CNWRA 94-016

INVENT: A MODULE FOR THE CALCULATION OF RADIONUCLIDE INVENTORIES, SOFTWARE DESCRIPTION, AND USER GUIDE

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-93-005

Prepared by

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

July 1994

CNWRA 94-016

INVENT: A MODULE FOR THE CALCULATION OF RADIONUCLIDE INVENTORIES, SOFTWARE DESCRIPTION, AND USER GUIDE

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-93-005

Prepared by

A.S. Lozano
H. Karimi
J.P. Cornelius
R.D. Manteufel
R.W. Janetzke

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

July 1994

ABSTRACT

This report describes the INVENT module and a set of FORTRAN subroutines which were developed for use with the Nuclear Regulatory Commission (NRC) Total-System Performance Assessment (TPA) computer code. The report describes the structure of the INVENT module and provides instructions for its general use. The new module is a unique integration of: (i) a graphical user interface (GUI) development program, (ii) a relational database program, and (iii) a computer code that calculates the radionuclide inventory for spent fuel. The basic output of the INVENT module is a set of radionuclides and their associated time-dependent curie content per metric ton of heavy metal for spent fuel. Because the radionuclide inventories are repetitively required in the TPA code, a set of simplified FORTRAN subroutines have been developed which allow the efficient storage and retrieval of the inventories of a few key radionuclides.

CONTENTS

CONTENTS (Cont'd)

FIGURES

TABLES

ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
BWR	Boiling Water Reactor
CFR	Code of Federal Regulations
Ci	Curie
CM	Consequence Module
CNWRA	Center for Nuclear Waste Regulatory Analyses
DOE	U.S. Department of Energy
EPA	Environmental Protection Agency
Exec	Executive Module
GUI	Graphical User Interface
GWTT	Groundwater Travel Time
HLW	High-Level Waste
IPA	Iterative Performance Assessment
MTU	Metric Ton (= 1,000 kg) of Uranium
MTIHM	Metric Ton of Initial Heavy Metal
MWd	Mega Watt days
NRC	Nuclear Regulatory Commission
NWPA	Nuclear Waste Policy Act
ORNL	Oak Ridge National Laboratory
PA	Performance Assessment
PWR	Pressurized Water Reactor
SNL	Sandia National Laboratories
TPA	Total-System Performance Assessment Code
TSPA	Total-System Performance Assessment

ACKNOWLEDGMENTS

This report was prepared to document work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Nuclear Regulatory Commission (NRC) under Contract No. NRC-02-93-005. The activities reported here were performed on behalf of the NRC Office of Nuclear Regulatory Research (RES), Division of Regulatory Applications. This report is an independent product of the CNWRA and does not necessarily reflect the views or regulatory positions of the NRC.

The authors wish to thank R.G. Baca, J.P. Hageman, R.W. Mielke, and R.L. Marshall for their technical reviews, and B. Sagar for his programmatic review.

The following trademarks and products are discussed in this report:

- GALAXY is a trademark of Visix Software, Inc.
- ORACLE is a trademark of Oracle Corporation
- SUN Workstation is a trademark of Sun Microsystems, Inc.
- UNIX is a trademark of UNIX System Laboratories, Inc.
- X Window System is a product of the Massachusetts Institute of Technology

Quality of Data, Analyses, and Code Development

Data: There is no CNWRA-generated original data contained in this report. Sources for other data should be consulted for determining the level of quality for those data.

Analyses and Codes: The INVENT module is controlled under the CNWRA's Software Configuration Procedure. One requirement of this procedure is this User Guide.

FOREWORD

In accordance with the provisions of the Nuclear Waste Policy Act of 1982 (see Nuclear Regulatory Commission, 1991), the Nuclear Regulatory Commission (NRC) has the responsibility of evaluating and granting a license for the first and subsequent, if any, geological repositories for high-level radioactive waste (HLW). This act was amended in 1987 to designate one site in the unsaturated region of tuffaceous rocks of Yucca Mountain in southern Nevada for detailed characterization. The Center for Nuclear Waste Regulatory Analyses (CNWRA) at Southwest Research Institute is a Federally Funded Research and Development Center created to support the NRC in its mission of evaluating and licensing the proposed HLW repository. To meet its licensing function, the NRC will review the application submitted by the U.S. Department of Energy (DOE). One critical section of the license application will deal with the assessment of the future performance of the repository system, which has to meet certain minimum standards established by regulations.

The INVENT module, described in this report, is designed to function as a component of the Total-System Performance Assessment (TPA) computer code. This module was developed by integrating three general computer programs: (i) GALAXY, a graphical user interface (GUI) development software package, (ii) ORACLE, a relational database program, and (iii) ORIGEN2, a radionuclide inventory model. The INVENT module was developed to facilitate the calculation of radionuclide inventories for spent nuclear fuel. In addition to the stand-alone INVENT module, a set of FORTRAN subroutines have been developed for the efficient storage and repetitive retrieval of the inventories of a few key radionuclides. These FORTRAN subroutines are designed to be used in the TPA code.

1 INTRODUCTION

1.1 REGULATORY AND TECHNICAL BACKGROUND

The Nuclear Regulatory Commission (NRC), in conjunction with the Center for Nuclear Waste Regulatory Analyses (CNWRA), is developing an independent performance assessment (PA) capability to:

- Aid in developing an independent understanding of the processes, conditions, and events important to predicting long-term repository performance and their relative significance in such predictions
- Provide an independent capability for reviewing the U.S. Department of Energy (DOE) demonstrations of compliance with the overall system and subsystem performance objectives
- Aid in the detailed technical review of the DOE iterative total-system and subsystem PAs
- Contribute to the development of guidance to the DOE on the adequacy of site characterization data and repository design, with respect to demonstrating compliance with the regulations

This independent PA capability is designed to quantitatively calculate estimates of repository performance. To make predictions of performance and comparisons with regulatory performance measures, the Total-System Performance Assessment (TPA) computer code has been developed to provide computational algorithms for estimating values of various performance measures [See Sagar and Janetzke (1993) for the description of the TPA code]. To estimate the performance measures, the TPA computer code contains a set of Consequence Modules (CMs) that are independent computational units.

Regulations applicable to the performance objectives of high-level radioactive waste (HLW) geological repository were promulgated by the NRC in 10 CFR Part 60—Disposal of High-Level Radioactive Wastes in Geologic Repositories. Two sections of 10 CFR Part 60 pertain specifically to post-closure performance. These sections include 10 CFR 60.112—Overall System Performance Objective for the Geologic Repository After Permanent Closure; and 10 CFR 60.113—Performance of Particular Barriers After Permanent Closure. 10 CFR 60.112 makes reference to satisfying the generally applicable environmental standards for radioactivity established by the U.S. Environmental Protection Agency (EPA). These environmental standards referred to in 10 CFR 60.112 were promulgated by the EPA in 40 CFR Part 191 in 1985. However, on litigation, certain provisions of these standards were remanded by a federal court. Proposed revisions of 40 CFR Part 191 were under review in early 1993. In late 1992, the U.S. Congress enacted a new law known as the Energy Policy Act, according to which the EPA will develop standards applicable specifically to Yucca Mountain that may be different from those in 40 CFR Part 191.

Three different performance measures are used in 40 CFR Part 191. These measures are: (i) release of radioactivity over the entire accessible environment boundary (integrated over areal space) cumulated over a 10,000-yr period (integrated over time) after closure must not exceed specific limits at specified levels (40 CFR 191.13—Containment Requirements), where the preferred method of representing this performance measure is through a Complementary Cumulative (Probability) Distribution

Function (CCDF); (ii) dose to humans in the first 1,000 years after repository closure must not exceed specified limit (40 CFR 191.15—Individual Protection Requirements), this requirement is deterministic; and (iii) concentration of alpha-, beta-, and gamma-emitting radionuclides must not exceed specified limits (40 CFR 191.16—Groundwater Protection Requirements), this requirement is also deterministic. While the first performance measure is to consider all future credible scenarios, the other two apply only to undisturbed performance.

In addition, three other performance measures are used in 10 CFR 60.113 to define performance of individual barriers (in contrast to the total system). These performance measures are: (i) life of the waste package must exceed specified limits [10 CFR 60.113(a)(1)(ii)(A)—Substantially Complete Containment Requirement]; (ii) release from engineered barriers must be less than specified limits [10 CFR 60.113(a)(1)(ii)(B)—Groundwater Release Requirement]; and (iii) Groundwater Travel Time (GWTT) must be greater than specified limits [10 CFR 60.113(a)(2)—Groundwater Travel Time Requirement].

In all, there are six distinct performance measures. In general, a TPA computer code must allow for estimation of the three measures related to 40 CFR Part 191 and preferably, but not necessarily, for the other three related to 10 CFR 60.113. The steps for the assessment of the six performance measures include model conceptualization of process, assembly of data suitable for input to the mathematical models, consequence analysis, sensitivity uncertainty analysis, and regulatory compliance assessment. The proposed utilization of INVENT as a central inventory information source is beneficial, since it provides both uniformity and flexibility, and can easily be customized for the related models in each TPA code.

1.2 TOTAL-SYSTEM PERFORMANCE ASSESSMENT BACKGROUND

To estimate the performance measures, the TPA computer code contains a set of CMs that are computationally independent units, with their execution controlled by an Executive Module (Exec) (Sagar and Janetzke, 1993). The Exec acts as the manager and assures that CMs are executed in the desired sequence and that appropriate values of the common parameters are passed to CMs. The Exec of the TPA directs data flow between different subprocesses and controls their execution. After incorporation of INVENT into this system, all the required inventory information will be supplied from this common source. INVENT will be connected to the rest of the system through Exec intermodule communication interfaces.

1.3 PURPOSE OF SOFTWARE

The purpose of the INVENT module is to generate and store radionuclide inventory data for use by the TPA computer code. A strength of the INVENT module is that the user can easily change the key parameters which dictate the radionuclide inventory without being an expert in the ORIGEN2 software (the ORIGEN2 code performs the detailed calculations necessary to generate the radionuclide inventories). The software has been designed for the user to readily change the list of radionuclides of interest, the Pressurized Water Reactor/Boiling Water Reactor (PWR/BWR) mixture of fuel, the burnup of the fuel, the presence/absence of the cladding material, and the times at which the inventories are desired. A set of predefined choices have been implemented in the graphical user interface (GUI) software to facilitate these choices.

Off-the-shelf software used for the INVENT module included GALAXY for providing a graphical application development environment, ORIGEN2 for calculating the decay and buildup of the radionuclides, and the ORACLE database for storing and reporting results. GALAXY is an application developmental tool, and its Application Programming Interface (API) is utilized to provide the graphical development environment. ORIGEN2 is an existing FORTRAN code developed by Oak Ridge National Laboratory (ORNL) that can calculate radionuclide decay values for any desired time. ORACLE is a database application program. Although both FORTRAN and C compatibility is possible, the current implementation of ORACLE requires C code to access it (additional software would need to be purchased from Oracle Corporation to allow FORTRAN callable access to the ORACLE database). A more detailed description and capability information on these codes are provided in Chapter 4. In addition to the stand-alone INVENT module, a set of FORTRAN subroutines have been developed to efficiently store and retrieve the inventories of select radionuclides. It is envisioned that the FORTRAN subroutines can be used directly in the TPA code.

2 INSTRUCTIONS FOR USE OF THE INVENT MODULE

An executive routine of the INVENT module was developed using the GALAXY software to provide a user-friendly, window-type environment. The routine **invent** allows the user to select the characteristics of the spent fuel to be considered. The routine works by creating and issuing operating system commands that are used to run the ORIGEN2 and ORACLE software.

The routine and input files necessary for running ORIGEN2 were prepared in advance from which the user can select via the GUI. The software can be expanded to allow automatic edits of the input files using information from the GUI.

C programs for storing the data from files generated by ORIGEN2 into an ORACLE database and for generating reports were developed. Chapter 3 provides descriptions of the programs and Chapter 4 provides descriptions of external softwares used.

2.1 SOFTWARE CAPABILITIES ASSUMPTIONS

In order to use the INVENT module, the following must be satisfied:

- The GALAXY, ORIGEN2, and ORACLE software are installed
- The users **.cshrc** file is correct. A typical file is included in Appendix A
- The user's running environment is correctly configured, especially the DISPLAY environment variable

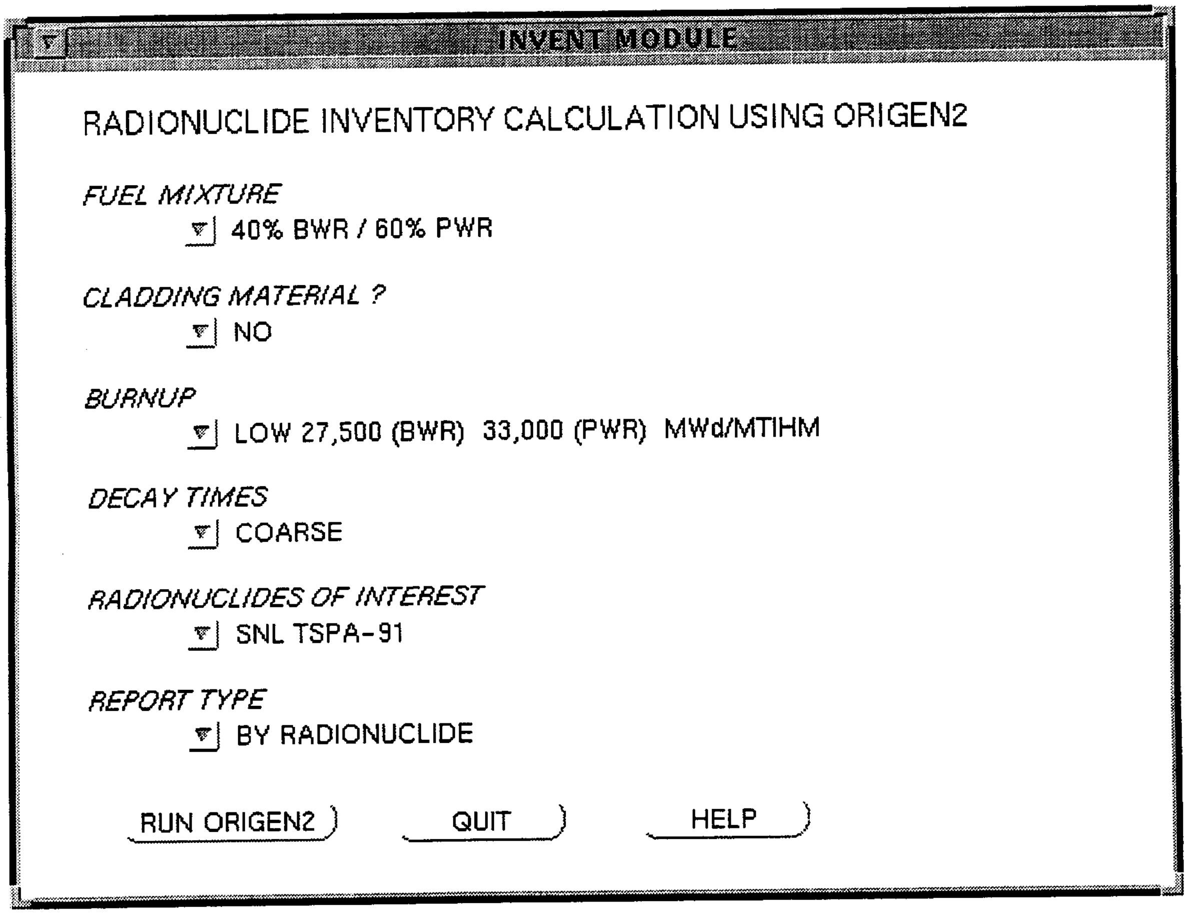
2.2 PROCEDURES FOR RUNNING THE INVENT MODULE

The **invent** program was developed on a SUN Workstation. The **invent** program can be created using the procedures outlined in Appendix B. It can be started by typing "invent &" at the prompt. The Graphical menu INVENT MODULE will appear (Figure 2-1). Clicking the RUN ORIGEN2 button with the mouse will run the codes and produce the selected inventory report. Other reports are generated based on the selections made. The possible selections are listed below.

