

**Idaho
National
Engineering
Laboratory**

INEL-96/0057

RECEIVED

OCT 07 1996

OSTI

SWEPP Assay System Version 2.0 Software Design Description

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

August 1996

ph

 **Lockheed**
Idaho Technologies Company

**SWEPP Assay System Version 2.0
Software Design Description**

**L.V. East
E.S. Marwil**

Published August 1996

**Idaho National Engineering Laboratory
Lockheed Martin Idaho Technologies Company
Idaho Falls, Idaho 83415**

**Prepared for the
U.S. Department of Energy
Office of Environmental Restoration and Waste Management
Under DOE Idaho Operations Office
Contract DE-AC07-761D01570**

DISCLAIMER

**Portions of this document may be illegible
in electronic image products. Images are
produced from the best available original
document.**

DISCLAIMER

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.

Acknowledgments

The comments and suggestions provided by the RWMC/SWEPP Operations Personnel under the direction of G. K. Tedford during the development and testing of this software are gratefully acknowledged. Extensive testing and evaluation of the software during development of Version 2.0 was provided by G. W. Twedell.

Approvals

Approval signatures for this document are contained in DAR-EO-RS-2799 on file in the Document Control Office of the INEL Radioactive Waste Management Complex. *AK*

Contents

Acknowledgments.....	ii
Approvals.....	ii
1. INTRODUCTION	1
1.1 Purpose.....	2
1.2 Scope	2
1.3 Definitions, Acronyms, and Abbreviations	2
1.4 Overview.....	2
2. REFERENCES	3
3. NEEDED RESOURCES.....	4
3.1 Hardware	4
3.2 Software.....	5
4. OBJECT MODEL	6
5. OBJECT DESCRIPTION	16
5.1 about_dlg	16
5.2 acquire_dlg.....	17
5.3 active_dlg.....	18
5.4 active_signals	18
5.5 assay_system.....	19
5.6 calibration_dlg.....	20
5.7 camac.....	21
5.8 camac_217	21
5.9 camac_2323	21
5.10 camac_2551	22
5.11 camac_450	22
5.12 camac_8901	23
5.13 camac_crate	23
5.14 chamber	24
5.15 chamber_wnd	24
5.16 clock_generator	25
5.17 clock_port	26
5.18 coincidence_detector	26
5.19 container	26
5.20 container_dlg.....	27
5.21 container_list.....	28
5.22 controller.....	29
5.23 delay_dlg.....	29

5.24	detector	30
5.25	device.....	30
5.26	device list	31
5.27	dms_dlg.....	31
5.28	error_bar.....	32
5.29	gate	32
5.30	gate_generator.....	33
5.31	gate_port.....	33
5.32	hstreambuf	34
5.33	istream	34
5.34	link.....	34
5.35	list.....	35
5.36	login_dlg	35
5.37	main_wnd.....	36
5.38	matrix	37
5.39	matrix_dlg.....	38
5.40	matrix_list.....	39
5.41	matrix_parameter_dlg.....	39
5.42	message_dlg	40
5.43	nuclide	41
5.44	options_dlg.....	41
5.45	passive_dlg.....	42
5.46	passive_signals.....	43
5.47	port.....	44
5.48	port_2323.....	44
5.49	results_dlg.....	45
5.50	scaler	46
5.51	scaler_port	46
5.52	signals.....	47
5.53	timer	47
5.54	user.....	48
5.55	users_dlg.....	48
5.56	user_list	48
5.57	waste_dlg	49
6.	REQUIREMENTS TRACEABILITY	51
7.	DETAIL DESIGN.....	54
8.	FUTURE DEVELOPMENT DIRECTIONS	54
Appendix A	– Include Files.....	A-1
Appendix B	– Data Files.....	B-1

List of Figures

1. Assay system object model, level 1.	7
2. Assay system object model, level 1.1.	8
3. Assay system object model, level 1.2.	9
4. Assay system object model, level 1.3.	10
5. Assay system object model, level 1.4.	11
6. Assay system object model, level 1.4.1.....	12
7. Assay system object model, level 1.4.1.1.....	13
8. Assay system object model, level 1.4.1.2.....	14
9. Assay system object model, level 1.4.1.3.....	15
10. Assay system object model, level 1.4.2.....	15
11. About dialog window.....	16
12. Acquire dialog window.....	17
13. Active dialog (data) window	18
14. Calibration dialog window	20
15. Assay chamber window and child windows.....	25
16. Container dialog window	28
17. Delay dialog window	30
18. DMS dialog window.....	32
19. User login dialog window	36
20. Main window with one chamber as a child window	37
21. Matrix dialog window	38
22. Matrix parameter dialog window	40
23. Message dialog window.....	41
24. Processing options (assay times) dialog window.....	42
25. Passive dialog (data) window.....	43
26 Results dialog window.....	45
27. Waste dialog window	50

List of Tables

1. CAMAC crate module configuration.....	4
2. Chamber configuration files.....	6
3. Requirements traceability matrix.	51

SWEPP Assay System Software Design Description

1. INTRODUCTION

The Idaho National Engineering Laboratory (INEL) Stored Waste Examination Pilot Plant (SWEPP) operations staff use nondestructive analysis methods to characterize the radiological contents of contact-handled radioactive waste containers. Containers of waste from Rocky Flats Environmental Technology Site and other Department of Energy (DOE) sites are currently stored at SWEPP. Before these containers can be shipped to the Waste Isolation Pilot Plant (WIPP), SWEPP must verify compliance with storage, shipping, and disposal requirements. This program has been in operation since 1985 at the INEL Radioactive Waste Management Complex (RWMC).

One part of the SWEPP program measures neutron emissions from the containers and estimates the mass of plutonium and other transuranic (TRU) isotopes present. A Passive/Active Neutron (PAN) assay system developed at the Los Alamos National Laboratory [1]¹ is used to perform these measurements. A computer program named NEUT2 was originally used to perform the data acquisition and reduction functions for the neutron measurements. This program was originally developed at Los Alamos and extensively modified by a commercial vendor of PAN systems and by personnel at the INEL. NEUT2 uses the analysis methodology outlined in [1], but no formal documentation exists on the program itself. The SWEPP Assay System (SAS) computer program replaced the NEUT2 program in early 1994. The SAS software was developed using an "object model" approach and is documented in accordance with American National Standards Institute (ANSI) and Institute of Electrical and Electronic Engineers (IEEE) standards. The new program incorporates the basic analysis algorithms found in NEUT2. Additional functionality and improvements include a graphical user interface, the ability to change analysis parameters without program code modification, an "object model" design approach and other features for improved flexibility and maintainability.

This software design description (SDD) presents the SAS computer program Version 2.0 design and supersedes the original SAS software design description [2]. Companion documents include a software requirements specification [3], a software verification and validation plan [4], a software test plan and report [5], and a software configuration management plan [6]. Development of this technology is expected to continue following the Version 2.0 release.

Version 1.0 of the SAS software was developed by staff at the INEL Scientific Computing unit under the technical direction of the Nuclear and Radiation Physics unit. Version 2.0 is under development by staff from the INEL Nuclear Physics and Radiation Chemistry Department, with documentation and testing assistance provided by personnel from the Software and Electronics Department and the Engineering Analysis Department. Funding for this work has been provided through the INEL TRU Waste Programs Department.

¹ References cited in this document are listed in Section 2.

1.1 Purpose

This SDD covers SAS internal and external interfaces. The primary purpose of this design description is to provide a foundation for SAS development and verification and validation conformable to INEL/RWMC quality requirements [7].

The intended audiences for this specification are the developers and verification and validation analysts for the SAS software. In addition, this SDD provides a basis for understanding between the developers and the users of SAS.

This description follows the design description format and content guideline suggested by IEEE software engineering standards [8]. Methods employed are those commonly associated with Object Oriented Design, specifically the Rumbaugh et al. [9] implementation of object models and state transition diagrams. A computer program called MacAnalyst/MacDesigner [10] was employed as a Computer Aided Software Engineering (CASE) tool to provide automation of these methods during the Version 1.0 design.

