     do CaseSlct
94     do WriteIt with "C", casetxt
95     else
96         tcase = " "
97         casetxt = nocstxt           && Message saying only 1 case
98         do WriteIt with "C", nocstxt
99     endif
100
101     do PhotSlct
102     do writeIt with "I", photxt
103     do decySlct
104     do writeIt with "T", decytxt
105
106 return
107
108 *****
109 * SiteSlct -- user selects the site. *
110 *****
111
112 procedure siteSlct
113 *
114
115 set color to n/bg
116 @ 17,0 clear
117 @ 17, 1 say '(A.) Select the Site: '
118
119
120 @ 18,5 say ' 1. '
121 @ 19,5 say ' 2. '
122 @ 20,5 say ' 3. '
123 @ 21,5 say ' 4. '
124 @ 22,5 say ' 5. '
125 @ 23,5 say ' 6. '
126
127
128 @ 18, 9 say sitetxt1
129 @ 19, 9 say sitetxt2
130 @ 20, 9 say sitetxt3
131 @ 21, 9 say sitetxt4
132 @ 22, 9 say sitetxt5
133 @ 23, 9 say sitetxt6
134
135 choice = 0
136 @ 24, 1 say 'Enter your selection (1-6):'
137 set color to w+/bg
138 do while choice = 0
139     @ 24,30 get choice picture '9' range 1,6
140     read
141 enddo
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

142
143 choicechr = str(choice,1)
144 site = choice
145 sitetxt = sitetxt&choicechr
146 @17,1 clear
147
148 return
149
150 *****
151 *   WsteSlct -- user selects the Waste Type.   *
152 *****
153
154 procedure wsteSlct
155 *
156
157 set color to n/bg
158 @ 17, 1 say '(B.) Select the Waste Type:'
159
160
161 @ 18,5 say ' 1. '
162 @ 19,5 say ' 2. '
163
164
165 @ 18, 9 say wtxt1
166 @ 19, 9 say wtxt2
167
168 choice = 0
169 @ 24, 1 say 'Enter your selection (1-2):'
170 set color to w+/bg
171 do while choice = 0
172     @ 24,30 get choice picture '9' range 1,2
173     read
174 enddo
175
176 choicechr = str(choice,1)
177 waste = choice
178 wtxt = wtxt&choicechr
179 @17,0 clear
180
181 return
182
183 *****
184 *   ImbSlct -- user selects the Data Requested for Immobilized Waste.   *
185 *****
186
187 procedure ImbSlct
188 *
189
190 set color to n/bg
191 @ 10,0 clear
192 frame = chr(205)
193 if (clipper)
194     @ 10,0,10,79 box frame
195 else
196     @ 10,0 to 10,79 double
197 endif
198 @ 11, 1 say '(C.) Select the Data Requested for ' + sitetxt + ':'
199
200 @ 12,4 say ' 1. '
201 @ 13,4 say ' 2. '
202 @ 14,4 say ' 3. '
203 @ 15,4 say ' 4. '
204 @ 16,4 say ' 5. '
205 @ 17,4 say ' 6. '
206 @ 18,4 say ' 7. '
207 @ 19,4 say ' 8. '
208 @ 21,4 say ' 9. '
209 @ 22,4 say '10. '
210
211 @ 12, 8 say 'Physical description of canisters and waste form'
212 @ 13, 8 say 'Number of canisters produced vs. years'
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

10

```

213 @ 14, 8 say 'Maximum radionuclide content per canister at fill time'
214 @ 15, 8 say 'Maximum radioactivity and thermal power per canister vs. decay time'
215 @ 16, 8 say "Average radioactivity per canister by year of production"
216 @ 17, 8 say "Cumulative average radioactivity and thermal power per canister by year"
217
218 @ 18, 8 say 'Chemical composition'
219 @ 19, 8 say 'Radionuclide content (grams, total curies, alpha curies, watts)'
220 @ 20, 8 say 'per canister vs. decay time'
221 @ 21, 8 say 'Photon spectrum per canister vs. decay time'
222 @ 22, 8 say 'Calculated integrated heat release per canister vs. decay time'
223
224 * Special case. If user has selected multiple sites (site=5 or 6), he can
225 * only choose query number 2
226 promptMsg = 'Enter your selection ('
227 promptCol = 31
228
229 do case
230     case site = 1
231         @ 23,1 say '*** Items 5 and 6 are not selectable for ' + sitetxt
232         promptMsg = promptMsg + '1-4,7-10):'
233         promptCol = 35
234
235     case site = 2 .or. site = 4
236         @ 23,1 say '*** Item 5 is not selectable for ' + sitetxt
237         promptMsg = promptMsg + '1-4,6-10):'
238         promptCol = 35
239
240     case site = 5
241         @ 23,1 say '*** Only Items 1-2 are selectable for ' + sitetxt
242         promptMsg = promptMsg + '1-2):'
243
244     case site = 6
245         @ 23,1 say '*** Only Items 1-2 are selectable for ' + sitetxt
246         promptMsg = promptMsg + '1-2):'
247
248     otherwise
249         promptMsg = promptMsg + '1-10):'
250 endcase
251
252 choice = 0
253 @ 24, 1 say promptMsg
254 set color to w/bg
255 do while choice = 0
256     if (site = 5)
257         @24,promptCol get choice picture '9' range 1,2
258     else
259         if (site = 6)
260             @ 24,promptCol get choice picture '9' range 1,2
261         else
262             @ 24,promptCol get choice picture '99' range 1,10
263         endif
264     endif
265     read
266     if (site = 1)
267         if (choice = 5 .or. choice = 6)
268             choice = 0
269         endif
270     else
271         if (site = 2 .or. site = 4)
272             if (choice = 5)
273                 choice = 0
274             endif
275         endif
276     endif
277 enddo
278
279 if (choice < 10)
280     choicechr = str(choice,1)
281 else
282     choicechr = str(choice,2)
283 endif

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

284 squery = choice
285 querytxt = imbqtxt&choicechr
286 hdL = hdImb&choicechr
287
288 return
289
290 *****
291 *   IntSlct -- user selects the Data Requested for Interim Waste forms.   *
292 *****
293
294 procedure intSlct
295 *
296
297 set color to n/bg
298 @ 17, 1 say '(C.) Select the Data Requested for ' + sitetxt + ':'
299
300 @ 18,5 say ' 1. '
301 @ 19,5 say ' 2. '
302 @ 20,5 say ' 3. '
303 @ 21,5 say ' 4. '
304
305
306 @ 18, 9 say intqtxt1
307 @ 19, 9 say intqtxt2
308 @ 20, 9 say intqtxt3
309 @ 21, 9 say intqtxt4
310
311 choice = 0
312
313 * Special case. If user has selected multiple sites (site = 5 or 6),
314 * only item 1 is appropriate.
315 if (site > 4)
316     @ 23,1 say '*** Only item 1 is selectable for ' + sitetxt
317     @ 24, 1 say 'Enter your selection (1-1):'
318 else
319     @ 24, 1 say 'Enter your selection (1-4):'
320 endif
321
322
323 set color to w+/bg
324 do while choice = 0
325     if (site > 4)
326         @24,30 get choice picture '9' range 1,1
327     else
328         @ 24,30 get choice picture '9' range 1,4
329     endif
330     read
331 enddo
332
333 choicechr = str(choice,1)
334 squery = choice
335 querytxt = intqtxt&choicechr
336 hdL = hdInt&choicechr
337 @17,1 clear
338
339 return
340
341 *****
342 *   ODevSlct -- user selects the output device                               *
343 *****
344
345 procedure ODevSlct
346 *
347
348 set color to n/bg
349 @ 17, 1 say '(D.) Select the output device:'
350
351
352 @ 18,5 say ' 1. '
353 @ 19,5 say ' 2. '
354 @ 20,5 say ' 3. '

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

12