FUEL MIXTURE
- 40% BWR / 60% PWR
- 100% BWR
- 100% PWR
- OTHER (not supported at this time)

CLADDING MATERIAL?
- NO
- YES

BURNUP
- LOW 27,500 (BWR) 33,000 (PWR) MWd/MTIHM
- MEDIUM 30,000 (BWR) 45,000 (PWR) MWd/MTIHM (not supported at this time)



RADIONUCLIDE INVENTORY CALCULATION USING ORIGEN2

FUEL MIXTURE
40% BWR / 60% PWR

CLADDING MATERIAL ?
NO

BURNUP
LOW 27,500 (BWR) 33,000 (PWR) MWd/MTIHM

DECAY TIMES
COARSE

RADIONUCLIDES OF INTEREST
SNL TSPA-91

REPORT TYPE
BY RADIONUCLIDE

Figure 2-1. Main window for INVENT module showing an example of selections for fuel mixture, cladding, burnup, decay times, radionuclides, and report type

HIGH 40,000 (BWR) 60,000 (PWR) MWd/MTIHM (not supported at this time)
OTHER (not supported at this time)

DECAY TIMES
COARSE (10, 30, 100, 300, 1,000, 3,000, 10,000, 30,000, 100,000, 300,000, and 1,000,000 years)

```
FINE     (1, 1.2, 1.5, 2, 2.5, 3, 3.5, 4, 5, 6, 7, 8 years) (times 1)
              "     "     "     "     "     "   (times 10)
              "     "     "     "     "     "   (times 100)
              "     "     "     "     "     "   (times 1,000)
              "     "     "     "     "     "   (times 10,000)
              "     "     "     "     "     "   (times 100,000)
         (1,000,000 years)
```

OTHER (not supported at this time)

RADIONUCLIDES OF INTEREST
* SNL TSPA-91 (list from SNL TSPA-91 report, Barnard et al., 1992)
* NRC IPA-1 (Iterative PA Phase 1 Radionuclides, Codell et al., 1992)
* NRC IPA-2 (Iterative PA Phase 2 Radionuclides, Wescott et al., 1994)
ALL (all radionuclides that ORIGEN2 produced)
OTHER (not supported at this time)

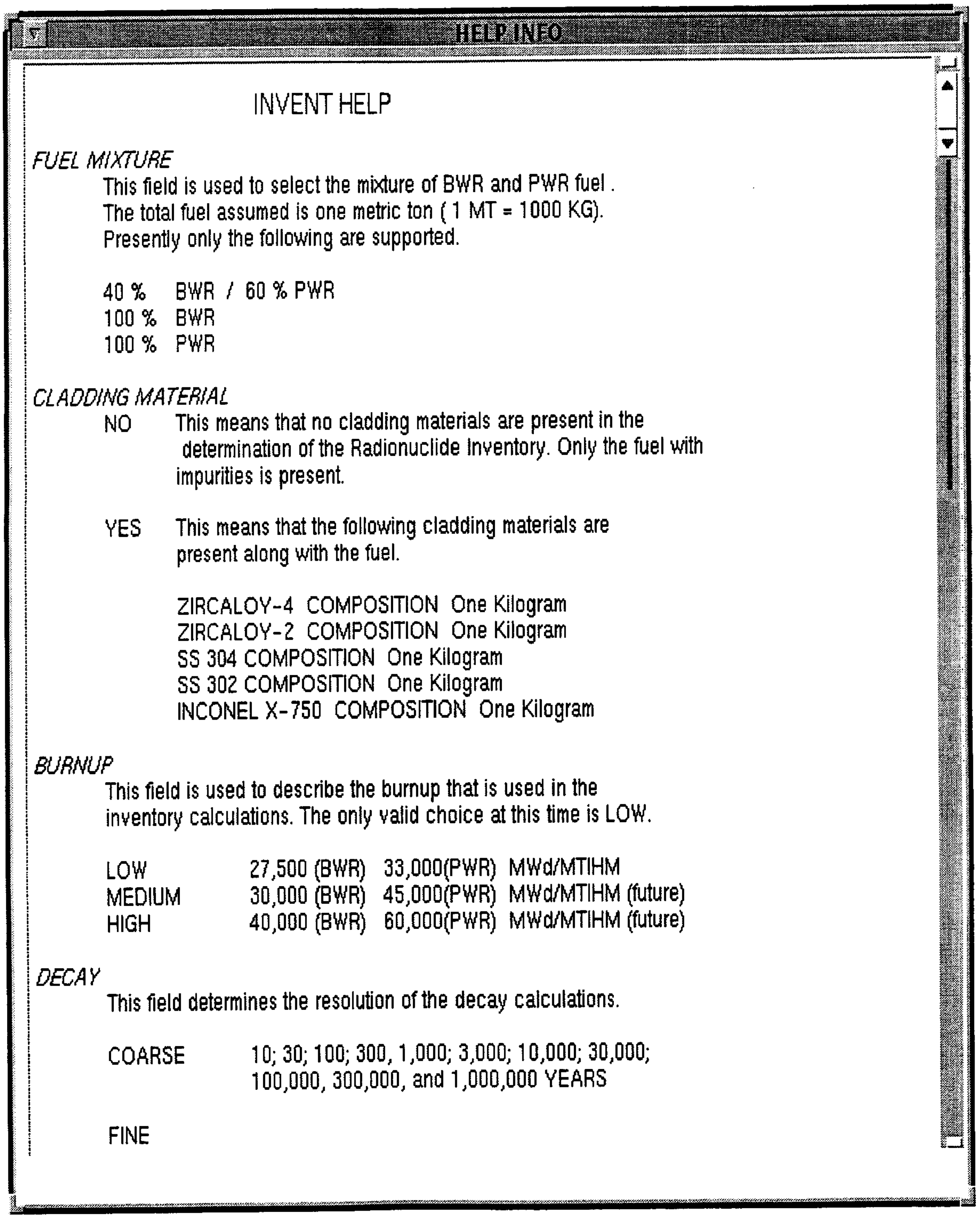
REPORT TYPE
SORT BY RADIONUCLIDE
SORT BY TIME

RUN ORIGEN2 BUTTON
When this button is selected, the appropriate C shell program (Figure 2-3) will run the ORIGEN2 code with the proper input files to produce the desired result file. Then the database build program will store the results in the ORACLE database (Figure 2-4). Finally the report program will run and produce the desired report (Figure 2-5).

HELP BUTTON
When this button is selected, the help dialog will appear (Figure 2-2).

QUIT BUTTON
This button terminates the **invent** program.

* Note: Refer to Help Dialog (Figure 2-2) for complete list of radionuclides.



INVENT HELP

FUEL MIXTURE

This field is used to select the mixture of BWR and PWR fuel .
The total fuel assumed is one metric ton (1 MT = 1000 KG).
Presently only the following are supported.

40 %	BWR / 60 % PWR
100 %	BWR
100 %	PWR

CLADDING MATERIAL

NO This means that no cladding materials are present in the determination of the Radionuclide Inventory. Only the fuel with impurities is present.

YES This means that the following cladding materials are present along with the fuel.

ZIRCALOY-4 COMPOSITION One Kilogram
ZIRCALOY-2 COMPOSITION One Kilogram
SS 304 COMPOSITION One Kilogram
SS 302 COMPOSITION One Kilogram
INCONEL X-750 COMPOSITION One Kilogram

BURNUP

This field is used to describe the burnup that is used in the inventory calculations. The only valid choice at this time is LOW.

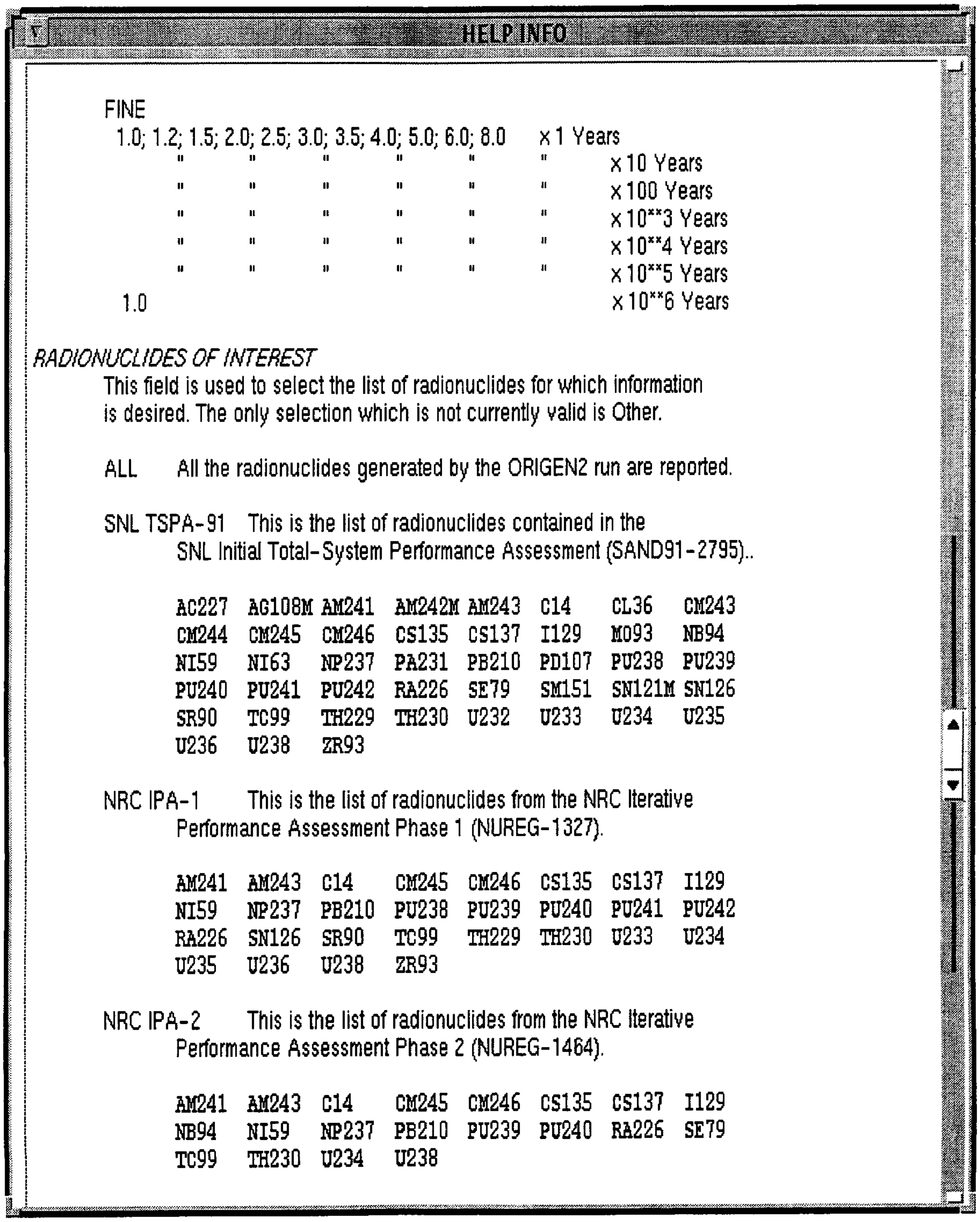
LOW	27,500 (BWR)	33,000(PWR)	MWd/MTIHM
MEDIUM	30,000 (BWR)	45,000(PWR)	MWd/MTIHM (future)
HIGH	40,000 (BWR)	60,000(PWR)	MWd/MTIHM (future)

DECAY

This field determines the resolution of the decay calculations.

COARSE	10; 30; 100; 300, 1,000; 3,000; 10,000; 30,000; 100,000, 300,000, and 1,000,000 YEARS
FINE	

Figure 2-2. Help dialog window for the INVENT module



```
FINE
 1.0; 1.2; 1.5; 2.0; 2.5; 3.0; 3.5; 4.0; 5.0; 6.0; 8.0    x 1 Years
       "      "      "      "      "      "          x 10 Years
       "      "      "      "      "      "          x 100 Years
       "      "      "      "      "      "          x 10**3 Years
       "      "      "      "      "      "          x 10**4 Years
       "      "      "      "      "      "          x 10**5 Years
 1.0                                                 x 10**6 Years
```

RADIONUCLIDES OF INTEREST

This field is used to select the list of radionuclides for which information is desired. The only selection which is not currently valid is Other.

ALL All the radionuclides generated by the ORIGEN2 run are reported.

SNL TSPA-91 This is the list of radionuclides contained in the SNL Initial Total-System Performance Assessment (SAND91-2795)..

```
AC227  AG108M AM241  AM242M AM243  C14    CL36   CM243
CM244  CM245  CM246  CS135  CS137  I129   MO93   NB94
NI59   NI63   NP237  PA231  PB210  PD107  PU238  PU239
PU240  PU241  PU242  RA226  SE79   SM151  SN121M SN126
SR90   TC99   TH229  TH230  U232   U233   U234   U235
U236   U238   ZR93
```

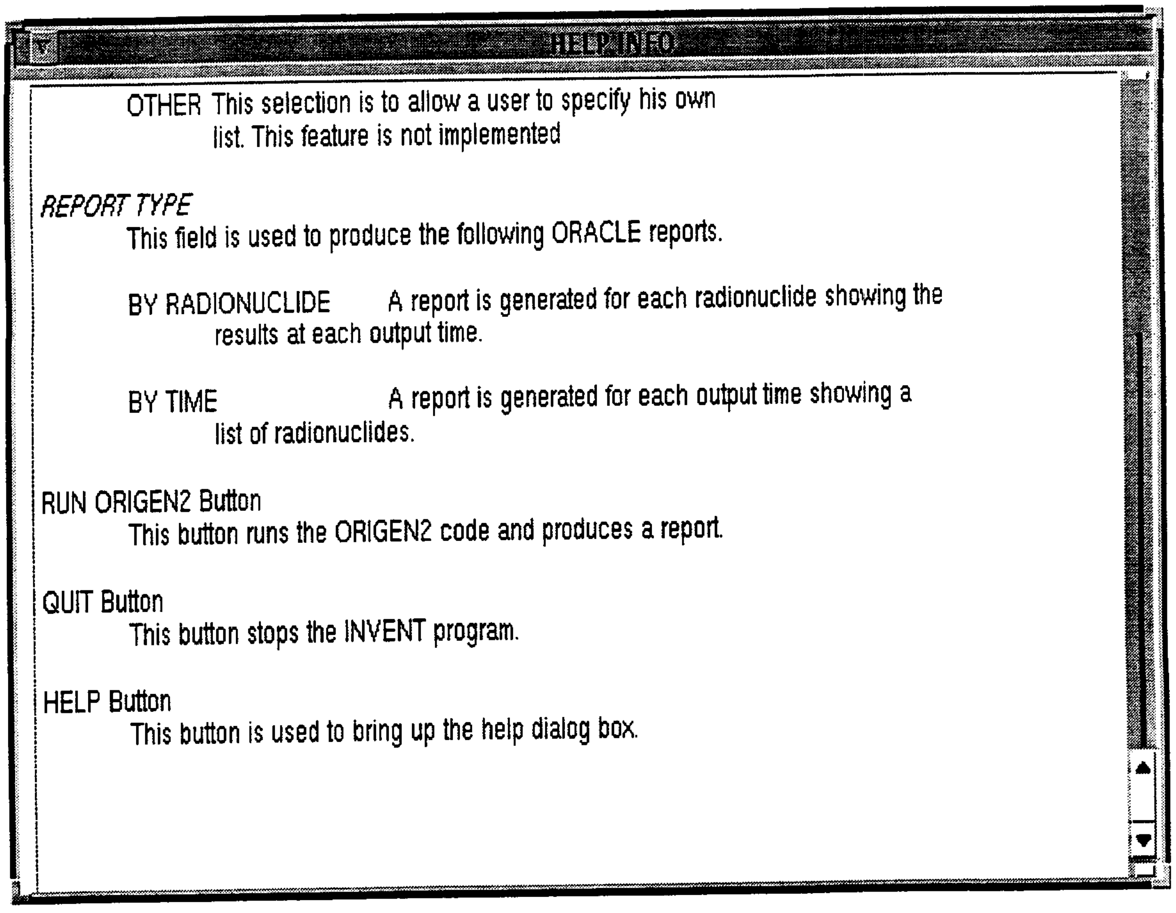
NRC IPA-1 This is the list of radionuclides from the NRC Iterative Performance Assessment Phase 1 (NUREG-1327).

```
AM241  AM243  C14    CM245  CM246  CS135  CS137  I129
NI59   NP237  PB210  PU238  PU239  PU240  PU241  PU242
RA226  SN126  SR90   TC99   TH229  TH230  U233   U234
U235   U236   U238   ZR93
```

NRC IPA-2 This is the list of radionuclides from the NRC Iterative Performance Assessment Phase 2 (NUREG-1464).

```
AM241  AM243  C14    CM245  CM246  CS135  CS137  I129
NB94   NI59   NP237  PB210  PU239  PU240  RA226  SE79
TC99   TH230  U234   U238
```

Figure 2-2 (Cont'd). Help dialog window for the INVENT module



OTHER This selection is to allow a user to specify his own list. This feature is not implemented

REPORT TYPE

This field is used to produce the following ORACLE reports.

BY RADIONUCLIDE A report is generated for each radionuclide showing the results at each output time.

BY TIME A report is generated for each output time showing a list of radionuclides.

RUN ORIGEN2 Button

This button runs the ORIGEN2 code and produces a report.

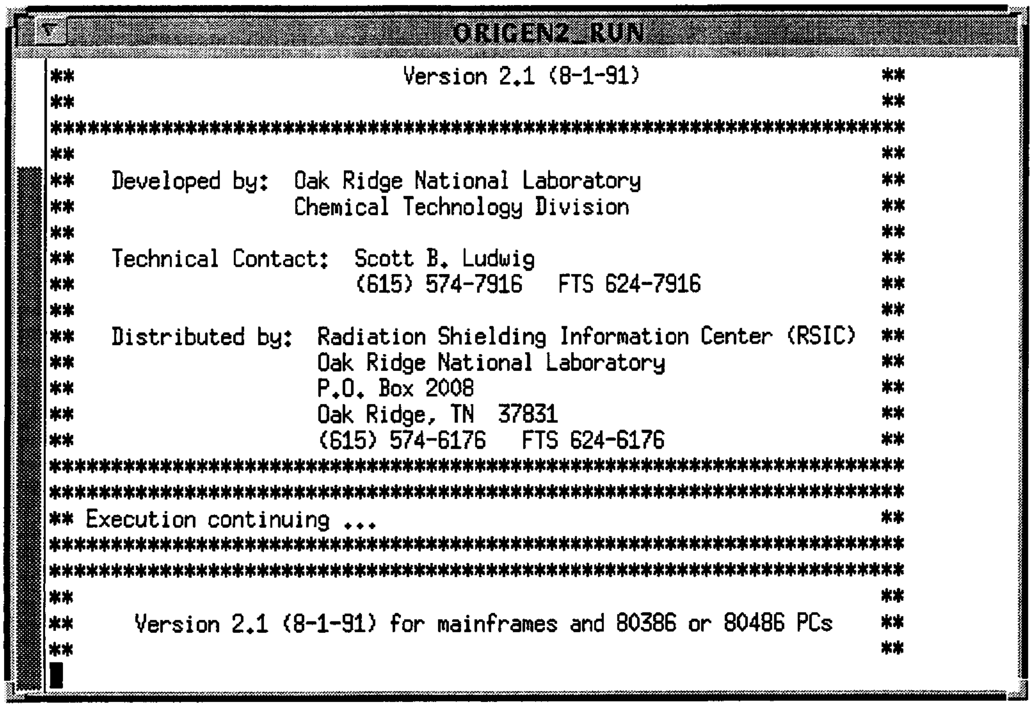
QUIT Button

This button stops the INVENT program.