1.2 Scope

The design description records the division of the software system into design entities. It describes the way the system has been structured and the purpose and function of each entity. For each entity, it provides a detailed description in terms of a Program Design Language (PDL). The design specifies the relationships between entities and the interfaces used.

1.3 Definitions, Acronyms, and Abbreviations

Definitions, acronyms, and abbreviations typical to an SDD and to the life cycle approach to software development in general are provided in Reference [11]. The following terms are specific to this document:

Design entity – An element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced.

Entity attribute – A named characteristic or property of a design entity. It provides a statement of fact about the entity.

MFC – Microsoft Foundation Class.

1.4 Document Overview

Section 2 lists the references for this document. Section 3 identifies the hardware and software resources required. Section 4 presents the object model for the SAS software, and section 5 gives the descriptions for each object. Section 6 provides the requirements traceability matrix. Section 7 presents detailed design information.

2. REFERENCES

- [1] J.T. Caldwell, et. al., *The Los Alamos Second-Generation System for Passive and Active Neutron Assays of Drum-Size Containers*, LA-10774-MS, Los Alamos National Laboratory, September 1986.
- [2] S.D. Matthews and E.S. Marwil, *SWEPP Assay System Design Description*, EGG-RAAM-10424, EG&G Idaho, Inc. September 1993.
- [3] S.D. Matthews, L.V. East, E.S. Marwil and J.J. Ferguson, *SWEPP Assay System Version 2.0 Software Requirements Specification*, INEL-96/0056, Lockheed Idaho Technologies Co., March 1996.
- [4] S.D. Matthews, *SWEPP Nondestructive Assay Software Verification and Validation Plan*, INEL-96/-0008, Lockheed Idaho Technologies Co., March 1996.
- [5] J.J. Ferguson and T.K. Overlin, *SWEPP Assay System Version 2.0 Software Test Plan and Report*, INEL-96/0058, Lockheed Idaho Technologies Co., April 1996.
- [6] J.J. Ferguson, K.D. Boyd and E.S. Marwil, *SWEPP Assay System Version 2.0 Software Configuration Management Plan*, INEL-96/0007, Lockheed Idaho Technologies Co., March 1996, (to be developed).
- [7] "RWMC Quality Assurance Project Plan (QAPjP)", INEL-94/0084, 1994.
- [8] ANSI/IEEE Std. 1016-1987, *IEEE Recommended Practice for Software Design Descriptions*, Institute of Electrical and Electronic Engineers, Inc.
- [9] James Rumbaugh et al., Object-Oriented Modeling and Design, Prentice-Hall, New York, 1991.
- [10] "MacAnalyst, Release 3.2," Excel Software, Marshalltown, Iowa, 1991.
- [11] ANSIEEE Std. 610.1a-10990, *IEEE Standard Glossary of Software Engineering Terminology*, Institute of Electrical and Electronic Engineers, Inc.
- [12] National Instruments Corp., Austin, Texas.
- [13] G.F. Knoll, Radiation Detection and Measurement, John Wiley and Sons, New York, NY, 1979.
- [14] H.W. Coleman and W.G. Steel, Experimentation and Uncertainty Analysis for Engineers, John Wiley and Sons, New York, NY, 1989.
- [15] K.D. Welker, B.C. Hong, G.V. Miller and B.J. Colborn, *Stored Waste Examination Pilot Plant Assay System Metrics Research Study*, INEL-95/0391, Lockheed Idaho Technologies Co., September 1995.
- [16] Microsoft Corporation, Class Library Reference for the Microsoft Foundation Class Library, Document No. DB35743-0193, 1993.

3. NEEDED RESOURCES

3.1 Hardware

SAS must execute in the current hardware environment. The current computer is an IBM compatible PC with an INTEL 486 CPU. A mouse (or other MS-Windows supported pointing device), VGA color monitor, and Hewlett Packard LaserJet III compatible printer are also required. At least eight megabytes of MS-Windows accessible memory, a floppy disk, and a hard disk with at least ten megabytes of free space are required. In addition, a free serial port is required for SAS communication with the SWEPP Data Management System (DMS).

The SAS program accesses signal data from a Computer Automated Measurement and Control (CAMAC) instrument crate. The required crate, modules, and controller shall be those currently in use at SWEPP. The PC requires an AT-GPIB controller card from National Instruments Corporation [12] to manage the interface to the CAMAC crate through the crate controller.

The current configuration of the CAMAC crate by station number and module number are given in Table 1.

Table 1. CAMAC crate module configuration.

Station	Module type	Module name
01-02	2323	LeCroy 2323A Gate Generator
03-04	2323	LeCroy 2323A Gate Generator
05	2551	LeCroy 2551 Scaler
06	2551	LeCroy 2551 Scaler
07-08	2323	LeCroy 2323A Gate Generator
09-10	2323	LeCroy 2323A Gate Generator
11	2551	LeCroy 2551 Scaler
12	2551	LeCroy 2551 Scaler
13	450	DSP 450 Gated Quad Scaler
14-15	2323	LeCroy 2323A Gate Generator
16-17	2323	LeCroy 2323A Gate Generator
18	2551	LeCroy 2551 Scaler
19	2551	LeCroy 2551 Scaler
20	450	DSP 450 Gated Quad Scaler
21	217	Jorway 217 Gated Clock Generator
22	202	Jorway 202 Dataway Display
23	2551	LeCroy 2551 Scaler
24-25	8901	LeCroy 8901A GPIB Interface

3.2 Software

3.2.1 Language and Tools

Version 2.0 of the SAS software was developed using the Microsoft Visual C++ V1.51 development environment for Microsoft Windows 3.1. The device driver for the AT-GPIB controller (GPIB.DLL) and the interface library (GPIB.LIB) are licensed from National Instruments [12].

3.2.2 Necessary Files for Execution

Data Files

These data files are described briefly here. Each file is listed in Appendix B.

System configuration files

AS.INI – the system initialization file. The program's root directory; user and error logging files, printer, serial port configuration, files containing the user list, the container list, and the matrix list, and the chamber initialization files are defined in this file.

ASSAYSYS.INI² – program initialization file. This file resides in the default Windows directory and is used to save window size, placement and color information. This file is automatically created the first time the program is executed and updated each time the program exits. It is not listed in the appendix.

DAC.INI – the configuration file for the drum assay chamber. This file identifies the name of the assay chamber, the defaults for printing the summary report, applying passive background correction and automatically saving data, customization parameters for background integrity checks and other functions, the directories for this assay chamber, and the files for this assay chamber.

CAC.INI – the configuration file for the crate assay chamber. This file identifies the name of the assay chamber, the defaults for printing the summary report, applying passive background correction and automatically saving data, customization parameters for background integrity checks and other functions, the directories for this assay chamber, and the files for this assay chamber.

USER.LST – the list of users who have access to the system. This file is not listed in the appendix.

CONTAIN.LST – the list of container types and volumes that the assay system can process.

MATRIX.LST – the list of matrix types and attributes that the assay system can process. The attributes are those which can be modified by a user with Physicist access level.

NUCLIDE.DAT – the data file that defines the physical properties of the plutonium, uranium and americium nuclides that are used by the SAS.

DEFAULT.GAM – the default distribution of the plutonium isotopes. This file is used if no other gamma data is available for the container being processed.

² The name of this file may change; it will be the same as the executable file name with the extension .INI. Unlike AS.INI, DAC.INI, etc., this is a "true" MS Windows initialization file.

Chamber configuration files

The chamber configuration files are identified in Table 2. The options data file defines the default processing options for background count time, active pulses and passive count times for both verification and production runs. The calibration file defines the chamber attributes (detector efficiencies, zero matrix calibration factors, and the mass fractions of plutonium used for calibration). The correlation data file defines the coefficients for the correlations taken from the NEUT2 program. The chi-square file defines the parameters for the Chi-square tests. Two CAMAC configuration files describe the CAMAC crate configuration, one for the active data acquisition and one for the passive data acquisition. Finally, the files for the current passive background data and current verification data, while not strictly part of the configuration, are listed for completeness.