```

355
356
357 @ 18, 9 say odtxt1
358 @ 19, 9 say odtxt2
359 @ 20, 9 say odtxt3
360
361 @ 22,5 say "If you select Disk File, you will be prompted for the file name."
362
363 choice = 0
364 @ 24, 1 say 'Enter your selection (1-3):'
365 set color to w+/bg
366 do while choice = 0
367     @ 24,30 get choice picture '9' range 1,3
368     read
369 enddo
370
371 choicechr = str(choice,1)
372 odtxt = odtxt&choicechr
373 oDevice = oDevice&choicechr
374
375 *** If user chose Disk File, ask him for file name.
376
377 if (choice = 3)
378     set color to w+/bg,n/w
379     oldFName = fName
380     @ 23,1 say "Enter the file name to which you would like to send data:"
381     fileOK = .f.
382     do while (.not. fileOK)
383         @ 23,60 get fName
384         read
385         if (fName <> " ")
386             do ChkFName with fileOK
387         endif
388         if (.not. fileOK)
389             @ 24,1 say 'That file name has a reserved extension. Please use a differ
ent extension.'
390         endif
391     enddo
392
393     if (file(fName))
394         @ 24,1 say fName + " already exists. Overwrite or Append? (O/A)"
395         fChoice = ' '
396         do while .not.(fChoice $ "QA")
397             @ 24,58 get fChoice picture '!'
398             read
399         enddo
400
401         if (fChoice = "O")                && Overwrite, destroy existing file
402             set alternate to &fName
403             set alternate on
404             close alternate
405         endif
406     endif
407
408     odtxt = fName
409 endif
410
411 @17,0 clear
412
413 return
414
415 *****
416 * ChkFName -- check that the user-specified filename does not have a *
417 * reserved extension of TAB, DBF, NTX. *
418 *****
419
420 procedure ChkFName
421     parameter fileOK
422
423     dotPos = AT('.',fName)
424     fileOK = .t.

```

APPENDIX A -- PROGRAM LISTINGS  
NOTZLIB.PRG

13

```

425     if (dotPos > 0)
426         extens = upper(substr(fName,dotPos+1,3))
427         do case
428             case extens = 'DBF'
429                 fileOK = .f.
430             case extens = 'NTX'
431                 fileOK = .f.
432             case extens = 'TAB'
433                 fileOK = .f.
434         endcase
435     endif
436     return
437
438     *****
439     *   IsoSlct -- user selects the Isotope for query Immobilized #5.   *
440     *****
441
442     procedure IsoSlct
443
444
445     set color to n/bg
446     @ 17, 1 say '(F.) Select the Isotope:'
447
448
449     @ 18,5 say ' 1. '
450     @ 19,5 say ' 2. '
451     @ 20,5 say ' 3. '
452     @ 21,5 say ' 4. '
453     @ 22,5 say ' 5. '
454
455     @ 18, 9 say isotxt1
456     @ 19, 9 say isotxt2
457     @ 20, 9 say isotxt3
458     @ 21, 9 say isotxt4
459     @ 22, 9 say "Summary (totals only) of all isotopes"
460
461     choice = 0
462     @ 24, 1 say 'Enter your selection (1-5):'
463     set color to w+/bg
464     do while choice = 0
465         @ 24,30 get choice picture '9' range 1,5
466         read
467     enddo
468
469     choicechr = str(choice,1)
470     misotope = choice
471     isotxt = isotxt&choicechr
472
473     *** Need to add procedure here if user selects 1, specific isotope
474     *** to let him pick the isotope
475
476     if (choice = 1)
477         do IsoSlct
478         @ 0,0 clear
479         if (clipper)
480             call ClipPop with "Main"
481             call ClipPop with "Fill Page 1"
482             call ClipPop with "Origen"
483         else
484             ?? chr(255) + chr(255) + "MAIN/"
485             ?? chr(255) + chr(255) + "FILL PAGE 1/"
486             ?? chr(255) + chr(255) + "ORIGEN/"
487         endif
488
489     *   run loader origen
490     do refresh
491     do refresh2
492     else
493         @17,1 clear
494     endif
495