HELP Button

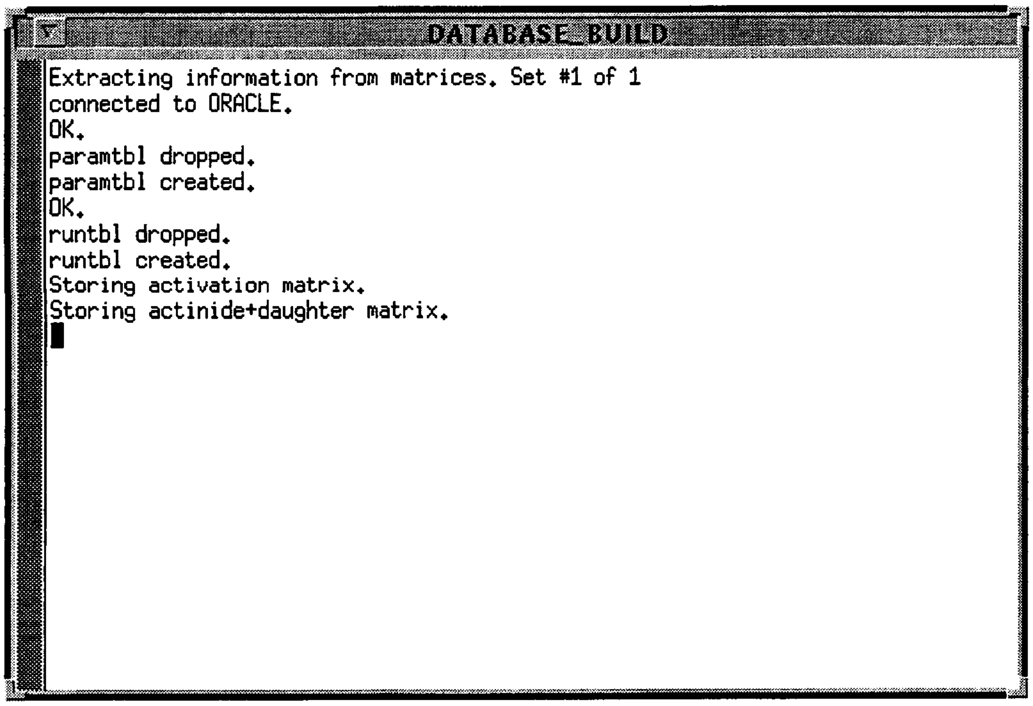
This button is used to bring up the help dialog box.

Figure 2-2 (Cont'd). Help dialog window for the INVENT module



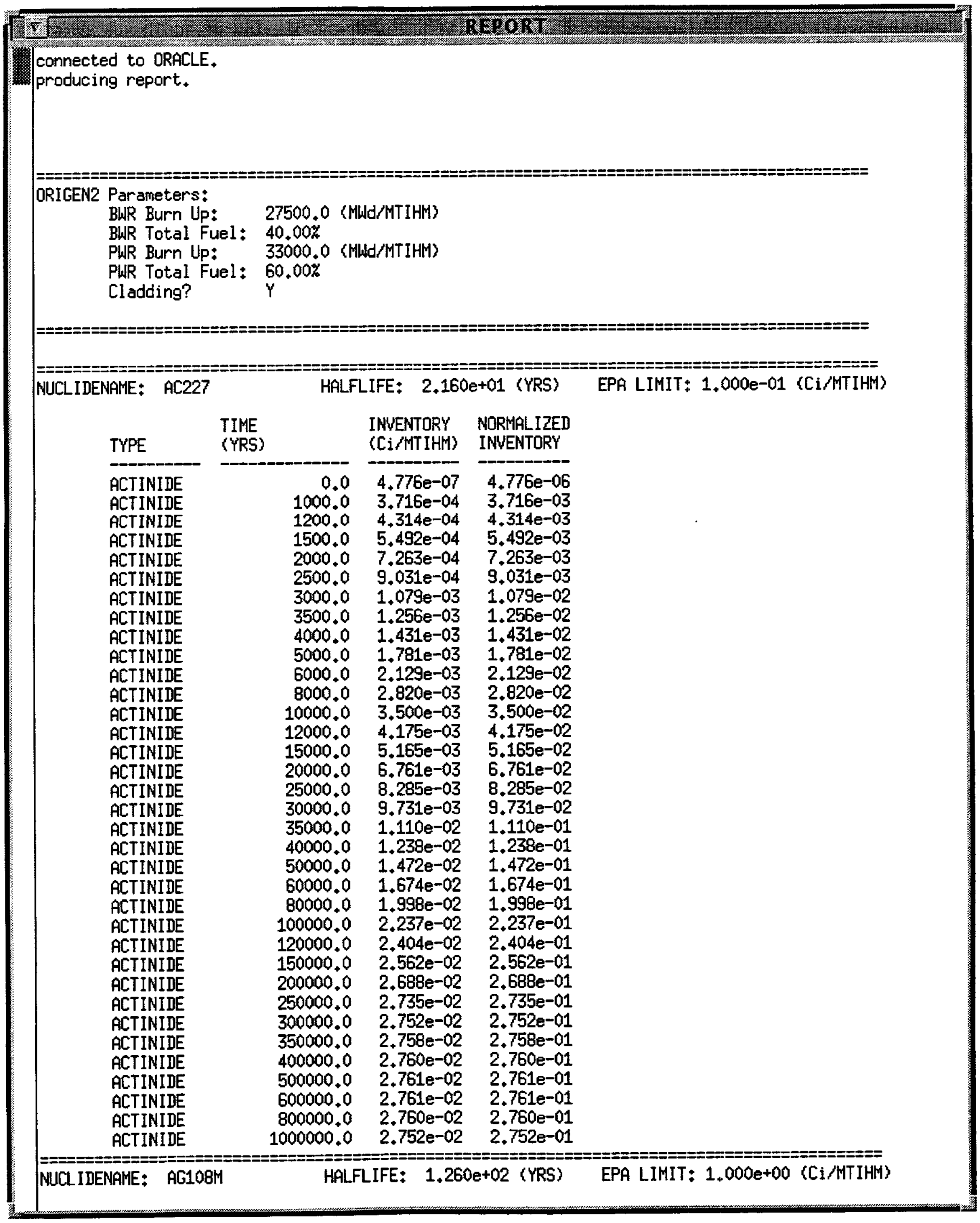
```
**                         Version 2.1 (8-1-91)                         **
**                                                                      **
**************************************************************************
**                                                                      **
**   Developed by:  Oak Ridge National Laboratory                       **
**                  Chemical Technology Division                        **
**                                                                      **
**   Technical Contact:  Scott B. Ludwig                                **
**                       (615) 574-7916   FTS 624-7916                  **
**                                                                      **
**   Distributed by:  Radiation Shielding Information Center (RSIC)    **
**                    Oak Ridge National Laboratory                     **
**                    P.O. Box 2008                                     **
**                    Oak Ridge, TN  37831                              **
**                    (615) 574-6176   FTS 624-6176                     **
**************************************************************************
**************************************************************************
** Execution continuing ...                                             **
**************************************************************************
**************************************************************************
**                                                                      **
**     Version 2.1 (8-1-91) for mainframes and 80386 or 80486 PCs      **
**                                                                      **
```

Figure 2-3. Example of a window from an ORIGEN2 run generated by the INVENT module



```
Extracting information from matrices. Set #1 of 1
connected to ORACLE.
OK.
paramtbl dropped.
paramtbl created.
OK.
runtbl dropped.
runtbl created.
Storing activation matrix.
Storing actinide+daughter matrix.
```

Figure 2-4. Example of a window from an ORACLE database operation generated by the INVENT module



```
connected to ORACLE.
producing report.



=====================================================================================
ORIGEN2 Parameters:
       BWR Burn Up:     27500.0 (MWd/MTIHM)
       BWR Total Fuel:  40.00%
       PWR Burn Up:     33000.0 (MWd/MTIHM)
       PWR Total Fuel:  60.00%
       Cladding?        Y

=====================================================================================

======================================================================================
NUCLIDENAME:  AC227            HALFLIFE:  2.160e+01 (YRS)    EPA LIMIT: 1.000e-01 (Ci/MTIHM)

                    TIME            INVENTORY   NORMALIZED
       TYPE         (YRS)           (Ci/MTIHM)  INVENTORY
       ----------   --------------  ----------  ----------
       ACTINIDE                0.0  4.776e-07   4.776e-06
       ACTINIDE             1000.0  3.716e-04   3.716e-03
       ACTINIDE             1200.0  4.314e-04   4.314e-03
       ACTINIDE             1500.0  5.492e-04   5.492e-03
       ACTINIDE             2000.0  7.263e-04   7.263e-03
       ACTINIDE             2500.0  9.031e-04   9.031e-03
       ACTINIDE             3000.0  1.079e-03   1.079e-02
       ACTINIDE             3500.0  1.256e-03   1.256e-02
       ACTINIDE             4000.0  1.431e-03   1.431e-02
       ACTINIDE             5000.0  1.781e-03   1.781e-02
       ACTINIDE             6000.0  2.129e-03   2.129e-02
       ACTINIDE             8000.0  2.820e-03   2.820e-02
       ACTINIDE            10000.0  3.500e-03   3.500e-02
       ACTINIDE            12000.0  4.175e-03   4.175e-02
       ACTINIDE            15000.0  5.165e-03   5.165e-02
       ACTINIDE            20000.0  6.761e-03   6.761e-02
       ACTINIDE            25000.0  8.285e-03   8.285e-02
       ACTINIDE            30000.0  9.731e-03   9.731e-02
       ACTINIDE            35000.0  1.110e-02   1.110e-01
       ACTINIDE            40000.0  1.238e-02   1.238e-01
       ACTINIDE            50000.0  1.472e-02   1.472e-01
       ACTINIDE            60000.0  1.674e-02   1.674e-01
       ACTINIDE            80000.0  1.998e-02   1.998e-01
       ACTINIDE           100000.0  2.237e-02   2.237e-01
       ACTINIDE           120000.0  2.404e-02   2.404e-01
       ACTINIDE           150000.0  2.562e-02   2.562e-01
       ACTINIDE           200000.0  2.688e-02   2.688e-01
       ACTINIDE           250000.0  2.735e-02   2.735e-01
       ACTINIDE           300000.0  2.752e-02   2.752e-01
       ACTINIDE           350000.0  2.758e-02   2.758e-01
       ACTINIDE           400000.0  2.760e-02   2.760e-01
       ACTINIDE           500000.0  2.761e-02   2.761e-01
       ACTINIDE           600000.0  2.761e-02   2.761e-01
       ACTINIDE           800000.0  2.760e-02   2.760e-01
       ACTINIDE          1000000.0  2.752e-02   2.752e-01
======================================================================================
NUCLIDENAME:  AG108M           HALFLIFE:  1.260e+02 (YRS)    EPA LIMIT: 1.000e+00 (Ci/MTIHM)
```

Figure 2-5. Example of a window containing the final INVENT module radionuclide inventory report

3 DESCRIPTION OF THE INVENT MODULE SOFTWARE

3.1 OVERVIEW OF SOFTWARE

The objective of this task is to develop an application that allows a user to interact with the inventory software using a GUI. The purpose of the application is to obtain an inventory report for a selected list of radionuclides for various combinations of fuel mixture, burnup, and cladding materials. In addition, the user must specify the times at which the radionuclide inventories are desired; this is referred to as the decay time resolution.

3.2 TECHNICAL APPROACH

3.2.1 External Software Required

The INVENT module utilized two commercial software packages and an off-the-shelf computer code to produce the radionuclide inventory for use in TPA. The two commercial packages are: (i) GALAXY, and (ii) ORACLE. The actual inventory computations are performed by ORIGEN2, a code obtained from Oak Ridge National Laboratory. The function and use of these programs are described below. The version that was used is indicated in parenthesis.

3.2.1.1 Program Control Using GALAXY (Version 1.2)

The control of INVENT is accomplished using a main program configured by the user. The configuration is accomplished by allowing the user to specify his requests via a GUI composed of windows created using the GALAXY software.

3.2.1.2 Inventory Calculations Using ORIGEN2 (Version 2.1)

The inventory is a function of fuel mixture, burnup, and cladding material. The ORNL Isotope GENeration and depletion code (ORIGEN2) is used for the determination of the inventory. The software allows the user to select from various C shell scripts written to execute the ORIGEN2 code with different input files to produce the desired results. This selection occurs automatically, dependent on the information entered via the GUI.

3.2.1.3 Inventory Information Storage and Retrieval Using ORACLE (Version 6.0)

After ORIGEN2 is run, the inventory information is written in an ASCII file containing more information than required, and in a format which makes retrievability difficult. The design of INVENT incorporates a commercial database system (ORACLE) to simplify the retrieval of inventory information.

The program **origdb** was written to extract information from the ORIGEN2 output file and place it in the ORACLE database. This program is automatically executed by the INVENT main program **invent** described in Sections 3.4 and 3.5.

3.2.1.4 Inventory Reporting Using ORACLE

Reporting of the inventory is accomplished using the program **rptdb**, which was written to produce reports using the information retrieved from the ORACLE database. Section 3.6 provides a detailed description of **rptdb** program. The program can produce a report sorted by either time or radionuclide.

3.2.1.5 Other

The current version of the INVENT module is designed to run on computers using a UNIX operating system with an X windows software. The design also assumes that an ANSI C compiler and other standard libraries are available.

3.2.2 Hardware Required

The INVENT module assumes that the workstation has sufficient capability to support UNIX Operating System with an X windows software, ORIGEN2, ORACLE, GALAXY, and other required software.

3.3 ENVIRONMENT

This section requires a basic understanding of UNIX and X windows because it describes the environmental details necessary for the INVENT module to run. The version of INVENT described in this report was developed for a SUN Workstation running the SunOS 4.1.3 operating system. The proper environment must exist before the INVENT module is run. The software assumes that the directory structure is fixed relative to a home position. The environment variables REPORTS_HOME and ORIGEN2_HOME must be set to the appropriate home directories. This setting of home directories is normally done in the user's **.cshrc** file (refer to Appendix A, Setup Files). The software uses the variables in constructing the paths to files needed at run time. This setup should be created with the assistance of someone experienced in both X Windows and UNIX.

Example:

```
setenv REPORTS_HOME /user3/goliath/jcornel/db/origen2
setenv ORIGEN2_HOME /user3/goliath/alozano/origen2
```

The DISPLAY environment variable must be set to the computer that is running the X Window server.

Example:

```
setenv DISPLAY geordi.space.swri.edu:0.0
```

The user's account must be setup so that the user has access to the ORACLE software.

3.4 DESCRIPTION OF DATA FLOW FOR INVENT MODULE

The INVENT module is made up of several processes which pass and receive information fom each other. This section is a verbal description of the information flow represented graphically in Figure 3-1. This figure represents only the information flow which is considered part of the INVENT module.

3.4.1 Preliminary INVENT Operation

Before the main operation can be performed, a table containing the half-life values for each radionuclide in the Decay library must be constructed for use in the main INVENT operation. This flow is depicted in Figure 3.1. The **human** runs the process **hlgen** and causes the half-lifes to be extracted and then stored in the database via the ORACLE processor. This information is later used in the report displays.

3.4.2 Main INVENT Operation

The INVENT module user is the initiator of the code execution sequence that eventually results in a display consisting of the radionuclide report that is requested. The main program is **invent**. The following are the steps that occur in the main INVENT operation.

(i) The **human** configures the INVENT module by making selections from the Main window for INVENT, Figure 2.1.

(ii) When the **human** presses the RUN ORIGEN2 button, the parameters which are not contained in the ORIGEN2 output files are written to a file for use by **origdb** (database builder).

iii) After the files are written, an xterm window is started with the appropriate C-shell program. The xterm provides the terminal emulation for an application that requires a terminal. The C-shell program that is executed is determined from the selections made in step (i).

(iv) The C-shell program defines all the appropriate input files and then runs the ORIGEN2 software that produces an ORIGEN2 output file. This file contains the selected radionuclide information in ASCII form.

(v) After the ORIGEN2 run is completed, the **invent** program runs the program **origdb**. This program places the information from the parameters file, extracts the radionuclide inventory information from the ORIGEN2 output file, and stores all the information, including half life, in an ORACLE database using the ORACLE processor. Although not shown in the figure, **origdb** runs in an xterm window so that print commands within the program can be viewed.