Table 2. Chamber configuration files.

Description	Drum Assay Chamber	Crate Assay Chamber
Processing options	OPTIONS.DAC	OPTIONS.CAC
Chamber calibration	CALIB.DAC	CALIB.CAC
Correlation coefficients	CORREL.DAC	CORREL.CAC
Chi-square parameters	CHISQ.DAC	CHISQ.CAC
Active CAMAC configuration	ACTCRATE.DAC	ACTCRATE.CAC
Passive CAMAC crate configuration	PASCRATE.DAC	PASCRATE.CAC
Current passive background data	BACKGRND	BACKGRND
Current verification data	000496	000496

4. OBJECT MODEL

This section provides the object model, using Rumbaugh's notation [9]. The object model, shown on the following pages, provides a graphical representation of the dependencies and associations.

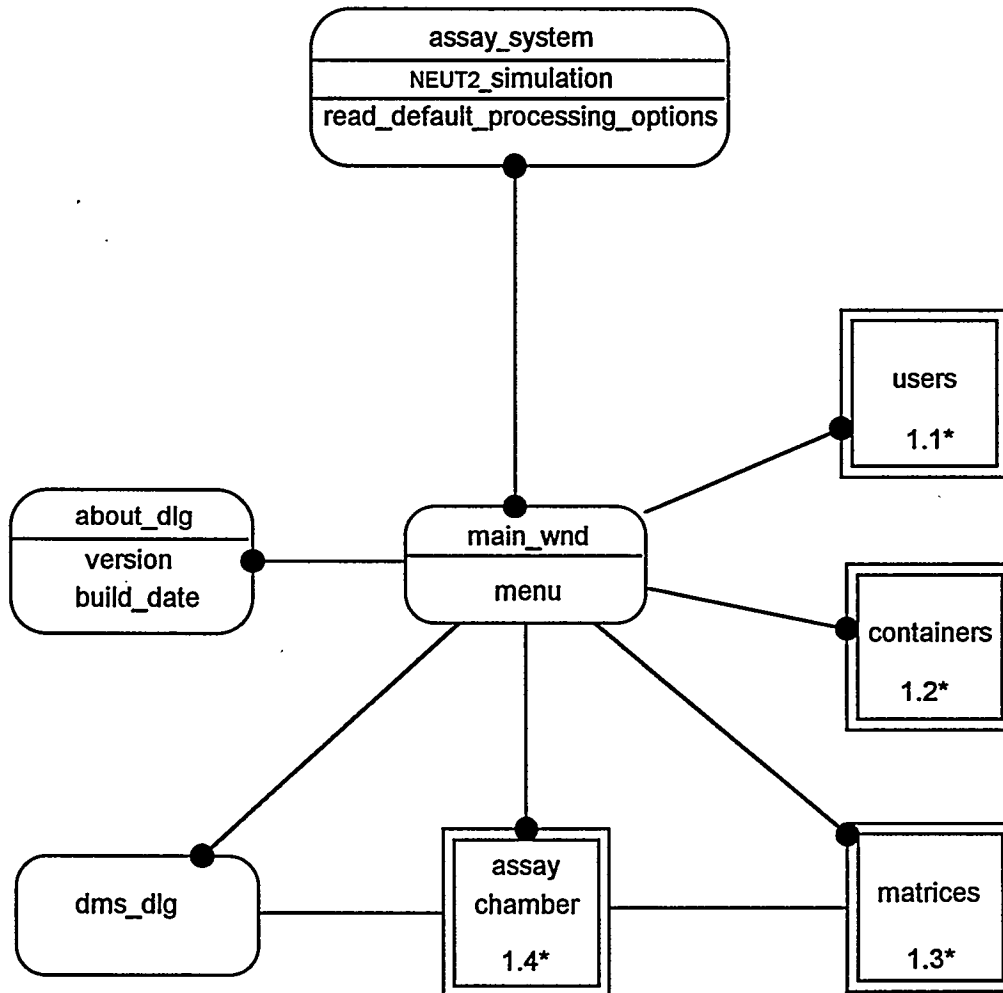


Figure 1. Assay system object model, level 1.

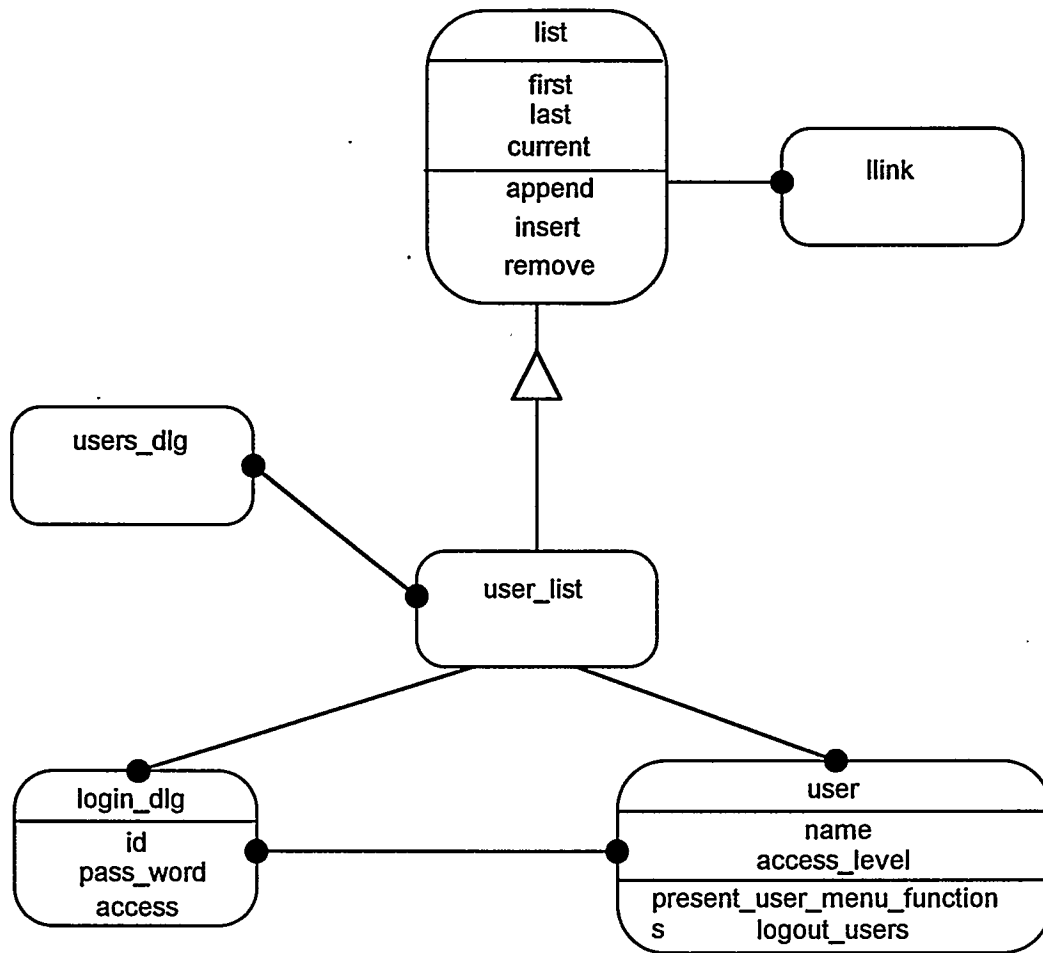


Figure 2. Assay system object model, level 1.1.