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

496
497 return
498
499 *****
500 * CaseS1ct -- user selects the case (average or maximum). Only for *
501 * Origen data queries if site is WVDP. *
502 *****
503
504 procedure CaseS1ct
505 *
506
507 set color to n/bg
508 @ 17,0 clear
509 @ 17, 1 say '(E.) Select the Case: '
510
511
512 @ 18,5 say ' 1. '
513 @ 19,5 say ' 2. '
514
515 @ 18, 9 say casetxt1
516 @ 19, 9 say casetxt2
517
518 choice = 0
519 @ 24, 1 say 'Enter your selection (1-2):'
520 set color to w+/bg
521 do while choice = 0
522     @ 24,30 get choice picture '9' range 1,2
523     read
524 enddo
525
526 choicechr = str(choice,1)
527 casetxt = casetxt&choiceChr
528 tcase = choiceChr
529 @17,1 clear
530
531 return
532
533
534 *****
535 * IsolS1ct -- Select one specific isotopes from all isotopes available *
536 * at a given site. *
537 *****
538
539 PROCEDURE IsolS1ct
540 set color to n/bg
541 @ 1,0 clear
542 @ 2,28 say 'HIGH LEVEL WASTE DATA BASE'
543 @ 3,10 say 'SITE: '
544 @ 4,10 say 'WASTE TYPE:'
545 set color to w+/bg
546 @ 3,23 say sitetxt
547 @ 4,23 say wttxt
548 @ 5,10 say querytxt
549 set color to n/bg
550 @ 7,1 say 'Select the Isotope: '
551
552 maxCols = 4
553 maxRows = 22
554 * maxRows = 10
555 colCount = 1
556 column = 1
557 colSpace = 22
558 startRow = 8
559 row = startRow
560
561 isoCount = 0
562
563 mZnum = 0
564 isotxt = " "
565 numStr = " "
566 set color to n/bg
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

567 select 3
568 go top
569 firstIso = 1
570
571 iChoice = 0
572
573 c = str(site,1)
574 sitechr = sitaAbrac
575
576 seek sitechr + tcase
577 do while (siteid = sitechr .and. (tcase = ocase) .and. .not. eof())
578     isoCount = isoCount + 1
579
580     do formstr with isoCount,numStr
581     @ row,column say numstr+. '+isotope->isotope
582 *   iso&numStr = isotope->isotope
583     row = row + 1
584
585     if (row > maxRows)
586         colCount = colCount + 1
587         column = column + colSpace
588         row = startRow
589         if colCount > maxCols
590 * Need to look ahead, see if this is the last isotope. If it is,
591 * user must pick one from this last screen.
592             skip
593             if (eof() .or. siteid <> sitechr)
594                 do GetIso with .t., mZnum, isotxt
595                 return
596             else
597
598                 do GetIso with .f., mZnum, isotxt
599 *if debug2
600 * @10,10 say "Returned from GetIso. isotxt: " + isotxt
601 * wait
602 *endif
603         if iChoice <> 0
604             return
605         endif
606         skip -1
607     endif
608 endif
609 endif
610
611 skip
612 enddo
613
614 ** User must select isotope from last screen.
615 do getIso with .t. , mZnum, isotxt
616 select 1
617 return
618
619 *****
620 * FormStr -- format a character string from the given number. *
621 *****
622
623 PROCEDURE FormStr
624     parameter num,S
625
626     do case
627         case num < 10
628             S = str(num,1)
629         case num >= 10 .and. num < 100
630             S = str(num,2)
631         case num >= 100
632             S = str(num,3)
633     endcase
634     return
635
636
637 *****
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

638 *   GetIso -- get the user's selection of an isotope number.  If the   *
639 * parameter lastScr is true, then this is the last screen and the user  *
640 * must enter a selection.  If it's false, then there are more isotopes to *
641 * view, and a null answer is acceptable.                               *
642 *****
643
644 PROCEDURE GetIso
645     parameter lastScr, mZnum, isotxt
646
647 *if debug
648 *   @ 15,10 say 'in GetIso'
649 *   @ 16,10 say 'isoCount: '
650 *   @ 16,30 say isoCount
651 *   wait
652 *endif
653     numStr2 = " "
654     if .not. lastScr
655         @ 23,1 say 'Strike Enter to view more isotopes.'
656     endif
657     do FormStr with firstIso,numStr2
658     @ 24,1 say 'Enter your selection('+numStr2 + '-' + numStr + '):'
659
660     valid = .f.
661     do while (.not. valid)
662         @ 24,40 get iChoice picture '999'
663         read
664         if (iChoice <> 0)
665
666 * Check for valid input -- a number in range or just <Enter>
667         if (iChoice < firstIso .or. iChoice > isoCount)
668             line = " " + "Range is ";
669             + numStr2 + " to " + numStr
670             @0,1 say line
671         else
672             go top
673             seek sitechr + tcase
674             skip (iChoice - 1)
675             isotxt = isotope->isotope
676             mznum = isotope->znum
677             return
678         endif
679     else
680         if (.not. lastScr)
681             firstIso = isoCount + 1
682             @ 8,0 clear
683             row = startRow
684             column = 1
685             colCount = 1
686             return
687         endif
688     endif
689     enddo
690     return
691
692
693 *****
694 *   DecySlct -- user selects a set of Decay Times.                       *
695 *****
696
697 procedure DecySlct
698 *
699
700 set color to n/bg
701 @ 17, 1 say '(G.) Select the Decay Time:'
702
703
704 @ 18,5 say ' 1. '
705 @ 19,5 say ' 2. '
706 @ 20,5 say ' 3. '
707 @ 21,5 say ' 4. '
708 @ 22,5 say ' 5. '
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