(vi) When all the data has been stored in the database, **invent** runs the program **rptdb** with arguments indicating the desired report type and radionuclide list. The program **rptdb** constructs the appropriate SQL query to extract the desired data. The SQL commands are

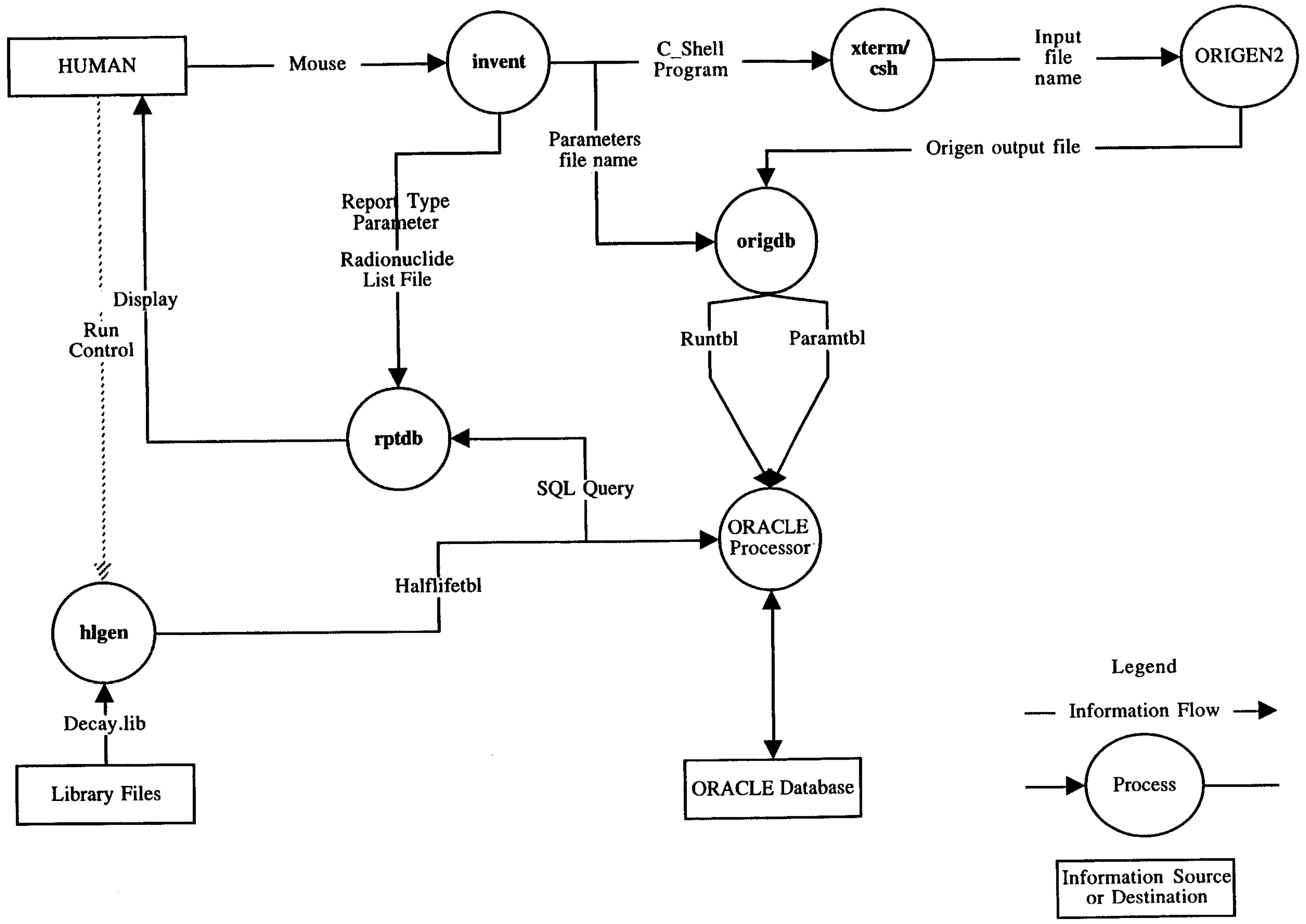


Figure 3-1. Overall data flow diagram for INVENT module

executed by the ORACLE processor and the appropriate information is returned. Although not shown in the figure, **rptdb** runs in an xterm window so that print commands within the program can be viewed.

(vii) As **rptdb** executes, the radionuclide report is displayed on the xterm window. The xterm supplies a scroll bar so that all data that is displayed can be viewed at completion. This xterm display closes the loop by visually reporting the requested information to the **human**.

3.5 DESCRIPTION OF invent

Most processing in programs using a GUI occurs in the event handler, not in the main program. The branching to functions is handled through a registration process of assigning functions to buttons, menus, etc. Branching within this type of program is impossible to describe accurately because events interrupt the flow; however, a functional flowchart can be developed by showing the event handlers as branches of decision diamonds. These assigned functions are called notify functions.

The main program for running the INVENT module is **invent**. The following sections provide a functional description of the software. Much of the functionality is not visible in the source code because it is handled within the GALAXY and X software. Most of the boxes in the main flowchart are implemented as calls to GALAXY functions. Some decision diamonds in the event handler flowchart represent the mechanism used within GALAXY to determine which notify function to call based on a user event. The boxes in the branches from these diamonds are implemented as notify functions and are visible in the source code.

3.5.1 Description of Main Program

Figure 3-2 is a flowchart of the **invent** main program and a listing of the source code is included in Appendix C. The numbered boxes in Figure 3-2 are described below. The purpose of this section is to expand on the information provided in the figure.

1. START

   Begin the program.

2. SET DEFAULTS

   The main program sets the appropriate data structures to defaults that match the startup settings of the GUI.

3. GET ENVIRONMENT VARIABLES FOR REPORTS_HOME AND ORIGEN2_HOME

   The environment parameters are obtained so that the program can properly locate data and program files.

4. INITIALIZE GALAXY

   Initialize the GALAXY software.

5. LOAD THE MAIN DIALOG

   Action specific for GALAXY to initialize the Main (see Figure 2-1).

6. LOAD THE HELP DIALOG

   Actions specific for GALAXY to initialize the Help window box (see Figure 2-2).

7. SETUP NOTIFY FUNCTION FOR MAIN DIALOG CLOSE

   Actions specific for GALAXY to assign functions to main window quit menu selection when a user activates main window buttons.

8. SETUP NOTIFY FUNCTIONS FOR THE RUN, HELP, AND QUIT BUTTONS

   Actions specific for GALAXY to assign functions to Run, Help, or Quit buttons.

9. SETUP NOTIFY FUNCTIONS FOR THE MENU ITEMS FOR THE MIXTURE, CLADDING, BURNUP, DECAY TIMES, RADIONUCLIDES, AND REPORTS MENUS

   Actions specific for GALAXY to assign functions to menu items.

10. OPEN THE MAIN DIALOG WINDOW

    Item 10 represents the GALAXY call to make the main window visible to the user.

11. CALL EVENT HANDLER FUNCTION

    The program remains within this function until a call to a GALAXY function is made that stops event processing. This call occurs when the user presses the window quit button or selects quit from window manager menu.

12. DESTROY HELP DIALOG

    Make help window invisible and cleanly release system resources.

13. DESTROY MAIN DIALOG

    Make the main window invisible and cleanly release system resources.

14. END

    Terminate the Main program.

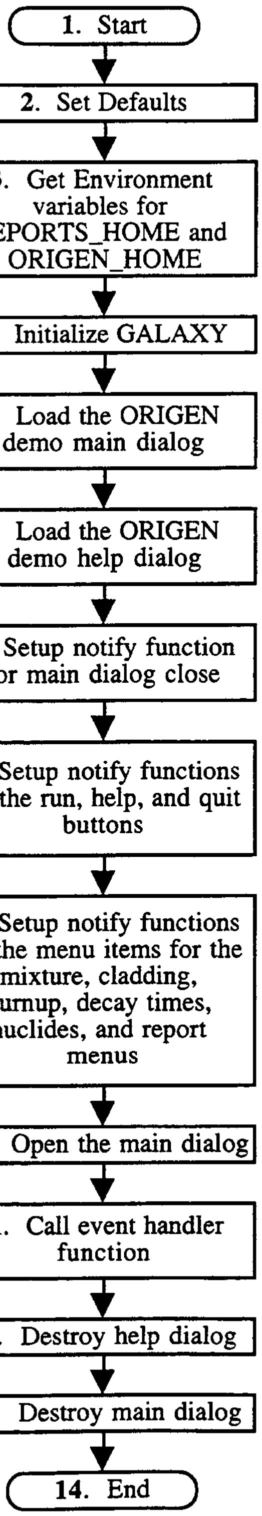

1. Start
2. Set Defaults
3. Get Environment variables for REPORTS_HOME and ORIGEN_HOME
4. Initialize GALAXY
5. Load the ORIGEN demo main dialog
6. Load the ORIGEN demo help dialog
7. Setup notify function for main dialog close
8. Setup notify functions for the run, help, and quit buttons
9. Setup notify functions for the menu items for the mixture, cladding, burnup, decay times, nuclides, and report menus
10. Open the main dialog
11. Call event handler function
12. Destroy help dialog
13. Destroy main dialog
14. End

Figure 3-2. Main program flowchart for invent

3.5.2 Description of Event Handler

Figure 3-3 is a flowchart representing the event handler and Notify functions which is from Box 11 in Figure 3-2. The WAIT FOR EVENT box represents the GALAXY event handler function. The decision diamonds for MIXTURE, CLADDING MATERIAL, BURNUP, DECAY TIMES, RADIONUCLIDE LIST, REPORT, HELP, RUN, and QUIT are not a part of the code, but are actually calls to the previously registered functions by the event handler. Each branch from these diamonds is a separate function in the code. The numbered boxes in Figure 3-3 are elaborated further.

1. START EVENT HANDLER

   This item represents the event handler initialization.

2. WAIT FOR EVENT

   This item represents the event idle state.

3. MIXTURE?

   Was the event a mixture event?

if YES

4. SET MIXTURE TO 40%BWR/60%PWR, 100%BWR, 100%PWR, or OTHER

   Go to 2.
   This item represents the setting from the mixture menu.

if NO Continue

5. CLADDING?

   Was the event a cladding event?

if YES

6. SET CLADDING TO YES OR NO

   Go to 2.
   Yes means include cladding materials; no means do not include cladding materials.

if NO Continue

7. BURNUP?

   Was the event a burnup event?

if YES

8. SET BURNUP TO LOW, MEDIUM, HIGH, OR OTHER

   Go to 2.

if NO Continue

9. DECAY TIMES?

   Was the event a decay times event?

if YES

10. SET DECAY TIMES TO COARSE, FINE, OR OTHER

    Go to 2.

if NO Continue

11. NUCLIDE LIST?

    Was the event a radionuclide list event?

if YES

12. SET RADIONUCLIDES TO ALL ORIGEN2, SNL TSPA-91, NRC IPA-1, NRC IPA-2, OR OTHER

    Go to 2.

if NO Continue

13. REPORT?

    Was the event a report event?

if YES

14. SET REPORT TO SORTED BY RADIONUCLIDE OR SORTED BY TIME

    Go to 2.

if NO Continue

15. HELP?

Was the event the help event?

if YES

16. OPEN THE HELP DIALOG WINDOW

Go to 2.

Make the help dialog visible. The help dialog can remain visible as an aid to setting the main dialog.

if NO Continue

17. RUN?

Was the event the run event.

if NO Go to 48

if YES Continue

18. OPEN PARAMETERS FILE

Information written into this file is used by the report program **rptdb**.

19. IS ANY PARAMETER SET TO OTHER?

if YES

20. RUN **xterm** AND OTHER.COM C SHELL PROGRAM

This is a filler path that could be expanded to fit other cases in the future. The other.com C shell program is run within an **xterm** window.

21. CLOSE PARAMETERS FILE

Close the parameters file.
Go to 2.

if NO Continue

22. IS BURNUP SET TO MEDIUM OR HIGH?

if YES

23. RUN **xterm** AND BURNUP.COM C SHELL PROGRAM

This is a filler path that could be expanded to fit the medium and high cases in the future. The burnup.com C shell program is run within an **xterm** window, a virtual terminal within an X window.

24. CLOSE PARAMETERS FILE

Close parameters file.
Go to 2.

if NO Continue

25. IS THIS THE FIRST RUN OR HAS A SETTING FOR MIXTURE, CLADDING, BURNUP, OR DECAY TIMES CHANGED FROM PREVIOUS RUN?

if NO Go to 44

Skip the ORIGEN2 run and the ORACLE database build, and proceed directly to the report generation.

if YES Continue

26. MIXTURE 40%BWR 60%PWR?

if YES

27A. CONSTRUCT FILENAME BASED ON BURNUP
LOW bwr_low_pwr_low
MEDIUM bwr_medium_pwr_medium
HIGH bwr_high_pwr_high

28A. WRITE PARAMETERS TO PARAMETERS FILE
Go to 31.

if NO Continue

29. MIXTURE 100%BWR?

if YES

27B.CONSTRUCT FILENAME BASED ON BURNUP
LOW bwr_low_
MEDIUM bwr_medium_
HIGH bwr_high_

28B.WRITE PARAMETERS TO PARAMETERS FILE
Go to 31.

```
if NO       Continue

30.   MIXTURE 100%PWR?

if YES

        27C.CONSTRUCT FILENAME BASED ON BURNUP
            LOW pwr_low_
            MEDIUM pwr_medium_
            HIGH pwr_high_

        28C.WRITE PARAMETERS TO PARAMETERS FILE
            Go to 31.

if NO       Continue

31.   CLADDING case?

if NO

        32. APPEND "nclad_" TO FILENAME

        33. WRITE PARAMETERS TO PARAMETERS FILE
            Go to 36.

if YES      Continue

34.   APPEND "wclad_" TO FILENAME

35.   WRITE PARAMETERS TO PARAMETERS FILE

36.   CLOSE PARAMETERS FILE

37.   DECAY TIMES COARSE?

if YES

        38. APPEND "coarse.com" TO FILENAME
            Go to 41. See description in item 40.

if NO       Continue

39.   DECAY TIMES FINE?

if YES
```

40. APPEND "fine.com" TO FILENAME
Go to 41.

Note: Up to this point, an input filename is created. The filename is that of a previously prepared C shell program that configures the libraries and input files to run ORIGEN2 for the requested conditions.

Example: For a mixture of 40%BWR 60%PWR with cladding materials at low burnup and fine decay time resolution, the filename is "bwr_low_pwr_low_wclad_fine.com". The com file is described in more detail in Appendix D.

if NO Continue

41. RUN **xterm** AND CONSTRUCTED FILENAME C SHELL PROGRAM (ORIGEN2)

The constructed filename C shell program is run within an **xterm** window (see Appendix D for a listing of ORIGEN2 command files and Appendix E for a listing of ORIGEN2 input files). This C shell program runs ORIGEN2 with the requested condition and produces a flat output file containing the results.

42. SETUP INPUTS FOR **origdb** PROGRAM

The inputs are command line arguments. The form of the command string is "origdb <ORIGEN2 OUTPUT FILE> 6 <PARAMETERS FILE>". The integer 6 is the setting for the FINE decay times resolution and 1 for the COARSE resolution.

43. RUN **xterm** AND **origdb** PROGRAM (ORACLE database build)

Reports Section

44. IS PARAMETERS FILE OPEN?

if NO Go to 46

if YES Continue

45. CLOSE PARAMETERS FILE

46. SETUP INPUTS FOR **rptdb** PROGRAM

The inputs are command line arguments. The form of the command string is "rptdb <Radionuclide List Filename> 0". The integer 0 means sort the results by radionuclide name and 1 means sort the results by time. The Radionuclide List Filename is a file that contains the list of radionuclides requested for the report.

47. RUN **xterm** AND **rptdb** PROGRAM (ORACLE database report)

48. QUIT?

Was the event the quit event?

if YES Go to 50

if NO Continue

49. MAIN WINDOW CLOSE?

if YES Go to 50

if NO Go to 2

50. STOP EVENT HANDLER

51. RETURN

This box represents the termination of the event loop, which will allow the main program to execute to completion.

Each branch from these diamonds is implemented as a set of notify functions. The function from the set that is called is based on the selection made by the user.

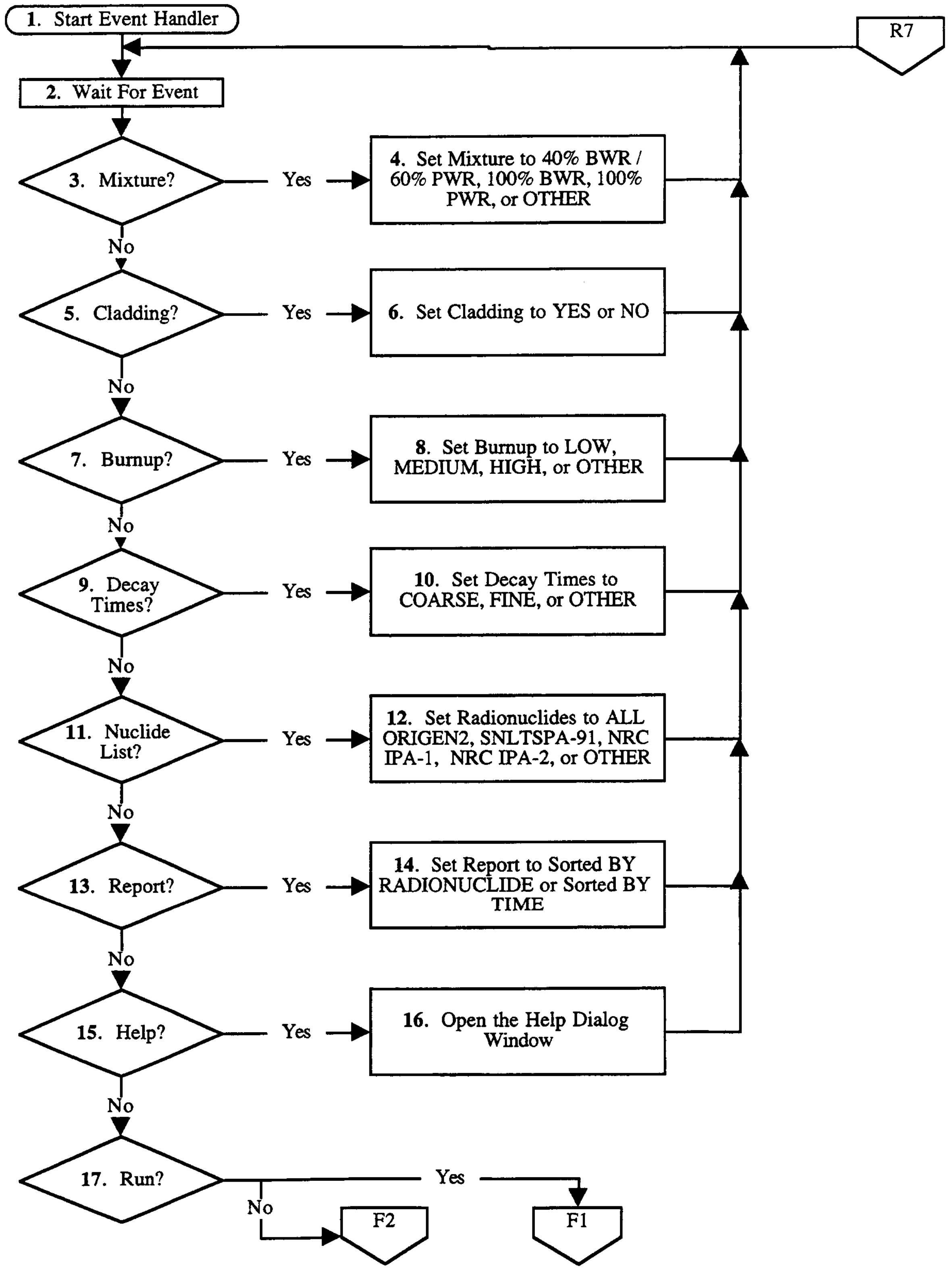

1. Start Event Handler
R7
2. Wait For Event
3. Mixture?
Yes
4. Set Mixture to 40% BWR / 60% PWR, 100% BWR, 100% PWR, or OTHER
No
5. Cladding?
Yes
6. Set Cladding to YES or NO
No
7. Burnup?
Yes
8. Set Burnup to LOW, MEDIUM, HIGH, or OTHER
No
9. Decay Times?
Yes
10. Set Decay Times to COARSE, FINE, or OTHER
No
11. Nuclide List?
Yes
12. Set Radionuclides to ALL ORIGEN2, SNLTSPA-91, NRC IPA-1, NRC IPA-2, or OTHER
No
13. Report?
Yes
14. Set Report to Sorted BY RADIONUCLIDE or Sorted BY TIME
No
15. Help?
Yes
16. Open the Help Dialog Window
No
17. Run?
Yes
No
F2
F1