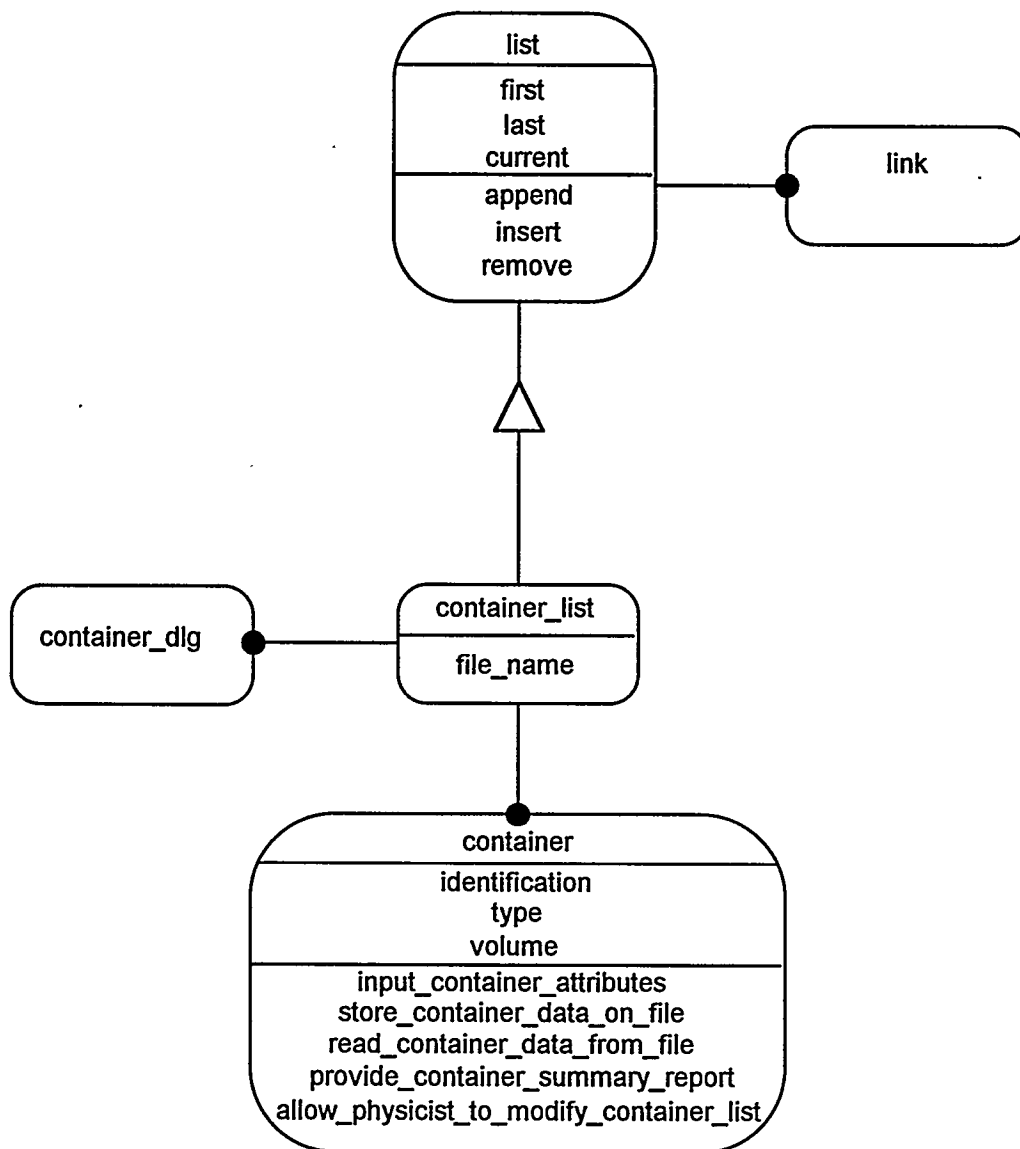


Figure 3. Assay system object model, level 1.2.

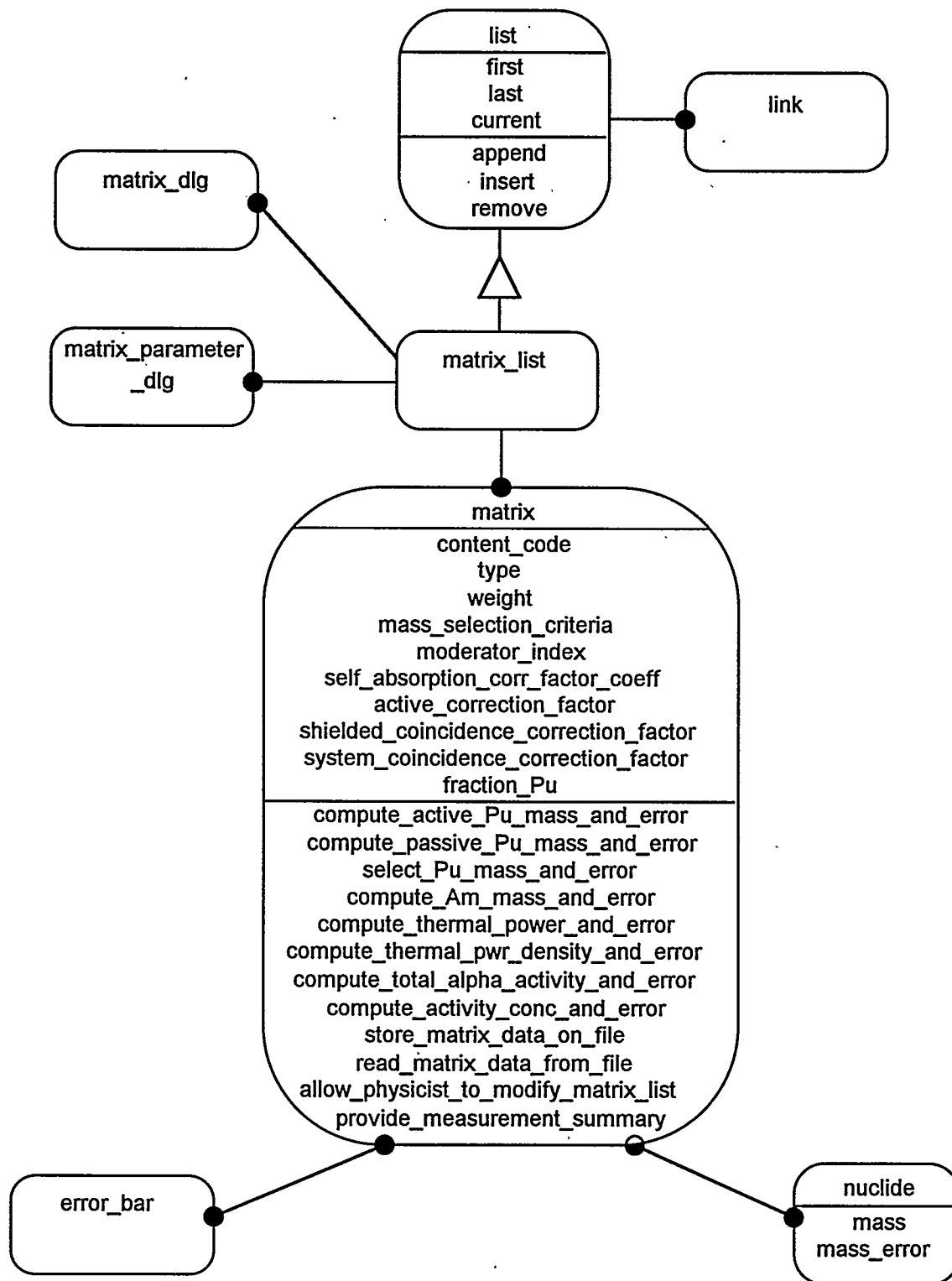


Figure 4. Assay system object model, level 1.3.