709
710
711 @ 18, 9 say decytxt1
712 @ 19, 9 say decytxt2
713 @ 20, 9 say "200, 300, 350, 500, 1000, 1050 years"
714 @ 21, 9 say decytxt4
715 @ 22, 9 say decytxt5
716
717 choice = 0
718 @ 24, 1 say 'Enter your selection (1-5):'
719 set color to w+/bg
720 do while choice = 0
721     @ 24,30 get choice picture '9' range 1,5
722     read
723 enddo
724
725 choicechr = str(choice,1)
726 decay = choice
727 decytxt = decytxt&choicechr
728 @17,1 clear
729
730 return
731
732
733 *****
734 *   UnitSlct -- user selects the Waste Type. *
735 *****
736
737 procedure UnitSlct
738 *
739
740 set color to n/bg
741 @ 17, 1 say '(E.) Select the Units:'
742
743
744 @ 18,5 say ' 1. '
745 @ 19,5 say ' 2. '
746 @ 20,5 say ' 3. '
747 @ 21,5 say ' 4. '
748
749
750 @ 18, 9 say unitxt1
751 @ 19, 9 say unitxt2
752 @ 20, 9 say unitxt3
753 @ 21, 9 say unitxt4
754
755 choice = 0
756 @ 24, 1 say 'Enter your selection (1-4):'
757 set color to w+/bg
758 do while choice = 0
759     @ 24,30 get choice picture '9' range 1,4
760     read
761 enddo
762
763 choicechr = str(choice,1)
764 unit = choice
765 unitxt = unitxt&choicechr
766 @17,1 clear
767
768 return
769
770 *****
771 *   PhotSlct -- user selects the group for the photon data. *
772 *****
773
774 procedure PhotSlct
775 *
776
777 set color to n/bg
778 @ 17, 1 say '(F.) Select the Isotopes for the photon data:'
779

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

18

```

780
781 @ 18,5 say ' 1. '
782 @ 19,5 say ' 2. '
783 @ 20,5 say ' 3. '
784 @ 21,5 say ' 4. '
785
786 @ 18, 9 say photxt1
787 @ 19, 9 say photxt2
788 @ 20, 9 say photxt3
789 @ 21, 9 say photxt4
790
791 choice = 0
792 @ 24, 1 say 'Enter your selection (1-4):'
793 set color to w+/bg
794 do while choice = 0
795     @ 24,30 get choice picture '9' range 1,4
796     read
797 enddo
798
799 choicechr = str(choice,1)
800 photiso = choice
801 photxt = photxt&choicechr
802
803 @17,1 clear
804 return
805
806
807 *****
808 *   RunRpt -- run the report according to all the options the user has   *
809 * selected.                                                                *
810 *   For most cases, the "report" is a previously prepared text file that *
811 * can be displayed on the screen using the List utility, or printed using *
812 * the Fprint utility. In the case of origen data, the report must be    *
813 * extracted from the data base and formatted as a text file first, then  *
814 * displayed as the others are.                                           *
815 *****
816
817 procedure RunRpt
818     parameter fName
819
820 if (waste = 1)
821     W = 'MB'
822 else
823     W = 'NT'
824     if (debug)
825         @ 2,10 say "W: " + W
826     endif
827 endif
828
829
830 if (squery < 10)
831     Q = str(squery,1)
832 else
833     Q = str(squery,2)
834 endif
835 c = str(site,1)
836 S = trim(siteAbr&c)
837
838 slctRpt = W + Q + S + ".tab"
839
840 if (debug2)
841     @ 3,10 say "slctRpt: " + slctRpt
842     wait "" to dummy
843 endif
844
845 if ((waste = 1) .and.(squery = 8))
846 ** If a floppy disk system, have to ask user to put in the diskette with
847 ** Origen data
848     if (floppy)
849         set color to n/bg
850         f = drive + "data1.mrk"

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

19

```

851     do while (.not. file(f))
852         @ 21,10 say "Please put the diskette labeled HLW Data #1 into drive B."
853         @ 22,25 say "Strike any key to continue."
854         wait "" to dummy
855     enddo
856 endif
857
858 if (squery = 8)
859     select 1
860     do NuclRpt with slctRpt
861 endif
862
863 else                                     %% Other queries require floppy #2
864
865 *** If floppy based system, check that the Interim Forms data diskette, if
866 *** necessary, is in the drive.
867
868     slctRpt = drive + slctRpt
869     if (floppy)
870         f = drive + "data2.mrk"
871         do while (.not. file(f))
872             set color to n/bg
873             @ 21,10 say "Please put the diskette labeled HLW Data #2 into drive B."
874             @ 22,25 say "Strike any key to continue."
875             wait "" to dummy
876         enddo
877     endif
878
879     if (squery = 9)
880         do PhotRpt with slctRpt
881     endif
882 endif
883
884 ** Special case. If report is number 2 for immobilized for Defense or All
885 ** sites, we need additional width.
886 if (slctRpt = "MB2ALL")
887     lineLen = "132"
888 else
889     if (slctRpt = "MB2DEF")
890         saveHdL = hdL
891         lineLen = "90"
892     else
893         lineLen = "80"
894     endif
895 endif
896
897 if oDevice = 'Screen'
898     if (debug2)
899         @ 21, 0 clear
900         @ 21,10 say "hdL:  "
901         @ 21,25 say hdL
902         Wait "" to dummy
903     endif
904     run hdlsplay &slctRpt -h &hdL -l &lineLen
905 else                                     %% print or file
906     clear
907     set color to n/gb
908     @ 1,28 say "HIGH LEVEL WASTE DATABASE"
909     if (oDevice = 'Print')
910         @ 10,20 say 'Your report is being sent to the Printer.'
911         run dbprint &slctRpt -h &hdL -l &lineLen > nul
912     endif
913     if (oDevice = 'File')
914         @ 10,20 say 'Your report is being sent to ' + fName
915         run copy &fName + &slctRpt &fName > nul
916     endif
917     @ 12,20 say "Strike any key to continue."
918     wait "" to dummy
919 endif
920
921 return
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