Figure 3-3. Event handler flowchart for invent

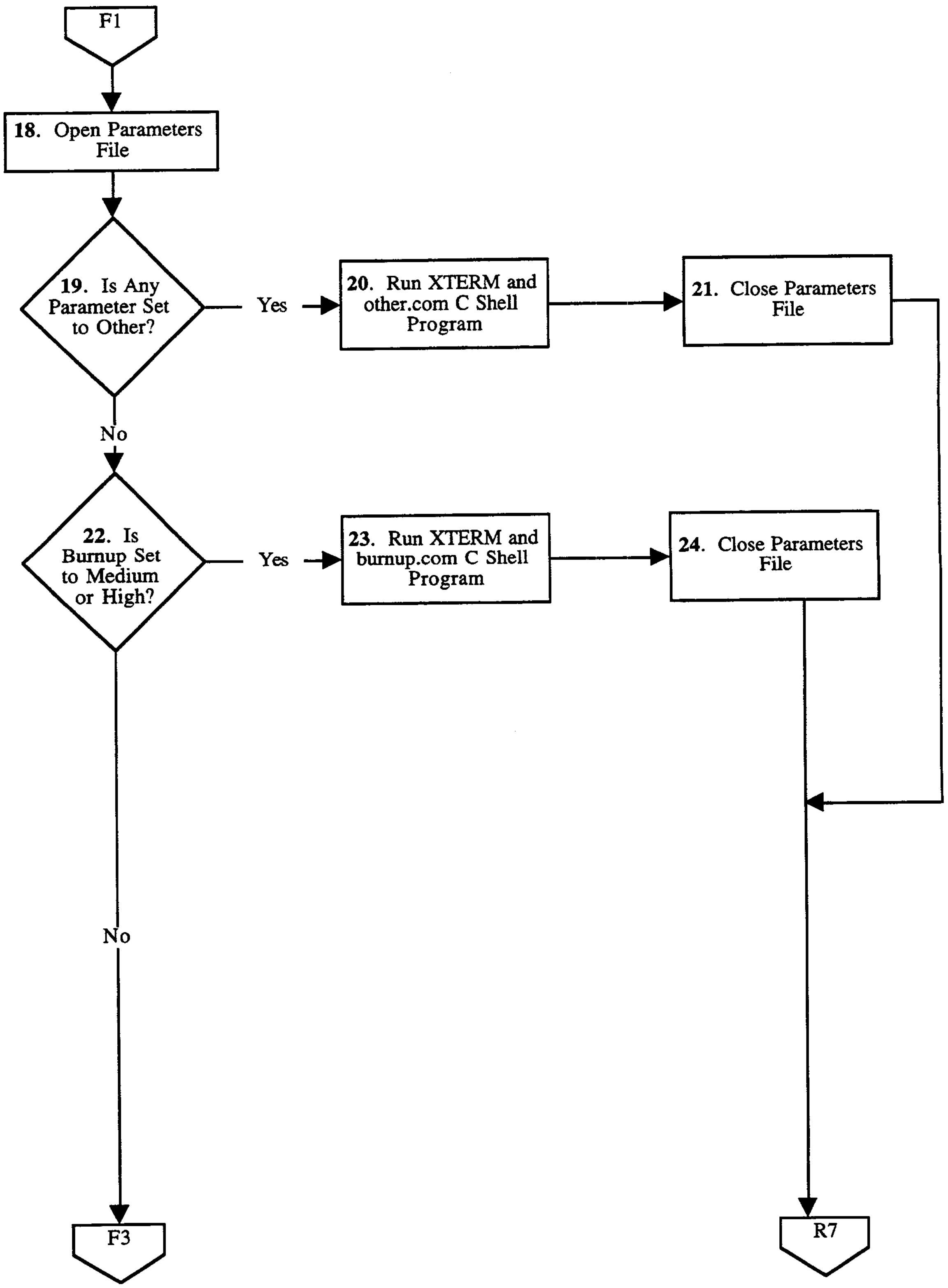

F1
18. Open Parameters File
19. Is Any Parameter Set to Other?
Yes
20. Run XTERM and other.com C Shell Program
21. Close Parameters File
No
22. Is Burnup Set to Medium or High?
Yes
23. Run XTERM and burnup.com C Shell Program
24. Close Parameters File
No
F3
R7

Figure 3-3. (Cont'd). Event handler flowchart for invent

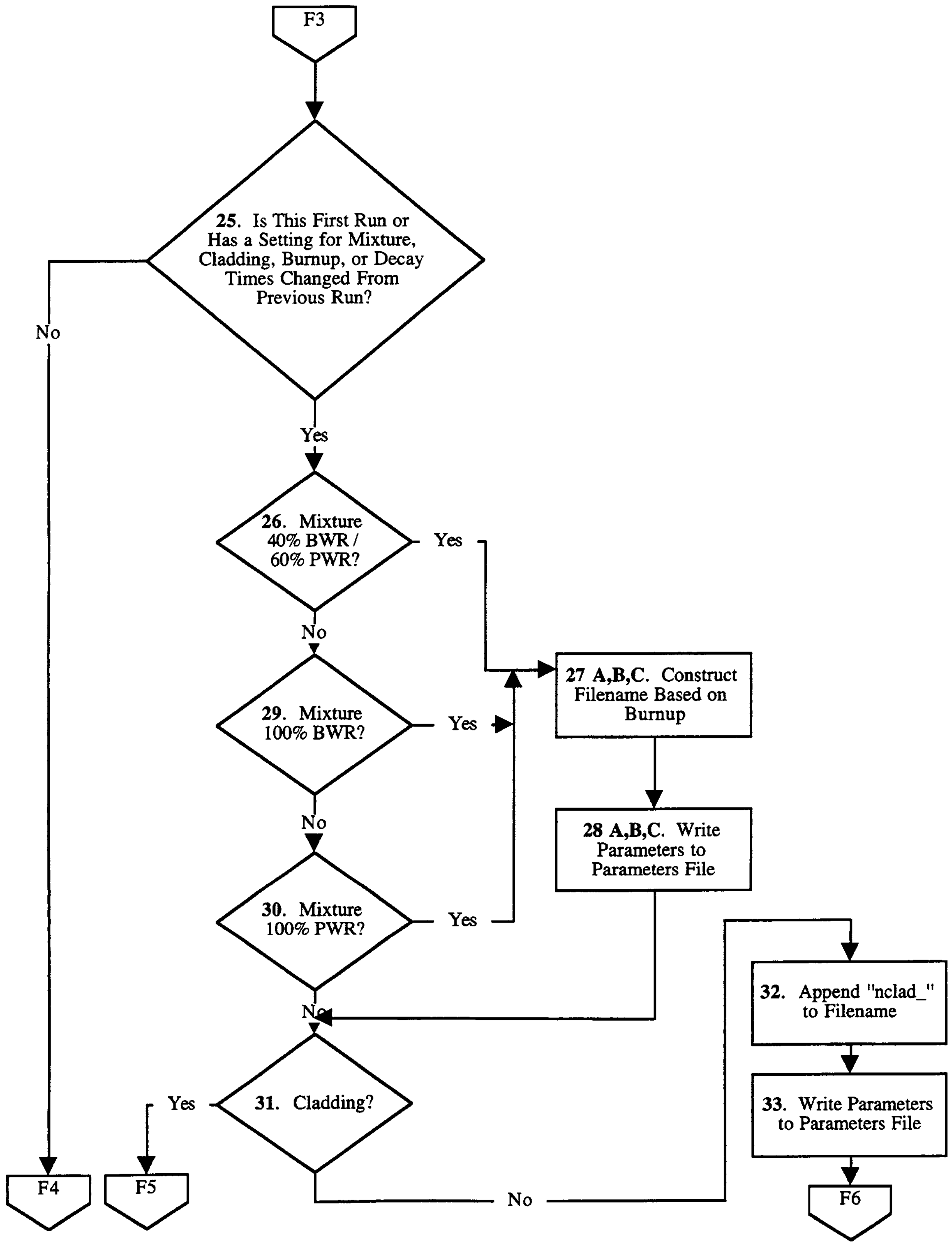

F3
25. Is This First Run or Has a Setting for Mixture, Cladding, Burnup, or Decay Times Changed From Previous Run?
No
Yes
26. Mixture 40% BWR / 60% PWR?
Yes
No
27 A,B,C. Construct Filename Based on Burnup
29. Mixture 100% BWR?
Yes
No
28 A,B,C. Write Parameters to Parameters File
30. Mixture 100% PWR?
Yes
No
32. Append "nclad_" to Filename
31. Cladding?
Yes
33. Write Parameters to Parameters File
F4
F5
No
F6

Figure 3-3 (Cont'd). Event handler flowchart for INVENT

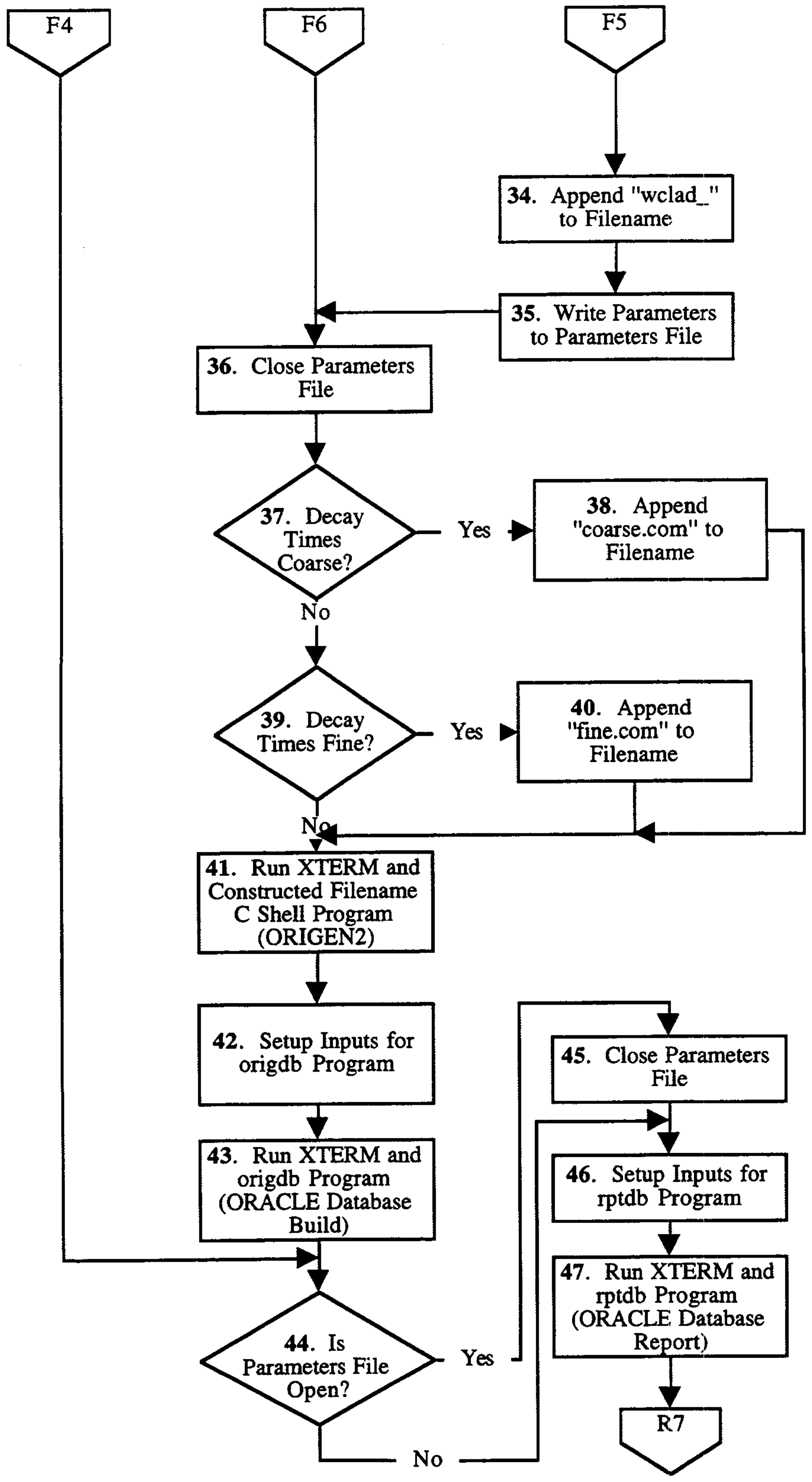

F4
F6
F5
34. Append "wclad_" to Filename
35. Write Parameters to Parameters File
36. Close Parameters File
37. Decay Times Coarse?
Yes
No
38. Append "coarse.com" to Filename
39. Decay Times Fine?
Yes
No
40. Append "fine.com" to Filename
41. Run XTERM and Constructed Filename C Shell Program (ORIGEN2)
42. Setup Inputs for origdb Program
43. Run XTERM and origdb Program (ORACLE Database Build)
44. Is Parameters File Open?
Yes
No
45. Close Parameters File
46. Setup Inputs for rptdb Program
47. Run XTERM and rptdb Program (ORACLE Database Report)
R7

Figure 3-3 (Cont'd). Event handler flowchart for INVENT

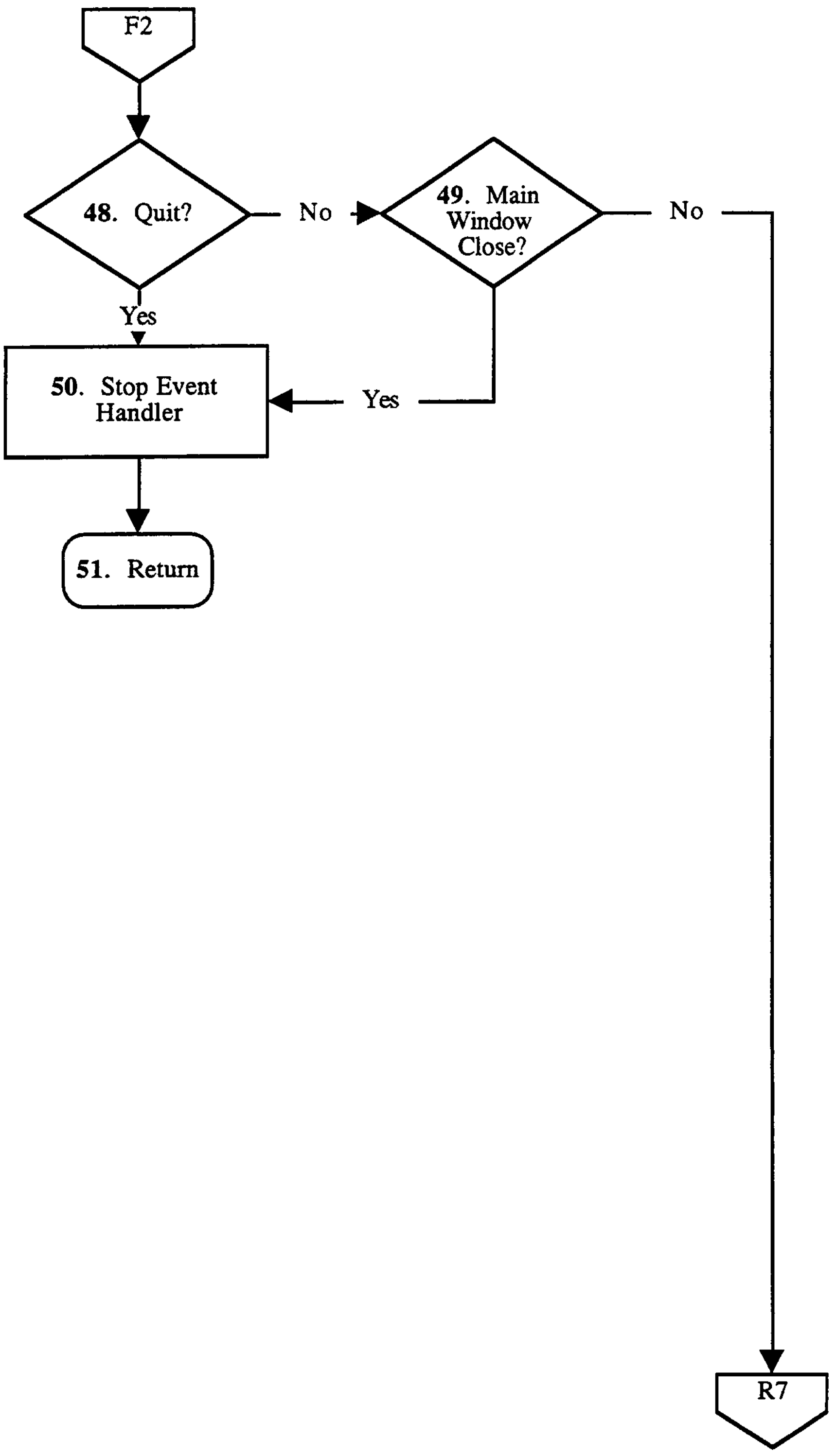


Figure 3-3 (Cont'd). Event handler flowchart for INVENT

3.6 DETAILED DESCRIPTION OF origdb

A listing of the source code for **origdb** is included in Appendix C. The program **origdb** deletes and creates a new ORACLE database using the radionuclide inventory data from the ORIGEN2 output file. The system call is as follows:

```
origdb <ORIGEN2 file> <# of sets> <parameter file>
```

where:

<ORIGEN2 file>	=	the name of the ORIGEN2 output file.
<# of sets>	=	the number of runs executed within the ORIGEN2 program. So far, only 1 (corresponding to coarse) or 7 (corresponding to fine) are valid.
<parameter file>	=	the name of the file in which the parameter information is stored (i.e., BWRburnup, BWRbwrpctoftotfuel, PWRburnup, PWRpwrpctoftotfuel, and Cladding?). The first four parameters in <parameter file> are floating point numbers, while the last parameter is a Y or N.