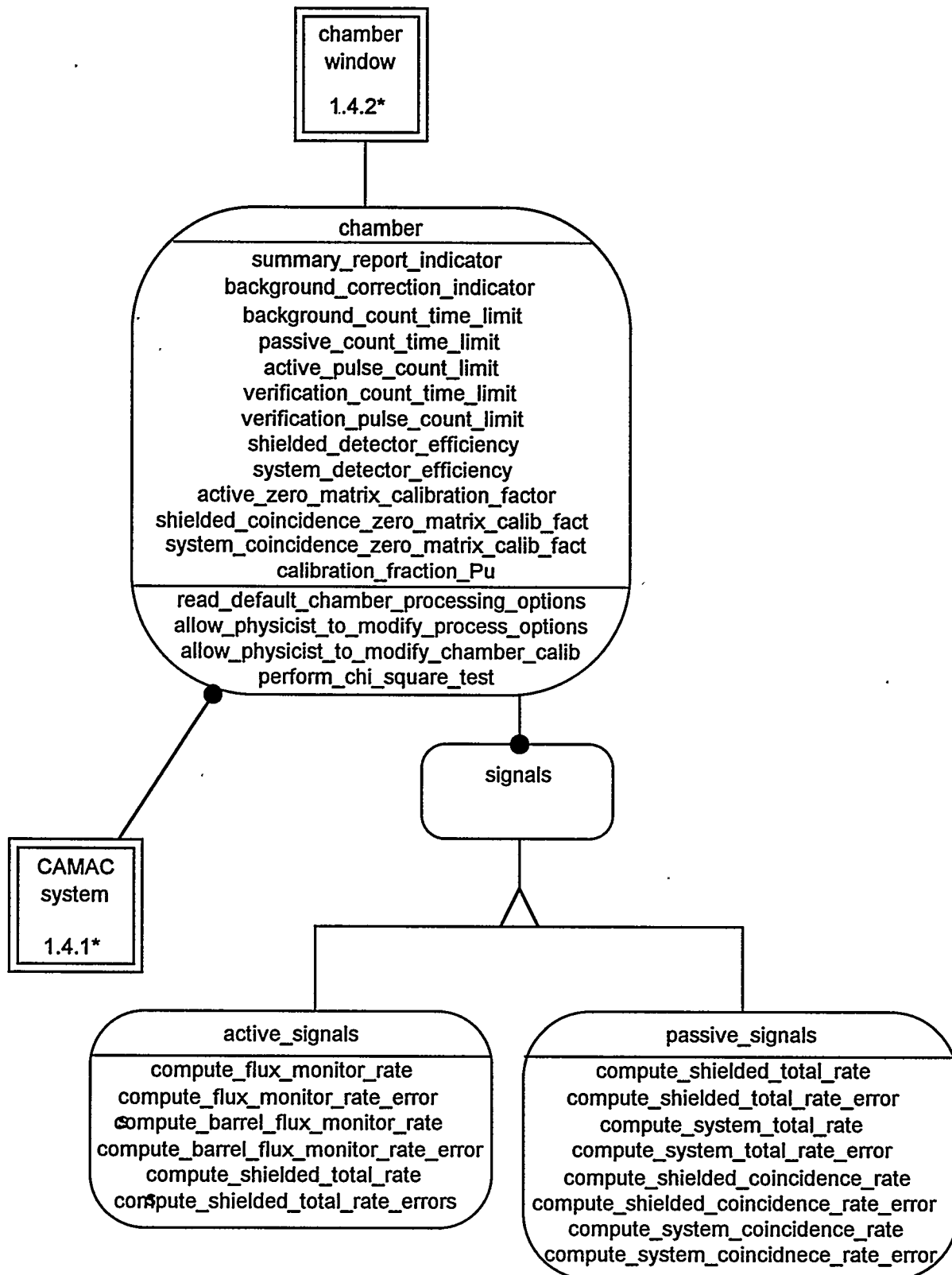


Figure 5. Assay system object model, level 1.4.

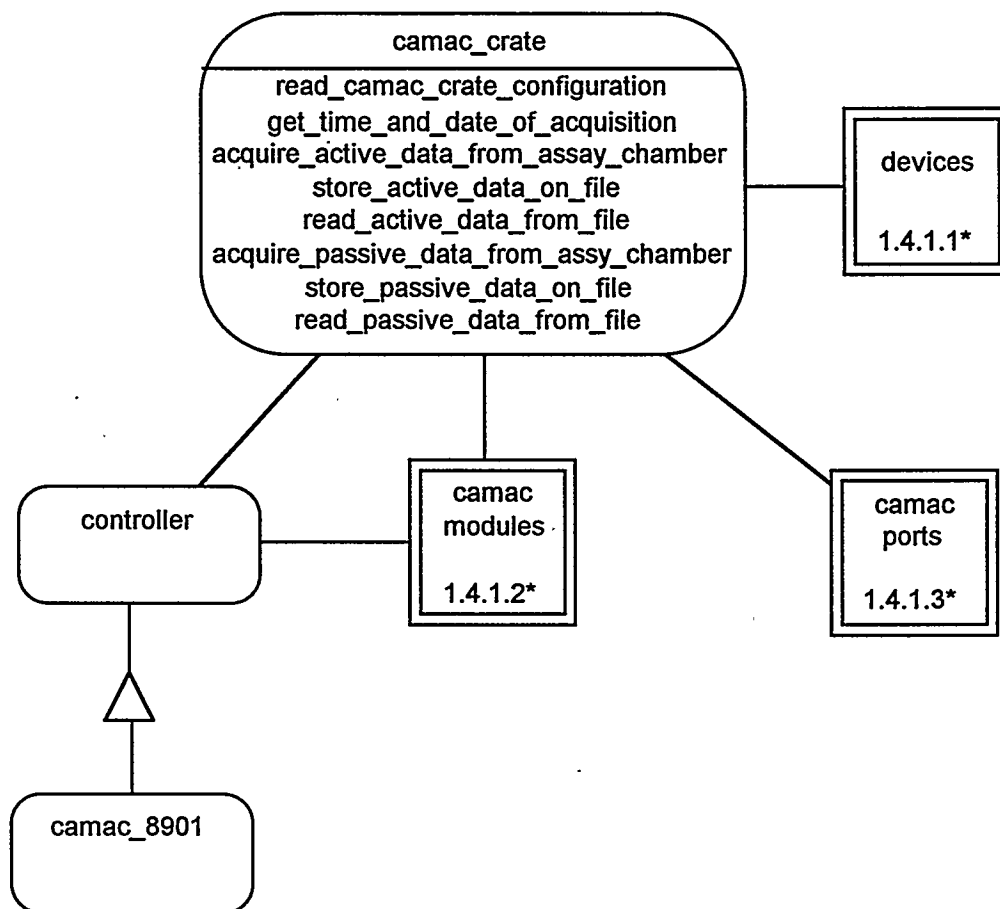


Figure 6. Assay system object model, level 1.4.1.

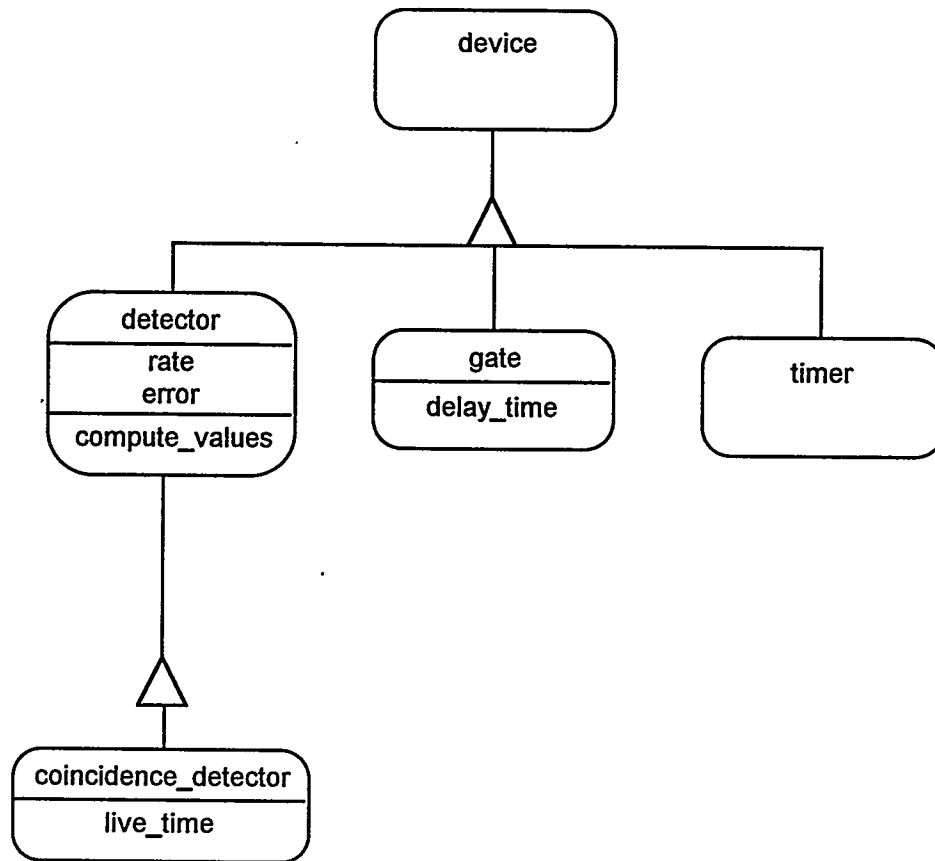


Figure 7. Assay system object model, level 1.4.1.1.