922
923 *****
924 * NuclRpt -- format report for immobilized query #8 -- nuclide content *
925 * vs. years. *
926 * The report varies according to the isotope(s) the user has selected. *
927 * If he has selected a single isotope, the value(s) for that isotope are *
928 * retrieved from the origin data base. If he has selected a group of *
929 * isotopes, such as all activation products, the entire group must be *
930 * retrieved. *
931 * The data is then sent to a text file. This file can then *
932 * be browsed using the hdisplay utility, as are the other reports. *
933 *****
934
935 procedure NuclRpt
936     parameter slctRpt
937
938     select 1
939     * Initialize time and value variables
940         tTotal1 = 0
941         tTotal2 = 0
942         tTotal3 = 0
943         tTotal4 = 0
944         tTotal5 = 0
945         tTotal6 = 0
946         val1 = " "
947         val2 = " "
948         val3 = " "
949         val4 = " "
950         val5 = " "
951         val6 = " "
952         dcount = 0
953
954     do DecaySet
955
956     pageRcws = 60
957     do case
958         case unit = 1
959             unitStr = "g"
960         case unit = 2
961             unitStr = "c"
962         case unit = 3
963             unitStr = "A"
964         case unit = 4
965             unitStr = "W"
966     endcase
967
968     c = str(site,1)
969
970     set alternate to &slctRpt
971     set console off
972     set alternate on
973
974     blanks = " "
975     htxt = "ISOTOPE(S): " + isotxt
976     l = len(htxt)
977     htxt = htxt + substr(blanks,1,48-l)
978     htxt = htxt + "UNITS: " + unitxt
979     do Heading with htxt,"O"
980
981     c = str(site,1)
982     sitechr = siteAbr&c
983
984     do case
985         case misotope = 1
986             do Iso1Rpt
987         case misotope = 2
988             do IsoGRpt with "V"           && Activation products
989         case misotope = 3
990             do IsoGRpt with "A"           && Actinides and daughters
991         case misotope = 4
992             do IsoGRpt with "F"           && Fission products
    
```

APPENDIX A -- PROGRAM LISTINGS  
NOTZLIB.PRG

21

```

993         case misotope = 5
994             do IsoTRpt
995         endcase
996
997         close alternate
998         set alternate off
999         set console on
1,000        return
1,001
1,002 *****
1,003 *   Heading -- Print heading for origen query report.           *
1,004 *****
1,005 procedure Heading
1,006     parameter Str1,rpt
1,007
1,008 * center the subject of the report
1,009 blanks = "
1,010 startpos = (80 - len(querytxt))/2
1,011 line = substr(blanks,1,startpos) + upper(querytxt)
1,012 ?? line
1,013 line = 'SITE: ' + sitetxt
1,014 line = line + space(4) + upper(trim(casetxt))
1,015 if (site = 1)
1,016     line = line + " CASE"
1,017 endif
1,018
1,019 l = len(line)
1,020 startPos = (80-l)/2
1,021 ? space(startPos) + line
1,022
1,023 line = '          WASTE TYPE: ' + WDXT
1,024 l = len(line)
1,025 startPos = (80-l)/2
1,026 ? space(startPos) + line
1,027
1,028 ? Str1
1,029
1,030 * adjust heading for origen report or photon report
1,031 if rpt = "O"
1,032     line = " ISOTOPE      "
1,033 else
1,034     line = " EMEAN        "
1,035 endif
1,036
1,037 i = 1
1,038 do while (i <= dcount)
1,039     c = str(i,1)
1,040     s = substr(val&c,3)
1,041     line = line + s + "YR"
1,042     l = 12 - (len(s) + 2)
1,043     line = line + substr(blanks,1,l)
1,044     i = i + 1
1,045 enddo
1,046 ? trim(line)
1,047 ** Write a heading separator line
1,048 *? "*****"
1,049 return
1,050
1,051
1,052 *****
1,053 *   IsoGRpt -- Report on a group of isotopes -- Actinides, fission products*
1,054 *****
1,055 procedure IsoGRpt
1,056     parameters group
1,057     key = sitechr + tcase + unitStr + group
1,058     set exact off
1,059     seek key
1,060     if (.not. found())
1,061 ? "No isotopes were found in this group with significant values."
1,062     return
1,063     endif

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

1,064
1,065 do while (siteid = sitechr .and. (tcase = ocase) .and. (unitStr = units);
1,066     .and. (group = activation) .and. .not. eof())
1,067     line = origen->isotope + " " + sval1 + " " + sval2 + " " ;
1,068     +sval3 + " " + sval4 + " " + sval5 + " " + sval6
1,069     ? trim(line)
1,070
1,071     skip
1,072 enddo
1,073 return
1,074
1,075 *****
1,076 * IsoIRpt -- Report values for one specific isotope. *
1,077 *****
1,078 procedure IsoIRpt
1,079     key = sitechr + tcase + unitStr + "A" + str(mznum,3) + isotxt
1,080
1,081     seek key
1,082     if (.not. found())
1,083         key = sitechr + tcase + unitStr + "F" + str(mznum,3) + isotxt
1,084         seek key
1,085         if (.not. found())
1,086             key = sitechr + tcase + unitStr + "V" + str(mznum,3) + isotxt
1,087             seek key
1,088             if (.not. found())
1,089                 ? "The isotope was not found for these units with significant value."
1,090             return
1,091         endif
1,092     endif
1,093 endif
1,094
1,095     line = substr(isotxt,1,8) + " " + sval1 + " " + sval2 + " " ;
1,096     +sval3 + " " + sval4 + " " + sval5 + " " + sval6
1,097     ? trim(line)
1,098     return
1,099
1,100
1,101 *****
1,102 * IsoTRpt -- Report of totals for all isotopes, with subtotals for the *
1,103 * different groups. *
1,104 *****
1,105 procedure IsoTRpt
1,106
1,107     gTotal1 = 0
1,108     gTotal2 = 0
1,109     gTotal3 = 0
1,110     gTotal4 = 0
1,111     gTotal5 = 0
1,112     gTotal6 = 0
1,113
1,114 * Use the origen total table
1,115 select 4
1,116 ? "Activation Products"
1,117 do IsoSub with "V" && Find and print Act. Product totals
1,118 ? " "
1,119
1,120 ? "Actinides and daughters"
1,121 do IsoSub with "A"
1,122 ? " "
1,123
1,124 ? "Fission Products"
1,125 do IsoSub with "F"
1,126 ? " "
1,127
1,128 ? "Total of all Groups"
1,129 do IsoSub with "T"
1,130
1,131 select 1
1,132 return
1,133
1,134 *****
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