3.7 DETAILED DESCRIPTION OF rptdb

The program **rptdb** takes a file containing a list of query parameters and generates a final report containing the radionuclides of interests. A listing of the source code for **rptdb** is included in Appendix C. Sample query files and a sample report are included in Appendices F and G, respectively. The system call is as follows:

```
rptdb <query param file> <report type>
```

where <query param file> is the name of the query parameter file to be processed against the database. The file must be in the following format (all parameters must be justified left and capitalized):

```
Nuclide Name(1) or Time(1)
Nuclide Name(2) or Time(2)
.       .
.       .
.       .
Nuclide Name(n) or Time(n)
```

Nuclide Name is the name of the desired radionuclide (capitalized with no spaces).
Time is a number (i.e., 0, 10, 20, 30, etc.).
To report on all elements, place the keyword ALL (capitalized) anywhere within the file.

<report type> is either a 0 (sorted by radionuclide name, time, radionuclide type) or a 1 (sorted by time, radionuclide name, radionuclide type).

3.8 DETAILED DESCRIPTION OF hlgen

The program **hlgen** reads the ORIGEN2 decay.lib library and extracts the halflife information and stores the information in an ORACLE database. This program is considered a utility and supports the INVENT module. The program is run in stand-alone mode by typing **hlgen** from the directory where **hlgen** is located. A listing of the source code is included in Appendix C.

3.9 DETAILED DESCRIPTION OF DIRECTORIES

The following is a listing of the directories and files required to run the INVENT software. ORIGEN2_HOME and REPORTS_HOME are environments variables which are set to the top level directories.

albert/origen2_ifc-**invent** program source
ORIGEN2_HOME/code-ORIGEN codes
ORIGEN2_HOME/libs-ORIGEN libraries (cross-section, decay, etc.)
ORIGEN2_HOME/invent/inputs/bwr-input files
ORIGEN2_HOME/invent/inputs/bwr_pwr-input files
ORIGEN2_HOME/invent/inputs/pwr-input files
ORIGEN2_HOME/invent/inputs/decay-input files
ORIGEN2_HOME/invent-command and input files
ORIGEN2_HOME/invent INVENT command files

REPORTS_HOME/origdb-code to build database
REPORTS_HOME/rptdb-code to generate reports

3.10 SOFTWARE BUILDING PROCEDURE

The software is built by changing to the directory that contains the appropriate source code and then typing "make". The details on how to build the software are contained within UNIX makefiles. Refer to Appendix B for more information.

3.11 FORTRAN SUBROUTINES

This section describes a set of three FORTRAN subroutines which allow the efficient retrieval of radionuclide inventory data as required in the TPA code. It was originally envisioned that the TPA code could directly access the ORACLE database. This option is still available, however, it requires the purchase of an optional pre-compiler from the Oracle Corporation. Because of the relatively straightforward nature of storing and retrieving radionuclide inventory data, it was considered beneficial to abstract the radionuclide inventory data stored in ORACLE, and store the data in a FORTRAN database (i.e., array) with access via FORTRAN callable subroutines. Figure 3-4 describes the FORTRAN subroutines that allow storage and retireval of radionuclide inventory data. The details of the storage of the data is effectively hidden from a user so that the user need only consider the data input/output for the three subroutines. The source code for the three subroutines is included in Appendix H.

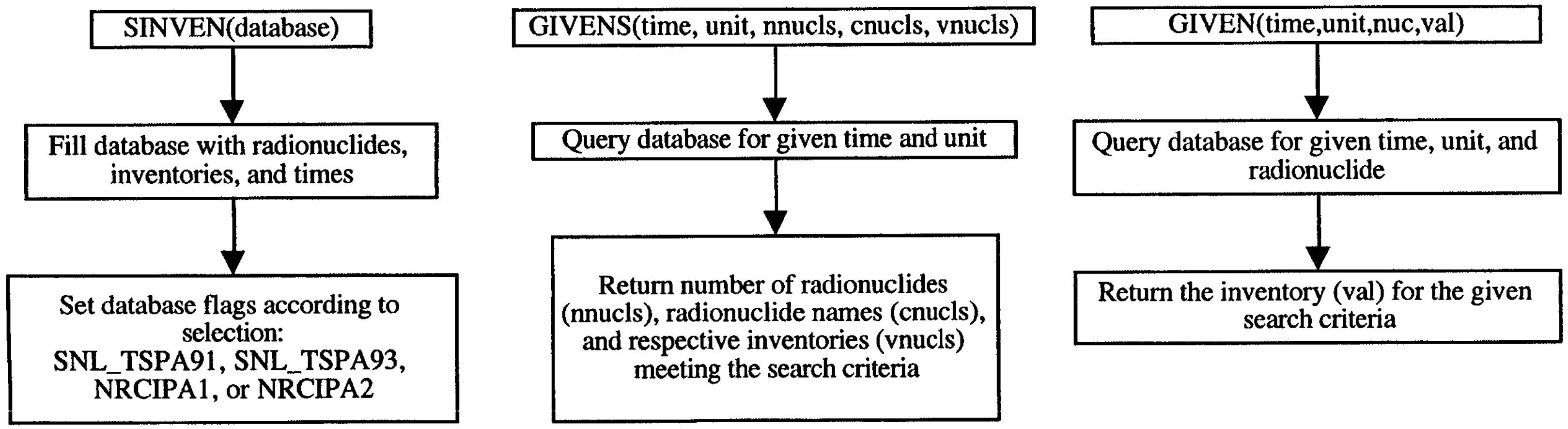

SINVEN(database)
Fill database with radionuclides, inventories, and times
Set database flags according to selection: SNL_TSPA91, SNL_TSPA93, NRCIPA1, or NRCIPA2
GIVENS(time, unit, nnucls, cnucls, vnucls)
Query database for given time and unit
Return number of radionuclides (nnucls), radionuclide names (cnucls), and respective inventories (vnucls) meeting the search criteria
GIVEN(time,unit,nuc,val)
Query database for given time, unit, and radionuclide
Return the inventory (val) for the given search criteria

Figure 3-4. Flowcharts for FORTRAN subroutines which store and retrieve the radionuclide inventory data

3.11.1 Subroutine siven

Subroutine **siven** is called once before either subroutine **givens** or **given** is called. Subroutine **siven** is passed one argument (character*10) which is used to select which database will be utilized. The argument must either be 'SNL_TSPA91', 'SNL_TSPA93', 'NRC_IPA1', or 'NRC_IPA2'. Based on the argument, **siven** initializes the database with the radionuclides used in either the SNL TSPA-91 report (Barnard et al., 1992), SNL TSPA-93 report (Wilson et al., 1994), NRC IPA-1 report (Codell et al., 1992), or NRC IPA-2 report (Wescott et al., 1994). The database containes the curie content per MTU for up to 43 distinct radionuclides (see Figure 2-2, Table 5-1, or Appendix F) for 73 distinct times (from 10 up to 1,000,000 years, see Figure 2-2). In order to simulate the SNL TSPA-93 inventories, the SNL TSPA-91 values are scaled by a factor as discussed in the report by Wilson et al. (1994).

3.11.2 Subroutine given

Subroutine **given** can be called after the database has been selected and initialized (by calling **siven**). The input and output variables for **given** include:

time	=	time of interest
unit	=	units for time (character*2, either 'S' or 'YR')
nuc	=	radionuclide of interest (character*10)
val	=	inventory (Ci/MTU)

The radionuclide name in nuc should be one of those shown in Figure 2-2. The calling routine specifies time, unit, and nuc, and the subroutine outputs val. The values of radionuclide inventories (measured in Ci/MTU) as a function of time is based on the values calculated at discrete times. An interpolation is used to evaluate the inventory for times between data points (this interpolation is based on the one suggested in Department of Energy, 1987).

$$C = C_0 \left(\frac{t}{t_0} \right)^p \tag{3-1}$$

where

C	=	inventory (Ci/MTU) at time t
Co	=	inventory (Ci/MTU) at time t_o
t	=	time of interest (same as time described above)
t_o	=	time which is nearest and smaller than t
p	=	power of interpolation.

The power of interpolation, p, is calculated to match the inventory value at the nearest time value greater than t (i.e., $t_o < t < t_1$).

$$p = \ln(C_0/C_1) \;/\; \ln(t_0/t_1) \tag{3-2}$$

A warning message will be printed if the input time is outside the range of stored time values.

3.11.3 Subroutine givens

Subroutine **givens** can be called after the database has been selected and initialized (by calling **siven**). The only difference between **givens** and **given** is that **givens** returns the inventories for all of the selected radionuclides. The input and output variable for **givens** include:

time	=	time of interest
unit	=	units for time (character*2, either 'S' or 'YR')
nnucls	=	number of radionuclides
cnucls	=	array of radionuclide names (character*10)
vnucls	=	array of inventory (Ci/MTU) of radionuclides

The subroutine considers 'time' and 'unit' as inputs, and returns 'nnucls,' 'cnucls,' and 'vnucls' as outputs. The source code for **givens** is included in Appendix H.

4 DESCRIPTION OF THE EXTERNAL SOFTWARE REQUIRED BY THE INVENT MODULE

4.1 GALAXY

The GALAXY software was selected to develop a GUI that would be user-friendly, as well as simplify the data entry process. This use of a GUI is consistent with early recommendations from the IPA Phase 2 activity. The primary advantages of using a GUI, rather than manual creation of input files, are three-fold: (i) ease of use, (ii) high accuracy and consistency, and (iii) greater functionality of the software.

In the following discussion, it is convenient to distinguish between a platform and a windowing system. The term platform can be used in several ways. In a hardware context, platform refers to the brand and model of a particular machine. In a software context, it means the operating system. The term as used in this document refers to a unique pairing of both meanings, that is, a platform is a given machine from a vendor with a given operating system. A windowing system can be one of several offered by computer vendors, such as Macintosh System 7, Microsoft Windows, SUN OPEN LOOK, or Motif.

The GALAXY tool can make a program user-friendly by providing a GUI that is compatible with many windowing systems. The GUI technique is becoming the most popular input mechanism for software control and is packaged with virtually every commercial application. GUI is said to be most effective when interfacing with users who are not experts in the domain of the application being exercised. While being user friendly, GALAXY can also provide software libraries that permit the porting of the software to several other platforms and window systems with no software changes. GALAXY is a one-for-one replacement for the Macintosh Toolbox, Windows SDK, Motif Toolkit, and OPEN LOOK Toolkits.

A disadvantage to this approach is that most windowing tools require implementation in the C programming language, which is typically not found on many scientific user platforms. However, its availability is increasing on scientific workstation platforms even though it may not be used by the scientist. In the near future, most scientists will have access to a workstation that can host the C language even though it may not be on their desks. Having the C language on the desk is not required to receive the benefits of the GUI environment. A machine without the C language requires only a window server with a network connection to a machine that can host the language. At modern network speeds, virtually the same capability can be provided with less burden on the user.

The application source code is compiled as usual using the native C or C++ compiler and the GALAXY include files. The native linker is used to bind the GALAXY libraries to the application code. The executable program can then use the GALAXY resource mechanism, which is one of the major facilitators of multiplatform operation. In a GUI environment, the graphical characteristics of a window need not be hardcoded in the program; rather they may be kept in volatile storage and changed by either the program or the user with the graphical input device (mouse). When a GALAXY application is run, it can reference any number of resource files to obtain its noncode resources. These resources can include such items as dialog windows, text, color images, and geometry management information. The resource files are stored in a compressed binary format that is machine independent. The GALAXY Resource Builder is a tool for constructing and editing the user interface portion of a given application.

This resource building toolkit can permit the application to be portable while taking advantage of the capabilities of each target platform. Building portable applications reduces development costs, eases maintenance, and simplifies the product development. Although more code is required, compared to a vendor specific window system, the effort is worth the benefits of a portable application.

The use of GALAXY for this application was successful since it demonstrated that the windows approach to the control of modeling codes is a workable and accurate technique, and is one that minimizes typing while providing a natural communication link to the program.

4.2 ORIGEN2

ORIGEN2 is a computer code developed to model the composition and characteristics of various kinds of spent nuclear fuels as a function of burnup and age (Croff, 1983). To do this, the code performs two major computational functions, isotope generation and isotope depletion, both within the core of an operating reactor and after shutdown. The name derives from "Oak Ridge Isotope Generation and Depletion Code." (The original version was called ORIGEN and a later, improved version is called ORIGEN2.) There is also another improved version called ORIGEN-S. All three versions perform the two basic functions cited above.

Generation: This refers to the generation of individual radionuclides resulting from neutron-induced fission or from neutron capture reactions or other transmutation reactions.

Depletion: This refers to the depletion (and concurrent buildup) of radionuclides resulting from natural decay processes.

Both functions can deal with the approximately 1,400 relevant radionuclides included in the ORIGEN2 database. Many of these radionuclides have very short half-lives and exist only within the reactor core or as transient intermediates in a decay chain. Neutron-induced fission and transmutation are highly dependent on both neutron flux and neutron energy. Neutron flux itself depends on the number of induced fissions. Neutron energy depends on the moderators that are present. The cross-sections (propensity for neutron interaction) are highly sensitive to neutron energy. Decay processes are governed by invariant constants: the half-lives and branching ratios. In a real-time situation, some transmutation may occur while decay is also taking place.

Thus, during generation there is a competition between transmutation and decay that is governed by the relative magnitudes of the cross-sections and the decay constants, with the former highly sensitive to the neutron moderation within the particular system being modeled. This relationship is handled mathematically by calculating a set of effective cross-sections that are applicable to the reactor scenario being modeled.

This calculation is accomplished by first computing a set of effective one-group cross-sections, that is, a weighted-average value for each radionuclide that is appropriate for the moderated neutron energies in that particular reactor model. After this effective cross-section library has been developed, it can be used for variations on that particular reactor model. For example, ORIGEN2 cross-section libraries have been developed specifically for Light Water Reactors (LWRs). Computing one-group cross-sections requires different models and libraries for PWRs and BWRs, and also for standard burnup and high burnup fuel designs. For a given LWR case, the appropriate model must be used. These cross-sections have recently been recalculated (Ludwig and Renier, 1989) for these conditions:

PWR standard burnup: 33,000 MWd/MTIHM;
PWR extended burnup: 50,000 MWd/MTIHM;
BWR standard burnup: 27,500 MWd/MTIHM; and
BWR extended burnup: 40,000 MWd/MTIHM.

For the above cross-section libraries, initial enrichments and cycle conditions can be varied. Also, variations in structural materials (both quantities and compositions) may be used if appropriate. These factors have been examined in a series of sensitivity studies (Welch et al., 1992). It was clearly shown that enrichment is a major factor in these calculations, especially for the actinides (and their derivative properties, such as neutron emission strength). For this reason, enrichments were handled in a very definitive manner, with enrichment tailored to burnup, and with ranges on either side to allow for normal variations. Cycle conditions were modeled after average conditions derived from utility data.

The neutron fluxes and energies, and the effective cross-sections, have been developed for the reactor core region, where the fuel resides, along with the fuel cladding and some of the assembly hardware. However, much of the hardware is outside the core region, in which the flux falls off very rapidly with distance, and moderation is also quite different. These latter effects have a profound influence on the quantities of activation products formed within these hardware components.

The major steps required in ORIGEN2 modeling and computation are shown on the schematic of Figure 4-1. The center column, boxes 1 through 4, are the major steps involved in running the ORIGEN2 code. Box 2 requires complex, multidimensional computations that apply to the generic reactor model defined in box 1; once done, the resulting library is used for all specific cases for that generic reactor model. Boxes A and B are independent data libraries utilized by ORIGEN2. Box X is the input data required to conduct a run for specific cases; this includes initial enrichment and cycle specifications, along with the desired output data and format. In the past, the near-core region was modeled using an approximate method to calculate the inventory associated with activation metals. With the increased interest in activated metal, this aspect was reexamined. Both calculations and experimental measurements at Pacific Northwest Laboratory (PNL) (Luksic, 1989) provided new factors, relative to the core region, for the top, plenum, and bottom zones. These new factors were utilized.

The basic data output from ORIGEN2 is in terms of the gram atoms of each radionuclide, as specified times. From this, the derived quantities and radiological properties are calculated, including thermal output, alpha activity, beta and gamma activities, neutron production from spontaneous fission and from alpha-induced reaction, and the photon energy spectra. These quantities can be provided for individual radionuclides or elements or for all radionuclides or elements within one or more of the three major categories: fission products, activation products, and actinides. Each group also includes decay daughters. The fission products derive from nuclear fission of fissionable isotopes. The activation products are generated by neutron activations of structural materials and components. The actinides derive from neutron capture (often multiple or sequential neutron captures) of the initially-present heavy metal isotopes (mainly U-238 and U-235 for LWRs).