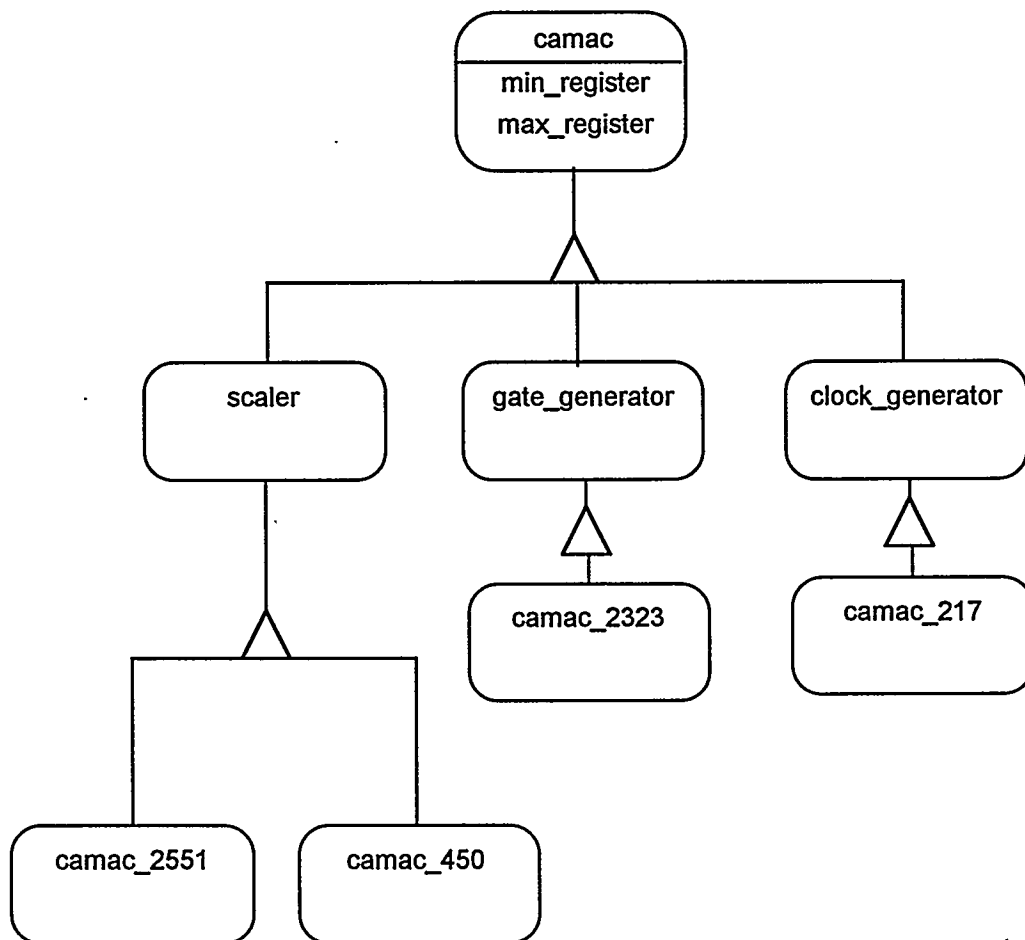


Figure 8. Assay system object model, level 1.4.1.2.

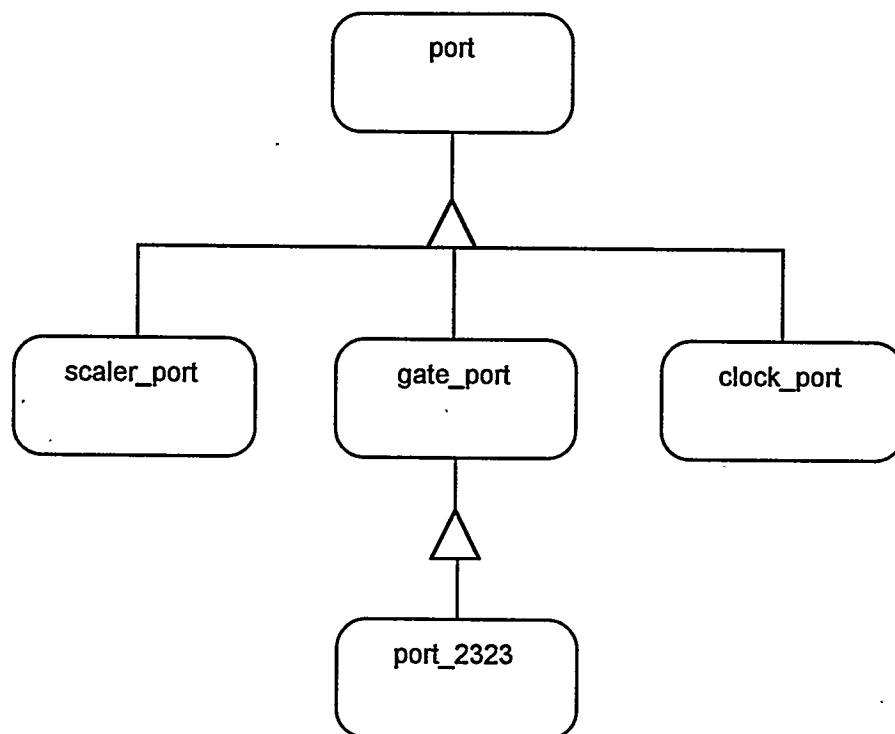


Figure 9. Assay system object model, level 1.4.1.3.

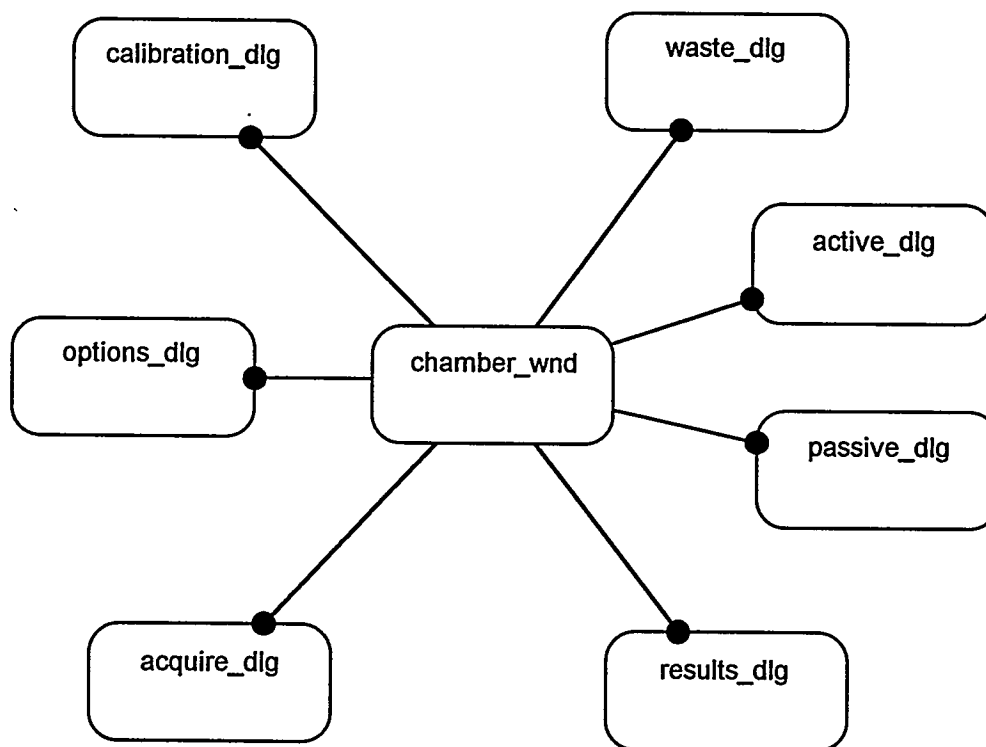


Figure 10. Assay system object model, level 1.4.2.