1,135 * IsoSub -- Find a particular record in the Oritot table and print it. *
1,136 *****
1,137 procedure IsoSub
1,138     parameter group
1,139
1,140 key = sitechr + tcase + unitStr + group
1,141 seek key
1,142 if (.not. found())
1,143 * @ 23,0 clear
1,144 * set console on
1,145 * @ 23,10 say "Can't find oritot data for key: " + key
1,146 * wait " " to dummy
1,147     return
1,148 endif
1,149
1,150 i = 1
1,151 line = " "
1,152 do while (i <= dcount)
1,153     c = str(i,1)
1,154     m = "val" + c
1,155     j = &m
1,156     q = &j
1,157     line = line + q + " "
1,158     i = i + 1
1,159 enddo
1,160
1,161 ? trim(line)
1,162 return
1,163
1,164
1,165 *****
1,166 * Num2Str -- change a number into a string of the form 9.999E+99 *
1,167 *****
1,168 procedure Num2Str
1,169     parameters num,string
1,170
1,171
1,172     if (num = 0.0)
1,173         string = "0.000E+01"
1,174         return
1,175     endif
1,176
1,177     if num >= 1.0
1,178         expon = 0
1,179         do while (10**(expon+1) < num)
1,180             expon = expon + 1
1,181         enddo
1,182     else
1,183         expon = -1
1,184         do while (10**(expon) > num)
1,185             expon = expon - 1
1,186         enddo
1,187     endif
1,188
1,189     if expon >= 0
1,190         sign = '+'
1,191     else
1,192         sign = '-'
1,193     endif
1,194
1,195     absExpon = abs(expon)
1,196     if absExpon < 10
1,197         expstr = '0' + str(absExpon,1)
1,198     else
1,199         expstr = str(absExpon,2)
1,200     endif
1,201
1,202     n = num/(10**expon)
1,203     saveN = n
1,204     n = round(n,3)
1,205     string = str(n,5,3) + 'E' + sign + expstr

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

1,206
1,207 *@ 21,10 say "number before rounding: "
1,208 *@ 21,40 say saveN
1,209 *@ 22,10 say "Number after rounding: "
1,210 *@ 22,40 say n
1,211 *@ 23,10 say "String: " + string
1,212 *wait "" to dummy
1,213     return
1,214
1,215 *****
1,216 *   Str2Num -- Convert a string to a number.  The string is in the form *
1,217 *   9.999E+99. *
1,218 *****
1,219 procedure Str2Num
1,220     parameters string,num
1,221     num = val(substr(string,1,5)) * 10**(val(substr(string,7,3)))
1,222     return
1,223
1,224
1,225 *****
1,226 *   DecaySet - Set the decay times given the set number. *
1,227 *****
1,228 procedure DecaySet
1,229
1,230     do case
1,231         case decay = 1
1,232             dcount = 5
1,233             val1 = "YR0"
1,234             val2 = "YR1"
1,235             val3 = "YR2"
1,236             val4 = "YR5"
1,237             val5 = "YR10"
1,238             val6 = "blank"
1,239
1,240         case decay = 2
1,241             dcount = 5
1,242             val1 = "YR15"
1,243             val2 = "YR20"
1,244             val3 = "YR30"
1,245             val4 = "YR50"
1,246             val5 = "YR100"
1,247             val6 = "blank"
1,248
1,249         case decay = 3
1,250             dcount = 6
1,251             val1 = "YR200"
1,252             val2 = "YR300"
1,253             val3 = "YR350"
1,254             val4 = "YR500"
1,255             val5 = "YR1000"
1,256             val6 = "YR1050"
1,257
1,258         case decay = 4
1,259             dcount = 4
1,260             val1 = "YR2K"
1,261             val2 = "YR5K"
1,262             val3 = "YR10K"
1,263             val4 = "YR20K"
1,264             val5 = "blank"
1,265             val6 = "blank"
1,266
1,267         case decay = 5
1,268             dcount = 4
1,269             val1 = "YR50K"
1,270             val2 = "YR100K"
1,271             val3 = "YR500K"
1,272             val4 = "YR1000K"
1,273             val5 = "blank"
1,274             val6 = "blank"
1,275     endcase
1,276     return
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

1,277
1,278 *****
1,279 * PhotRpt -- Report of photon energies. *
1,280 *****
1,281 procedure PhotRpt
1,282     parameter slctRpt
1,283
1,284     select 2
1,285     use &drive.photon index &drive.photon
1,286
1,287 * Initialize variables
1,288     val1 = " "
1,289     val2 = " "
1,290     val3 = " "
1,291     val4 = " "
1,292     val5 = " "
1,293     val6 = " "
1,294     dcount = 0
1,295
1,296     do DecaySet
1,297
1,298     pageRows = 60
1,299
1,300     set alternate to &slctRpt
1,301     set console off
1,302     set alternate on
1,303
1,304     blanks = " "
1,305     htxt = "ISOTOPES: " + photxt
1,306     l = len(htxt)
1,307     htxt = htxt + substr(blanks,1,48-l)
1,308     htxt = htxt + "UNITS: Photons/sec"
1,309
1,310     do Heading with htxt,"P"
1,311
1,312     c = str(site,1)
1,313     sitechr = siteAbr&c
1,314
1,315     do case
1,316         case photiso = 1
1,317             do PhotGRpt with "V"
1,318         case photiso = 2
1,319             do PhotGRpt with "A"
1,320         case photiso = 3
1,321             do PhotGRpt with "F"
1,322         case photiso = 4
1,323             do PhotGRpt with "T"
1,324     endcase
1,325
1,326     close alternate
1,327     set alternate off
1,328     set console on
1,329     return
1,330
1,331
1,332 *****
1,333 * PhotGRpt -- Print the photon energy table for the specified group of *
1,334 * isotopes -either actinides or fission products. *
1,335 *****
1,336 procedure PhotGRpt
1,337     parameter group
1,338
1,339     n = 1
1,340     s = " "
1,341     num = 0
1,342     set exact off
1,343
1,344     key1 = sitechr + tcase + group
1,345     do while (n <= 18)
1,346         do FormStr with n,s
1,347         key = key1 + &meanfs
    