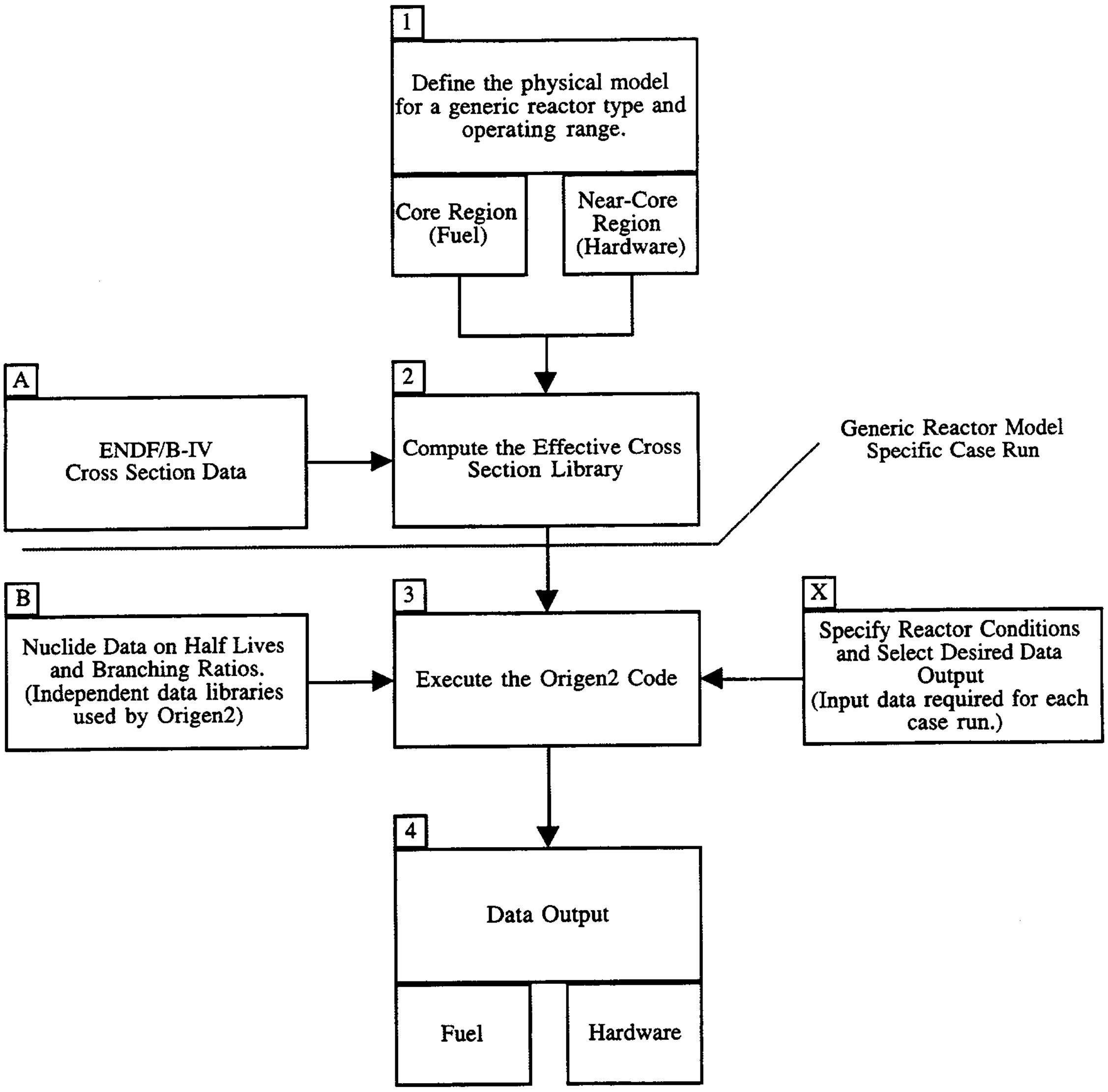

1
Define the physical model for a generic reactor type and operating range.
Core Region (Fuel)
Near-Core Region (Hardware)
A
ENDF/B-IV Cross Section Data
2
Compute the Effective Cross Section Library
Generic Reactor Model
Specific Case Run
B
Nuclide Data on Half Lives and Branching Ratios. (Independent data libraries used by Origen2)
3
Execute the Origen2 Code
X
Specify Reactor Conditions and Select Desired Data Output (Input data required for each case run.)
4
Data Output
Fuel
Hardware

Figure 4-1. Steps in ORIGEN2 modeling

4.3 ORACLE

The ORACLE database management system (DBMS) was selected to investigate the effectiveness of a relational database as a storage and retrieval system for model input parameters associated with the IPA task. Database systems with similar features have the advantage of sharing a single database among several users on different platforms, and may facilitate the use of a distributed data set by a single user. The ORACLE DBMS also permits the access of other vendor data sets, but this is of limited use to the IPA effort since all data will be generated internal to the project.

As implemented, one physical quantity was chosen to be stored as a function of time and radionuclide. User access to the database is achieved using the Structured Query Language (SQL). The term query is slightly misleading in this usage, since in its most familiar form it connotates a human activity. However, SQL involves the communication between software modules rather than the direct participation of the user. The version of SQL used was SQL*Plus Version 3.0.12.4.1.

ORACLE SQL requires interface modules to be written in a high-level language such as C or FORTRAN. These routines, in turn, must interface with the user to determine the exact request for data and the desired output form. Program **origdb** provided the database update function and program **rptdb** provided the database query function. Programs **origdb** and **rptdb** are described in Sections 3.5 and 3.6 in this document.

The power of the ORACLE DBMS is in its ability to update individual entries of a dynamic database while simultaneously satisfying query requests from users. These conditions are frequently found in applications such as airline reservation systems and bank transaction systems. However, these assume a different environment than the IPA system. The IPA database will be very static and requires minimal queries relative to what is commonly addressed with SQLs.

5 VERIFICATION AND VALIDATION STATUS

The INVENT module runs in stand-alone mode and a set of FORTRAN subroutines have been written to be used in the TPA system of codes. The stand-alone version can be used to change the characteristics and burnup for the spent fuel, which then affects the radionulicde inventories. After a user has selected a set of fuel characteristics, the stand-alone version can be used to update the FORTRAN subroutines which are designed to be repeatedly queried in the TPA program (however, the update process has not been automated and will require additional effort).

The INVENT module was developed on a Sun Workstation. The input parameters to the ORIGEN2 code are determined by **invent**, the main program, from information supplied by the user through the GUI, GALAXY windows. The detailed calculation of radionuclide inventories was performed by ORIGEN2, and there were no changes to the ORIGEN2 code which was received from ORNL. The results from ORIGEN2 are postprocessed and stored in an ORACLE database.

Verification of the code was performed by comparing the INVENT results at 10 years of cooling time with the results used in the NRC IPA-2 activities (Wescott et al., 1994), published in the SNL TSPA-91 report (Barnard et al., 1992), and those published in the SNL TSPA-93 report (Wilson et al., 1994). The comparison is summarized in Table 5-1. The NRC IPA-2 and SNL TSPA-91 results are based on spent fuel only, while the SNL TSPA-93 results are based on both spent fuel and other HLW. (The SNL TSPA-93 waste mixture is based on 63,000 MTIHM of spent fuel and 7,000 MTIHM from sources such as Savannah River, SC, Hanford, WA, Idaho Chemical Processing Plant, ID, and West Valley, NY.) The results generated by **invent** (via ORIGEN2) are very close to values published in the SNL TSPA-91 report. Differences exist with the SNL TSPA-93 report because of the addition of other wastes (in addition to spent fuel). The good comparison of radionuclide inventories are indications that the **invent** module is working properly.

Verification of the FORTRAN subroutines was performed by plotting the radionuclide inventories for times from 10 to 1,000,000 yr, see Figure 5-1. The trends in the inventories was compared with those published elsewhere (e.g., Roxburgh, 1987). As expected, some radionuclide inventories continuously decrease with increasing time, some remain relatively constant over long periods of time (those with long half-lives), and some increase with time (daughters in a decay chain). For example, Pu-238 has a 87.7 yr half-life and its inventory can be observed to continuously decrease with time. Another example is U-234, which has a 244,500 yr half-life, and remains relatively constant up to ~100,000 yr. An example of daughter ingrowth can be seen by Th-230, Ra-226, and Pb-210 which are in the U-238 decay series of radionuclides. Hence, the inventories of the daughters increases with increasing time. Based on these plots of radionuclide inventories, the FORTRAN subroutines were noted to be working properly.

Table 5-1. Comparison of radionuclide inventories used in Nuclear Regulatory Commission IPA-2 (spent fuel only), Sandia National Laboratories TSPA-91 (spent fuel only), Sandia National Laboratories TSPA-93 (combined spent fuel and high-level waste), and invent (spent fuel only)

Radionuclide	NRC IPA-2 Inventory at 10 yr (Ci/MTU)	SNL TSPA-91 Inventory at 10 yr (Ci/MTU)	SNL TSPA-93 Inventory at 10 yr (Ci/MTU)	invent Inventory at 10 yr (Ci/MTU)
U-238	3.18×10^{-1}	3.18×10^{-1}	3.15×10^{-1}	3.19×10^{-1}
Cm-246	2.58×10^{-2}	2.58×10^{-2}	3.22×10^{-2}	2.56×10^{-2}
Pu-242	—	1.60×10^{0}	1.74×10^{0}	1.60×10^{0}
Am-242m	—	7.46×10^{0}	9.62×10^{0}	7.49×10^{0}
Pu-238	—	2.12×10^{3}	2.43×10^{3}	2.11×10^{3}
U-234	1.89×10^{0}	1.13×10^{0}	1.51×10^{0}	1.13×10^{0}
Th-230	1.29×10^{-4}	1.29×10^{-4}	3.65×10^{-4}	1.29×10^{-4}
Ra-226	3.67×10^{-7}	3.67×10^{-7}	2.22×10^{-6}	3.67×10^{-7}
Pb-210	4.71×10^{-8}	4.71×10^{-8}	5.36×10^{-7}	4.72×10^{-8}
Cm-243	—	1.54×10^{1}	1.30×10^{1}	1.53×10^{1}
Am-243	1.55×10^{1}	1.55×10^{1}	1.71×10^{1}	1.54×10^{1}
Pu-239	3.08×10^{2}	3.08×10^{2}	3.33×10^{2}	3.08×10^{2}
U-235	—	1.68×10^{-2}	2.28×10^{-2}	1.69×10^{-2}
Pa-231	—	1.94×10^{-5}	3.72×10^{-5}	1.95×10^{-5}
Ac-227	—	5.19×10^{-6}	1.80×10^{-5}	5.19×10^{-6}
Cm-245	1.26×10^{-1}	1.26×10^{-1}	1.64×10^{-1}	1.25×10^{-1}
Pu-241	—	7.43×10^{4}	4.07×10^{4}	7.44×10^{4}
Am-241	1.64×10^{3}	1.64×10^{3}	3.26×10^{3}	1.64×10^{3}
Np-237	2.88×10^{-1}	2.88×10^{-1}	3.78×10^{-1}	2.87×10^{-1}
U-233	—	2.54×10^{-5}	5.77×10^{-5}	2.40×10^{-5}
Th-229	—	1.40×10^{-7}	3.10×10^{-7}	1.39×10^{-7}
Cm-244	—	1.15×10^{3}	7.62×10^{2}	1.15×10^{3}
Pu-240	5.08×10^{2}	5.08×10^{2}	5.09×10^{2}	5.08×10^{2}

Table 5-1 (Cont'd). Comparison of radionuclide inventories used in Nuclear Regulatory Commission IPA-2 (spent fuel only), Sandia National Laboratories TSPA-91 (spent fuel only), Sandia National Laboratories TSPA-93 (combined spent fuel and high-level waste), and invent (spent fuel only)

Radionuclide	NRC IPA-2 Inventory at 10 yr (Ci/MTU)	SNL TSPA-91 Inventory at 10 yr (Ci/MTU)	SNL TSPA-93 Inventory at 10 yr (Ci/MTU)	invent Inventory at 10 yr (Ci/MTU)
U-236	—	2.40×10^{-1}	3.12×10^{-1}	2.40×10^{-1}
U-232	—	2.50×10^{-2}	3.32×10^{-2}	2.48×10^{-2}
Sm-151	—	3.18×10^{2}	3.36×10^{2}	3.18×10^{2}
Cs-137	7.66×10^{4}	7.66×10^{4}	6.33×10^{4}	7.64×10^{4}
Cs-135	3.50×10^{-1}	3.50×10^{-1}	4.79×10^{-1}	3.51×10^{-1}
I-129	2.95×10^{-2}	2.95×10^{-2}	3.39×10^{-2}	2.95×10^{-2}
Sn-126	—	7.17×10^{-1}	8.04×10^{-1}	7.16×10^{-1}
Sn-121m	—	9.04×10^{-1}	7.29×10^{-1}	7.97×10^{-1}
Ag-108m	—	1.19×10^{-2}	1.08×10^{-2}	1.20×10^{-2}
Pd-107	—	1.05×10^{-1}	1.14×10^{-1}	1.05×10^{-1}
Tc-99	1.23×10^{1}	1.23×10^{1}	1.43×10^{1}	1.23×10^{1}
Mo-93	—	1.60×10^{-2}	1.80×10^{-2}	1.01×10^{-2}
Nb-94	7.93×10^{-1}	7.93×10^{-1}	9.22×10^{-1}	5.04×10^{-1}
Zr-93	—	1.88×10^{0}	2.23×10^{0}	1.85×10^{0}
Sr-90	—	5.32×10^{4}	4.51×10^{4}	5.31×10^{4}
Se-79	3.81×10^{-1}	3.81×10^{-1}	4.53×10^{-1}	3.80×10^{-1}
Ni-63	—	4.55×10^{2}	3.95×10^{2}	3.05×10^{2}
Ni-59	3.56×10^{0}	3.56×10^{0}	3.08×10^{0}	2.46×10^{0}
Cl-36	—	1.19×10^{-2}	1.10×10^{-2}	1.17×10^{-2}
C-14	1.54×10^{0}	1.54×10^{0}	1.45×10^{0}	1.33×10^{0}