5. OBJECT DESCRIPTION

This section provides a description, the decomposition, dependencies, and interfaces for each object. This information is most useful in conjunction with the object model in the previous section. Within this section objects are arranged alphabetically. A reference to a figure, where applicable, allows the reader to easily find the object in the object model.

5.1 about_dlg

5.1.1 Description and Purpose

The about dialog shows the version and the build date of the application (see Figure 1). The dialog is displayed when the user selects the menu item **About...** It is dismissed by clicking on the **Acknowledge** button (see Figure 11).

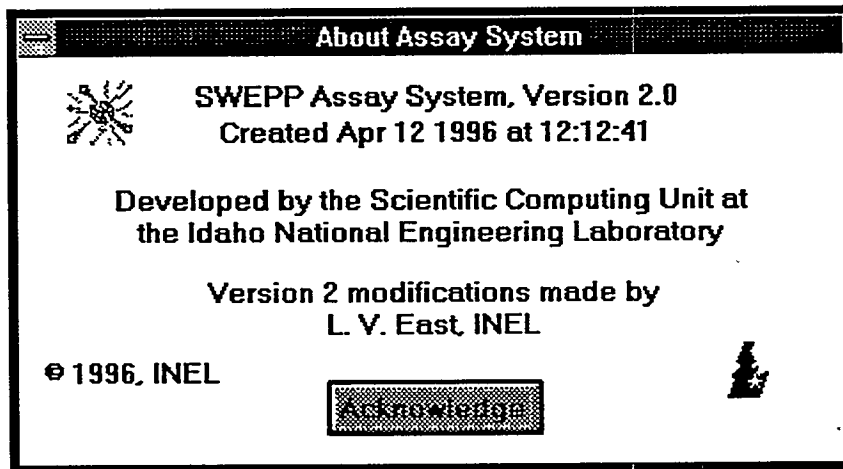


Figure 11. About dialog window.

5.1.2 Decomposition

This class is derived from CDialog, a class in the Microsoft Foundation Class (MFC) library. There are no subclasses.

5.1.3 Dependencies

It is created, displayed, and destroyed in a single function in the main window in response to the menu selection. The dialog receives the program identification and build date through the argument list of its creator. The about dialog may be displayed while a chamber window is active.

5.1.4 Interfaces

Files - none.

Windows - the about dialog box.

5.2 acquire_dlg

5.2.1 Description and Purpose

The acquire dialog shows the status of a data acquisition (see Figure 10). The dialog title identifies the type of acquisition (Active, Passive Gross, Passive Background, or Chi-square). The required acquisition limit (pulse counts for active or time for passive) is displayed. The current acquisition count is displayed along with a graphic status bar that shows the percent of the acquisition completed. In addition, a message line provides additional status information or requests operator action, such as "Ensure that the assay chamber door is closed. Start the neutron pulse generator when ready." An operator may cancel the acquisition by clicking on the **Cancel** button (see Figure 12).

5.2.2 Decomposition

This class is derived from CDialog, a class in the MFC library. There are no subclasses.

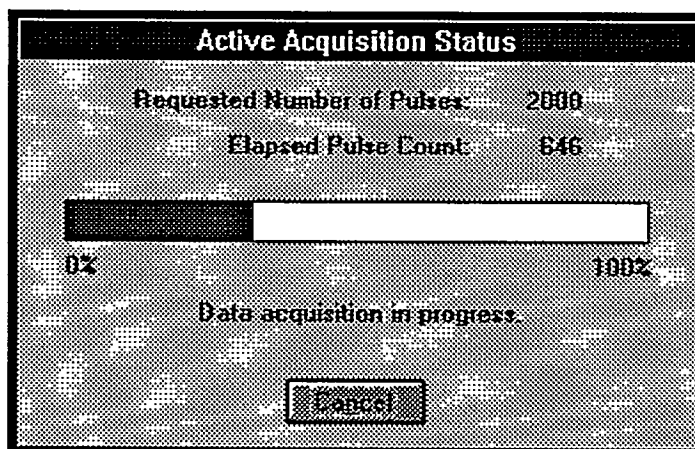


Figure 12. Acquire dialog window.

5.2.3 Dependencies

This object is created whenever one of the acquisition functions is invoked in the chamber window. The chamber window is responsible for ensuring that a CAMAC crate is ready to begin data acquisition. The acquire dialog creator function receives a pointer to a chamber through the argument list. The chamber pointer provides the dialog with the interface to get the name of the crate performing the acquisition and to get the limit value that determines when the acquisition is complete. The acquire dialog requests the chamber to start the acquisition, check the acquisition progress, and stop the acquisition.

5.2.4 Interfaces

Files - none.

Windows - the acquire dialog box.

5.3 active_dlg

5.3.1 Description

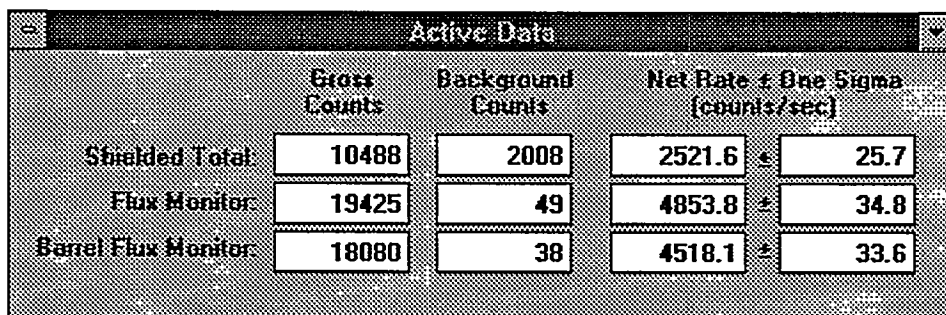
The active dialog shows the counts and rates from an active data acquisition (see Figure 10). Both the gross and background counts are displayed for the shielded total, flux monitor, and the barrel flux monitor signals (see Figure 13). Rates for those same signals are displayed with an uncertainty ("error") of one standard deviation based on counting statistics [13].

5.3.2 Decomposition

This class is derived from CDialog, a class in MFC library. There are no subclasses.

5.3.3 Dependencies

This object is created by a chamber window when active data are acquired or read from a file. The active dialog creator function receives a pointer to an active signals object through the argument list. The active signals pointer provides the dialog with the interface to get the counts and rates that the dialog displays.



	Gross Counts	Background Counts	Net Rate $\pm$ One Sigma (counts/sec)	
Shielded Total:	10488	2008	2521.6	$\pm$ 25.7
Flux Monitor:	19425	49	4853.8	$\pm$ 34.8
Barrel Flux Monitor:	18080	38	4518.1	$\pm$ 33.6

Figure 13. Active dialog (data) window.

5.3.4 Interfaces

Files - none.

Windows - the active dialog box.

5.4 active_signals

5.4.1 Description

The active signals class computes the net rates for the shielded total, flux monitor, and barrel flux monitor signals. (See Figure 5.)

5.4.2 Decomposition

This class is derived from the signals class (see below). There are no subclasses.