```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

1,348
1,349 seek key
1,350 if (.not. found())
1,351     ? "No photon records for this site and isotope group."
1,352     return
1,353 endif
1,354
1,355 line = photon->spectrum + sval1 + " " + sval2 + " ";
1,356     + sval3 + " " + sval4 + " " + sval5 + " " + sval6
1,357     ? trim(line)
1,358
1,359     n = n + 1
1,360 enddo
1,361 return
1,362
1,363
1,364 *****
1,365 * GetQuery -- Get the query (Data Selected) from the user. Different *
1,366 * selections are available according to the waste type selected. *
1,367 *****
1,368
1,369 procedure GetQuery
1,370     if (waste = 1)
1,371         do imbslct
1,372     else
1,373         do intslct
1,374     endif
1,375
1,376
1,377     if (squery = 8 .and. waste = 1)
1,378         do seq2
1,379     else
1,380         if (squery = 9 .and. waste = 1)
1,381             do PhotSeq
1,382         else
1,383             if (waste = 1)
1,384                 @0,0 clear
1,385                 if (clipper)
1,386                     call ClipPop with "Main"
1,387                 else
1,388                     ?? chr(255) + chr(255) + "MAIN/"
1,389                 endif
1,390
1,391                 do refresh
1,392
1,393             else
1,394                 do writeIt with "D", querytxt
1,395             endif
1,396             photflag = .f.
1,397             orgflag = .f.
1,398         endif
1,399     endif
1,400     return
1,401
1,402
1,403 *****
1,404 * Refresh -- Write variable values into the screen. *
1,405 *****
1,406
1,407 procedure REFRESH
1,408     set color to w+/bg
1,409     @ 3,21 say sitetxt
1,410     @ 4,21 say wtxt
1,411     @ 6,28 say querytxt
1,412     @ 8,18 say odtxt
1,413     @ 15,23 say FINALANS
1,414     return
1,415
1,416 *****
1,417 * Refresh2 -- Write the additional variable values into the screen. *
1,418 *****

```

APPENDIX A -- PROGRAM LISTINGS  
 NOTZLIB.PRG

```

1,419
1,420 procedure REFRESH2
1,421     @ 8,43 say casetxt
1,422     @ 9,47 say isctxt
1,423     @ 10,49 say decytxt
1,424     @ 11,44 say unitxt
1,425     return
1,426
1,427
1,428 *****
1,429 * writeIt -- Write a specific variable value into the screen. The first*
1,430 * parameter, code, indicates which variable to write. *
1,431 * The purpose of this procedure is to speed up the program by rewriting *
1,432 * only the variable that has changed in value, rather than re-writing all *
1,433 * values. *
1,434 *****
1,435
1,436 procedure WriteIt
1,437     parameters code, value
1,438     set color to w+/bg
1,439     if (debug)
1,440         @ 21,0 clear
1,441         @ 21,10 say "In WriteIt. code: " + code
1,442         @ 22,10 say "value: " + value
1,443         wait "" to dummy
1,444     endif
1,445     do case
1,446         case code = "S"
1,447             @ 3,21 say value
1,448         case code = "W"
1,449             @ 4,21 say value
1,450         case code = "D"
1,451             @ 6,28 say value
1,452         case code = "O"
1,453             @ 8,18 say value
1,454         case code = "C"                && Case
1,455             @ 8, 43 say value
1,456         case code = "I"
1,457             @ 9,47 say value
1,458         case code = "T"
1,459             @ 10,49 say value
1,460         case code = "U"
1,461             @ 11,44 say value
1,462     endcase
1,463     return
    
```

APPENDIX A -- PROGRAM LISTINGS  
START.PRG

```

1  set talk off
2  set heading off
3  set bell off
4  set confirm off
5  set exact off
6  set status off
7  set scoreboard off
8
9  * public PROGDIR,DBFDIR,NDXDIR,TDBFDIR
10
11 public site, sitetxt, waste, wtxt, squery, querytxt, tcase, casetxt
12 public misotope, isotxt, mznum, decay, decytxt, unit, unitxt, photiso, photxt
13 public orgflag, photflag, datadir, formfeed, dcount, unitStr, odtxt, oDevice
14 public debug,debug2,dummy,clipper
15 public row, val1,val2,val3,val4,val5,val6,blank, hdL, floppy, intDisk
16 set color to n/bg
17 clear
18
19 *** Check for floppy or hard disk system
20 if file ("Flopdisk.mrk")
21     drive = "B:"
22     floppy = .t.
23 ** Check that they start with immobilized data in B for floppy systems.
24     do while file ("b:ntlwvdp.tab")
25         @ 10,10 say "Please put the diskette labeled HLW Data #1 into drive B."
26         @ 12,25 say "Strike any key to continue."
27         wait "" to dummy
28     enddo
29     intDisk = .f.    && Interim forms data floppy is not in drive
30 else
31     drive = ""
32     floppy = .f.
33 endif
34
35 debug = .f.
36 debug2 = .f.
37 * PROGDIR = trim(prgpath)
38 * DBFDIR = trim(dbfpath)
39 * NDXDIR = trim(ntxpath)
40 * TDBFDIR = trim(totpath)
41
42 set procedure to notzlib
43
44 *****
45 * dBase III Main Menu for High Level Waste data base project.      *
46 * For Karl Notz of Chemical Technology Division.                    *
47 * Based on the NWTSP data base system written by Tim Rhyne.        *
48 *   Written by Kathy Jones 9/86.                                    *
49 *   This module displays an introductory message, sets up system  *
50 * variables, then passes control to Control.prg, which will accept *
51 * selections from the user.                                         *
52 *****
53
54 ANSWER = 0
55
56
57 @ 0,0 CLEAR
58 if (clipper)
59     call clippop with "IntroScr"
60 else
61     ?? chr(255) + chr(255) + "INTROSCR/"
62 endif
63
64 SET COLOR TO 7/0, 0/7
65
66 ** Set number of heading lines for different reports
67
68 hdInt1 = '8'
69 hdInt2 = '7'
70 hdInt3 = '4'
71 hdInt4 = '5'