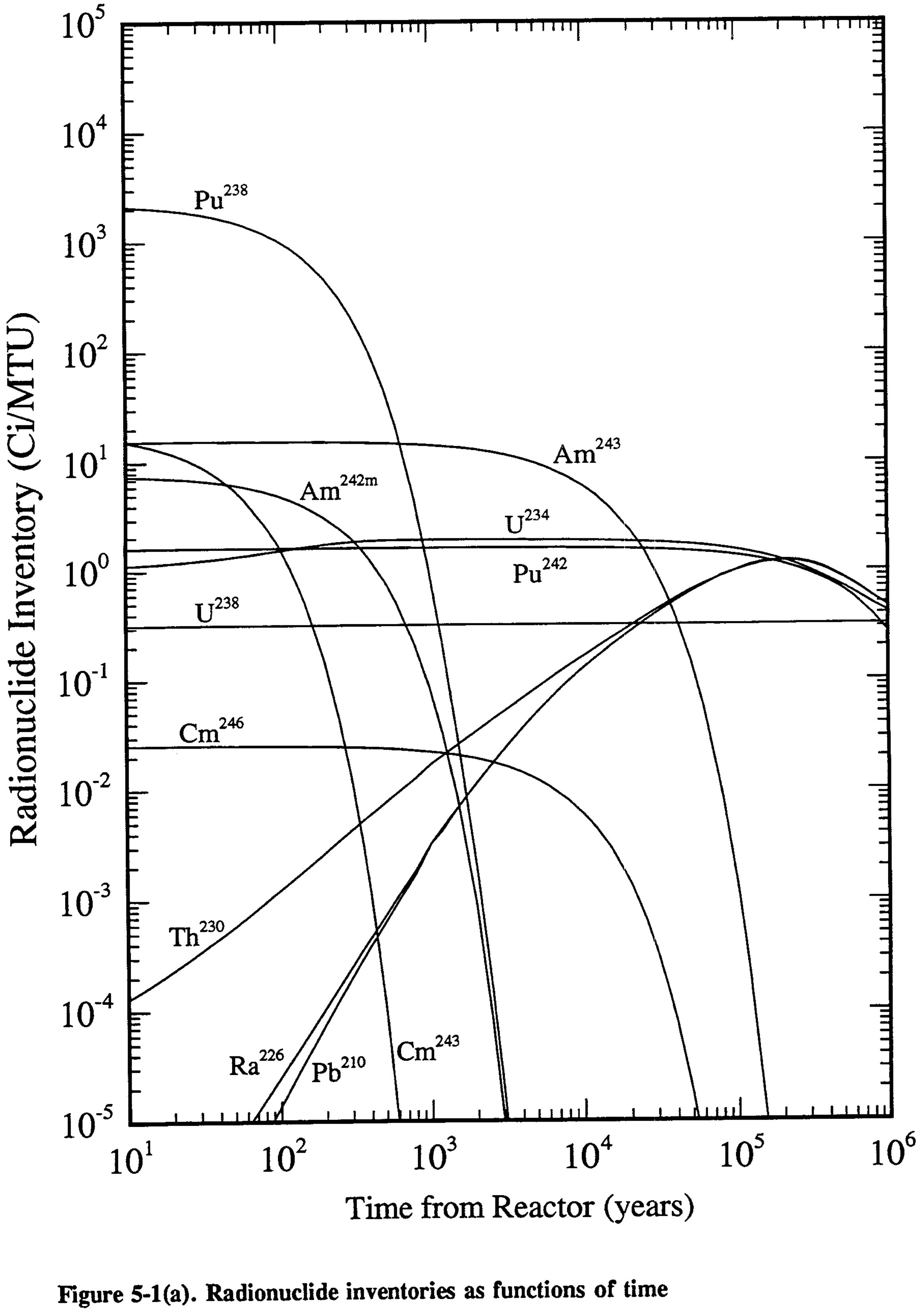


Figure 5-1(a). Radionuclide inventories as functions of time

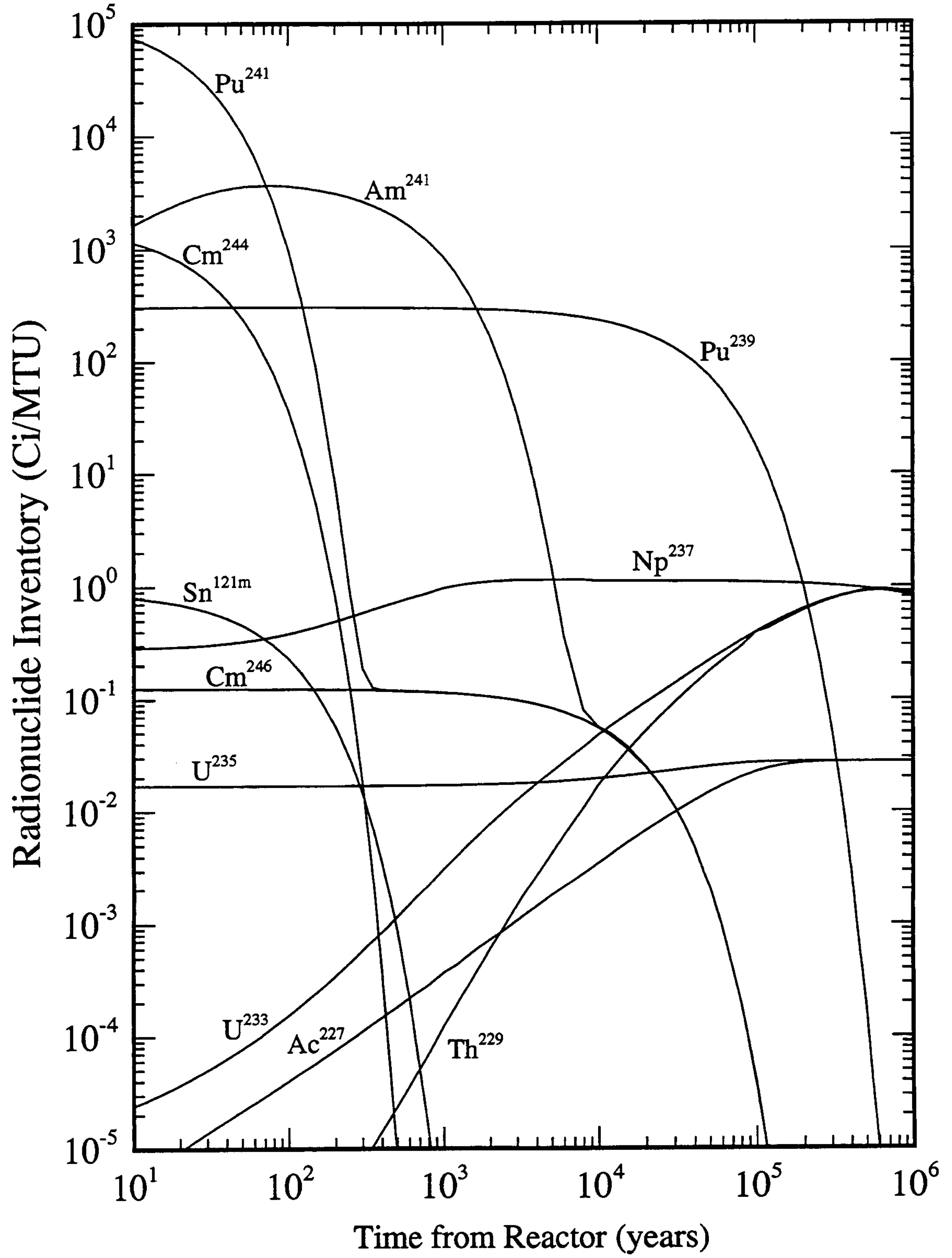


Figure 5-1(b). Radionuclide inventories as functions of time

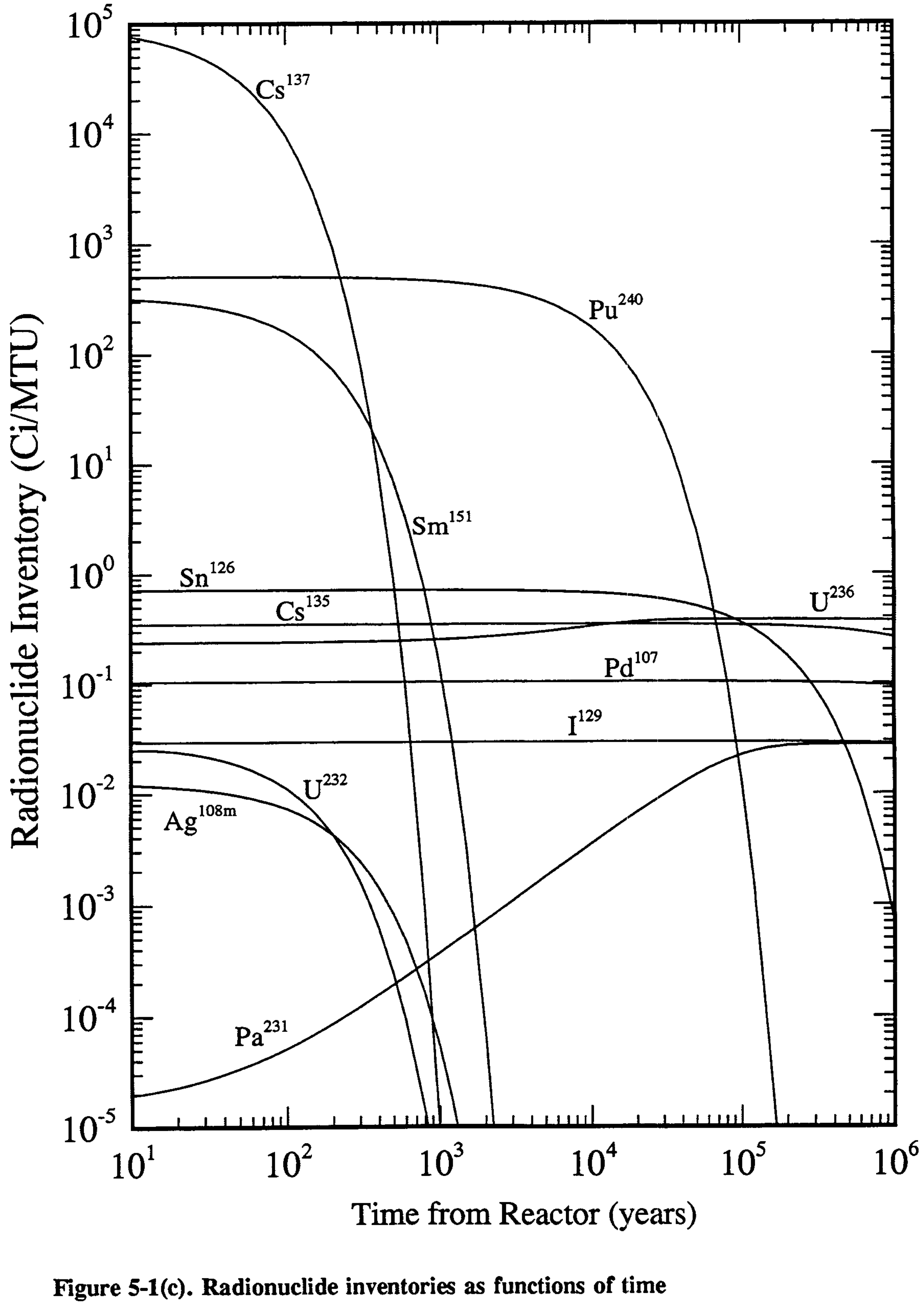


Figure 5-1(c). Radionuclide inventories as functions of time

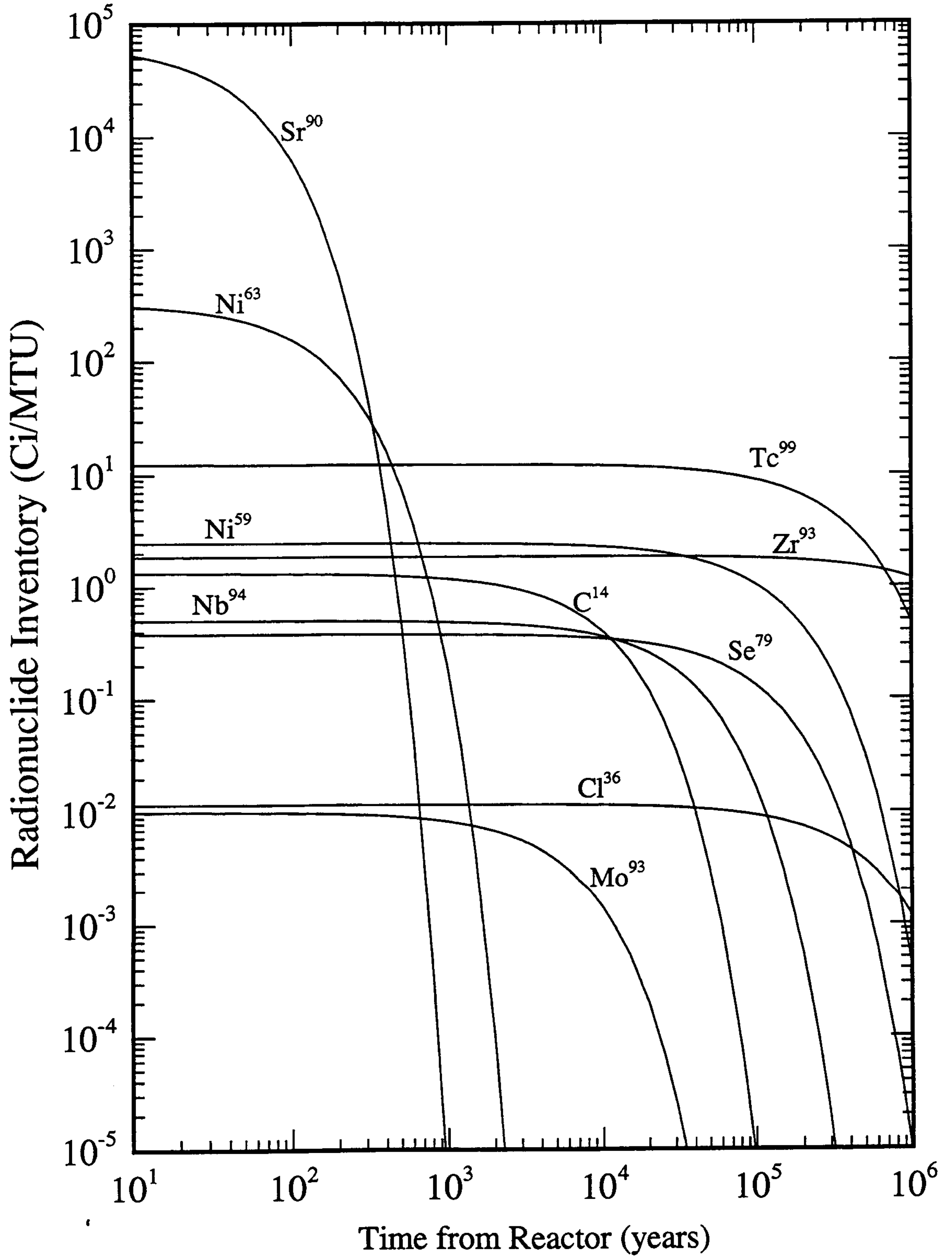


Figure 5-1(d). Radionuclide inventories as functions of time

6 REFERENCES

Barnard, R.W., M.L. Wilson, H.A. Dockery, J.W. Gauthier, P.G. Kaplan, R.R. Easton, F.W. Bingham, and T.H. Robey. 1992. *TSPA 1991: An Initial Total-System Performance Assessment for Yucca Mountain.* SAND 91-2795. Albuquerque, NM: Sandia National Laboratories.

Codell, R., N. Eisenberg, D. Fehringer, W. Ford, T. Margulies, T. McCartin, J. Park, and J. Randall. 1992. *Initial Demonstration of the NRC's Capability to Conduct a Performance Assessment for a High-Level Waste Repository.* NUREG-1327. Washington, DC: Nuclear Regulatory Commission.

Croff, A.G. 1983. ORIGEN2: A Versatile Computer Code for Calculating the Nuclide Compositions and Characteristics of Nuclear Materials. *Nuclear Technology* 62: 335–352.

Department of Energy. 1987. *Characteristics of Spent Fuel, High-Level Waste, and Other Radioactive Wastes Which May Require Long-Term Isolation.* DOE/RW-0184. Washington, DC: U.S. Department of Energy.

Luksic, A. 1989. *Spent Fuel Assembly Hardware: Characterization and 10 CFR 61 Classification for Waste Disposal.* PNL-6906, Vol. 1. Richland, WA: Pacific Northwest Laboratory.

Ludwig, S.B., and J.P. Renier. 1989. *Standard and Extended-Burnup PWR and BWR Reactor Models for the ORIGEN2 Computer Code.* ORNL/TM-11018. Oak Ridge, TN: Oak Ridge National Laboratory.

Nuclear Regulatory Commission. 1991. *Nuclear Regulatory Legislation.* NUREG-0980, Volume 1(1). Washington, DC: Nuclear Regulatory Commission. 227–319.

Roxburgh, I.S. 1987. *Geology of High-Level Nuclear Waste Disposal, An Introduction.* New York, NY: Chapman and Hall.

Sagar, B., and R.W. Janetzke. 1993. *Total-System Performance Assessment (TPA) Computer Code: Description of Executive Module, Version 2.0.* CNWRA 93-017. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Welch, T.D., K.J. Notz, and R.J. Andermann. 1992. *ORIGEN2 Sensitivity to Enrichment and Other Factors.* ORNL/TM-11333. Oak Ridge, TN: Oak Ridge National Laboratory.

Wescott, R.G., M.P. Lee, N.A. Eisenberg, and T.J. McCartin, (eds.). 1994. *Phase 2 Demonstration of the NRC's Capability to Conduct a Performance Assessment for a High-Level Waste Repository.* NUREG-1464. Washington, DC: Nuclear Regulatory Commission

Wilson, M.L., and J.H. Gauthier, R.W. Barnard, G.e. Barr, H.A. Dockery, E. Dunn, R.R. Eaton, D.C. Guerin, N. Lu, M.J. Martinez, R. Nilson, C.A. Rautman, T.H. Robey, B. Ross, E.E. Ryder, A.R. Schenker, S.A. Shannon, L.H. Skinner, W.G. Halsey, J.D. Gansemer, L.C. Lewis, A.D. Lamont, I.R. Triay, A. Meijer, and D.E. Morris. 1994. *Total-System Performance Assessment for Yucca Mountain—SNL Second Iteration (TSPA-1993).* SAND 93-2675. Albuquerque, NM: Sandia National Laboratories.

APPENDIX A

SETUP FILES (.cshrc)

SETUP FILES (.cshrc)

Workstation programs are quite complex and require much setup information. Default conditions are usually assumed in order that a user not be forced to specify every possible parameter. Some parameters assist the program with information about the computer system configuration and file layout. Assistance from the system manager is usually required to properly setup this file. This .cshrc file is the one which the system manager provided with the exception of the environment variables REPORTS_HOME, ORIGEN2_HOME, ORACLE_SID, and ORAENV_ASK.

See microfiche for text of Appendix A, Setup Files (.cshrc).

APPENDIX B

Makefiles

Makefiles

Many programs are composed of several functions. Makefiles relieve the software developer from having to remember all the functions that are necessary to build a program. The Makefile provides instructions to the make utility and usually resides in the same directory as the source code. Instructions list the functions that are needed and the libraries which should be searched to make a program. The make utility also looks at the age of files to determine if compiling is necessary or if linking will suffice. The programs are made by changing directory to where the appropriate source code is located and typing **make**. For more information, refer to readily available documentation on Makefiles under UNIX.

See microfiche for text of Appendix B, Makefiles.

APPENDIX C

LISTING OF C LANGUAGE SOURCE CODES:
invent, origdb, rptdb, and hlgen

LISTING OF C LANGUAGE SOURCE CODES: invent, origdb, rptdb, and hlgen

See microfiche for text of Appendix C, Listing of C Language Source Codes: **invent**, **origdb**, **rptdb**, and **hlgen**.

APPENDIX D

ORIGEN2 Command Files

ORIGEN2 Command Files

Command files are C shell programs. These programs are batch files to csh and perform system functions such as copying, renaming, and deleting of files, as well as controlling program execution.

See microfiche for text of Appendix D, Command Files.

APPENDIX E

ORIGEN2 Input Files

ORIGEN2 Input Files

The files described in this section are combined with either the coarse or fine file to construct an ORIGEN2 input file.

See microfiche for text of Appendix E, ORIGEN2 Input Files.

APPENDIX F

FOUR USER SELECTABLE LISTS FOR ORACLE QUERIES

FOUR USER SELECTABLE LISTS FOR ORACLE QUERIES

See microfiche for text of Appendix F, Four User Selectable Lists for ORACLE Queries.

APPENDIX G

EXAMPLE OF REPORT FROM INVENT MODULE

EXAMPLE OF REPORT FROM INVENT MODULE

See microfiche for text of Appendix G, Example of Report from INVENT Module.

APPENDIX H

LISTING OF FORTRAN SUBROUTINES: siven, givens, given

LISTING OF FORTRAN SUBROUTINES: siven, givens, given

This appendix contains the source code of the three FORTRAN subroutines: **siven**, **givens**, and **given**.

See microfiche for text of Appendix H, Listing of FORTRAN Subroutines: **siven**, **givens**, and **given**.