```

APPENDIX A -- PROGRAM LISTINGS  
START.PRG

29

```

72
73 hdImb1 = '6'
74 hdImb2 = '7'
75 hdImb3 = '4'
76 hdImb4 = '6'
77 hdImb5 = '6'
78 hdImb6 = '8'
79 hdImb7 = '6'
80 hdImb8 = '5'
81 hdImb9 = '5'
82 hdImb10 = '5'
83
84 sitetxt1 = "West Valley Demonstration Project"
85 sitetxt2 = "Savannah River Plant"
86 sitetxt3 = "Hanford"
87 sitetxt4 = "Idaho National Engineering Laboratory"
88 sitetxt5 = "All Defense Sites (SRP,HANF,INEL)"
89 sitetxt6 = "All Sites"
90
91 siteAbr1 = "WVDP"
92 siteAbr2 = "SRP"
93 siteAbr3 = "HANF"
94 siteAbr4 = "INEL"
95 siteAbr5 = "DEF"
96 siteAbr6 = "ALL"
97
98 wtxt1 = "Immobilized (in canisters)"
99 wtxt2 = "Interim (liquid, sludge, etc.)"
100
101 imbqtxt1 = "Physical description of canisters and waste form"
102 imbqtxt2 = "Number of canisters produced vs. years"
103 imbqtxt3 = "Max. Radionuclide content per canister at fill time"
104 imbqtxt4 = "Max. Radioactivity and thermal power vs. decay time"
105 imbqtxt5 = "Avg. radioactivity per canister by year of product."
106 imbqtxt6 = "Cum. avg. radioactivity and thermal power by year"
107 imbqtxt7 = "Chemical composition"
108 imbqtxt8 = "Radionuclide content per canister vs. decay time"
109 imbqtxt9 = "Photon spectrum per canister vs. decay time"
110 imbqtxt10 = "Calculated integrated heat release vs. decay time"
111
112 odtxt1 = "Screen"
113 odtxt2 = "Print"
114 odtxt3 = "Disk File"
115 oDevice1 = "Screen"
116 oDevice2 = "Print"
117 oDevice3 = "File"
118
119 intqtxt1 = "Volume of waste vs. years"
120 intqtxt2 = "Radioactivity and thermal power vs. years"
121 intqtxt3 = "Chemical composition"
122 intqtxt4 = "Radionuclide composition"
123
124 casetxt1 = "Average Radioactivity"
125 casetxt2 = "Maximum Radioactivity"
126 nocstxt = "Maximum Radioactivity (one case)"
127
128 isotxt1 = "One specific isotope"
129 isotxt2 = "All activation products"
130 isotxt3 = "All actinides and daughters"
131 isotxt4 = "All fission products"
132 isotxt5 = "Summary of all isotopes"
133
134 decytxt1 = "0, 1, 2, 5, 10 years"
135 decytxt2 = "15, 20, 30, 50, 100 years"
136 decytxt3 = "200,300,350,500,1000,1050"
137 decytxt4 = "2K, 5K, 10K, 20K years"
138 decytxt5 = "50K, 100K, 500K, 1000K years"
139
140 unitxt1 = "Grams"
141 unitxt2 = "Total curies"
142 unitxt3 = "Alpha curies"

```

APPENDIX A -- PROGRAM LISTINGS  
START.PRG

30

```
143 unitxt4 = "Total watts "
144
145 photxt1 = "Activation products  "
146 photxt2 = "Actinides and daughters"
147 photxt3 = "Fission products    "
148 photxt4 = "Total isotopes      "
149
150 emean1 = "1.000E-02"
151 emean2 = "2.500E-02"
152 emean3 = "3.750E-02"
153 emean4 = "5.750E-02"
154 emean5 = "8.500E-02"
155 emean6 = "1.250E-01"
156 emean7 = "2.250E-01"
157 emean8 = "3.750E-01"
158 emean9 = "5.750E-01"
159 emean10 = "8.500E-01"
160 emean11 = "1.250E+00"
161 emean12 = "1.750E+00"
162 emean13 = "2.250E+00"
163 emean14 = "2.750E+00"
164 emean15 = "3.500E+00"
165 emean16 = "5.000E+00"
166 emean17 = "7.000E+00"
167 emean18 = "9.500E+00"
168
169 formfeed = ""          && Actually a control-L; wouldn't print
170 blank = " "
171 set console off
172 wait
173 set console on
174 do control
```

INTERNAL DISTRIBUTION

- |                       |                                   |
|-----------------------|-----------------------------------|
| 1. R. J. Andermann    | 25. W. J. Reich                   |
| 2. R. C. Ashline      | 26. R. Salmon                     |
| 3. J. M. Begovich     | 27. S. N. Storch                  |
| 4. A. G. Croff        | 28. M. G. Stewart                 |
| 5. R. M. Gove         | 29. T. D. Welch                   |
| 6. V. T. Hinkel       | 30. Central Research Library      |
| 7. D. S. Joy          | 31. Laboratory Records Department |
| 8. E. K. Johnson      | 32. Laboratory Records, ORNL RC   |
| 9. P. E. Johnson      | 33. ORNL Patent Section           |
| 10. J. A. Klein       | 34. Y-12 Technical Library        |
| 11. S. B. Ludwig      |                                   |
| 12. A. P. Malinauskas |                                   |
| 13. W. C. McClain     |                                   |
| 14. J. W. Nehls       |                                   |
| 15 - 24. K. J. Notz   |                                   |

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. Gollhofer
- 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