5.4.3 Dependencies

An active signals object is created by a chamber. A pointer to the associated CAMAC crate that maintains the active data is passed as an argument to the active signals creator function. The CAMAC crate provides the interface to get the gross counts, background counts, and count rates for the active signals.

The gross count and rate are associated with the "early gate," and the background counts and rate are associated with the "late gate." These gates and the signal names are defined in the CAMAC crate configuration file. A change in the naming convention of the gates or signals will require a change in the active signals.

The active signals object is used by the active dialog to display the active data, and it is used by the matrix object in the data reduction.

5.4.4 Interfaces

Files - none.

Windows - none.

5.5 assay_system

5.5.1 Description

The assay system object provides the interface control for the SAS program. (See Figure 1.) This class is superordinate and addresses all the active and passive data assay requirements. Microsoft Windows runs the application as an instance of this object.

5.5.2 Decomposition

This class is derived from CWinApp in the MFC library. There are no subclasses.

This procedure creates the global assay system object. Once it is created, Microsoft Windows manages it.

5.5.3 Dependencies

The assay system defines the global character string for the program identification. This contains the system name and version number used for reports, data files, and the about dialog.

Microsoft Windows manages the windows interface when the application is created. The application is created when the user double clicks on the SWEPP Assay System icon. The assay system creates the main window and sends it the message to display the login dialog. Control is turned over to Microsoft Windows.

5.5.4 Interfaces

Files: none.

Windows: none.

5.6 calibration_dlg

5.6.1 Description

The calibration dialog provides the user interface to change the chamber calibration parameters (see Figures 10) These include the detector efficiencies, the zero matrix calibration factors (active and passive shielded coincidence and passive system coincidence) and the distribution of plutonium to which the chamber is calibrated (see Figure 14).

Drum Assay Chamber Calibration			
Detector efficiency (counts / source neutron)			
Shielded:	0.02900	±	0.00000
System:	0.12650	±	0.00000
Zero matrix calibration factors (grams Pu / net counts / sec)			
Active:	4.35000	±	0.00000
Shielded coincidence:	28.10000	±	0.00000
System coincidence:	1.28000	±	0.00000
Plutonium mass fractions			
Pu 238:	0.00012	±	0.00000
Pu 239:	0.93826	±	0.00000
Pu 240:	0.05820	±	0.00000
Pu 241:	0.00340	±	0.00000
Pu 242:	0.00024	±	0.00000
OK Cancel			

Figure 14. Calibration dialog window.

5.6.2 Decomposition

This class is derived from CDialog, a class in MFC library. There are no subclasses.

5.6.3 Dependencies

The calibration dialog is created by the chamber window in response to the menu selection **Calibration...** on the physicist's menu. A pointer to the chamber is passed as an argument to the creator function. The chamber provides the interface to the calibration parameters.

5.6.4 Interfaces

Files: none.

Windows: the calibration dialog.

5.7 camac

5.7.1 Description

The camac class models a generic CAMAC module. (See Figure 8.) A module has a station number in the CAMAC crate, a type identifier, a model number (e.g., 217, 2323, 2551, 450, or 8901), minimum and maximum registers, and an array of associated ports (registers or addresses).

5.7.2 Decomposition

There is no superclass. Subclasses are clock generator, gate generator, and scaler.

5.7.3 Dependencies

A camac object is created by the CAMAC crate if the model of CAMAC module is not already in the defined set. Any camac object can interrogate its array of ports to locate a port with a specified name.

5.7.4 Interfaces

Files: none.

Windows: none.

5.8 camac_217

5.8.1 Description

The camac_217 class encapsulates the behavior of the Jorway 217 Gated Clock Generator. (See Figure 8.) This class has an array of clock ports.

5.8.2 Decomposition

This class is derived from the clock generator class. There are no subclasses.

5.8.3 Dependencies

A camac_217 object is created by the CAMAC crate. The camac_217 is associated with the crate controller that provides the interface to control the hardware module.

5.8.4 Interfaces

Files: none.

Windows: none.

5.9 camac_2323

5.9.1 Description

The camac_2323 class encapsulates the behavior of the LeCroy 2323 Dual Gate and Delay Generator. (See Figure 8.)

5.9.2 Decomposition

This class is derived from the gate generator class. There are no subclasses.

5.9.3 Dependencies

A camac_2323 object is created by the CAMAC crate. The camac_2323 is associated with the crate controller that provides the interface to control the hardware board. This class has an array of gate ports specific to the 2323 module.

5.9.4 Interfaces

Files: none.

Windows: none.

5.10 camac_2551

5.10.1 Description

The camac_2551 class encapsulates the behavior of the LeCroy 2551 12 channel 100 MHz Scaler. (See Figure 8.)

5.10.2 Decomposition

This class is derived from the scaler class. There are no subclasses.

5.10.3 Dependencies

A camac_2551 object is created by the CAMAC crate. The camac_2551 is associated with the crate controller that provides the interface to control the hardware module. This class has an array of scaler ports.

5.10.4 Interfaces

Files: none.

Windows: none.

5.11 camac_450

5.11.1 Description

The camac_450 class encapsulates the behavior of the DSP 251 QS-450, 100 MHz Quad Scaler. (See Figure 8.)

5.11.2 Decomposition

This class is derived from the scaler class. There are no subclasses.

5.11.3 Dependencies

A `camac_450` object is created by the CAMAC crate. The `camac_450` is associated with the crate controller that provides the interface to control the hardware module. This class has an array of scaler ports.

5.11.4 Interfaces

Files: none.

Windows: none.

5.12 `camac_8901`

5.12.1 Description

The `camac_8901` encapsulates the behavior of the LeCroy 8901A GPIB interface. (See Figure 6.)

5.12.2 Decomposition

This class is derived from the controller class. There are no subclasses.

5.12.3 Dependencies

Each CAMAC module in a CAMAC crate is associated with a crate controller, such as the 8901.

5.12.4 Interfaces

Files: none.

Windows: none.

5.13 `camac_crate`

5.13.1 Description

The `camac_crate` object encapsulates the behavior of the CAMAC crate system. (See Figure 6.) The `camac_crate` object manages the configuration of its CAMAC modules, ports, and devices. It initializes the active and passive assays and sets up and tests the CAMAC crate module and channel configuration. `Camac_crate` also reads and clears the scalars.

5.13.2 Decomposition

There is no superclass. There are no subclasses.

5.13.3 Dependencies

The chamber object creates `camac_crate` objects for the active data, the passive gross data, and the passive background data. The appropriate `camac_crate` object performs the initialization, starts an acquisition, checks the acquisition progress, and stops the acquisition. In addition, the `camac_crate` reads its configuration from a file and writes the configuration and acquired data to a file.

5.13.4 Interfaces

Files: ACTCRATE.DAC, PASCRATE.DAC, ACTCRATE.CAC, PASCRATE.CAC, and data files for acquisition data.

Windows: none.

5.14 chamber

5.14.1 Description

The chamber class models an assay chamber. (See Figure 5.) Physically, the two assay chambers are the drum assay chamber and the crate assay chamber. Logical chambers may be used for calibration and test purposes.

5.14.2 Decomposition

There is no superclass. There are no subclasses.

5.14.3 Dependencies

A chamber object is associated with its corresponding chamber window. A chamber has public pointers to active signals and passive signals. A chamber also is associated with the CAMAC crate. Physically, there is only one CAMAC crate modeled logically as three CAMAC crate objects, one for active data, one for passive data, and one for passive background. The chamber provides the interface for the chamber window to start data acquisition, check acquisition progress, and stop data acquisition for the each type of data.

The chamber allows the calibration dialog friend class access to its attributes. The calibration dialog provides the user interface to change the calibration data (detector efficiencies, zero matrix calibration factors, and plutonium mass distribution to which the chamber is calibrated).

5.14.4 Interfaces

Files: CALIB.DAC and CALIB.CAC.

Windows: none.

5.15 chamber_wnd

5.15.1 Description

The chamber window class provides the user interface to the assay chamber operations and the file interface to the operating system (see Figure 10). Windows that display assay data and results are child windows of the chamber window (see Figure 15).

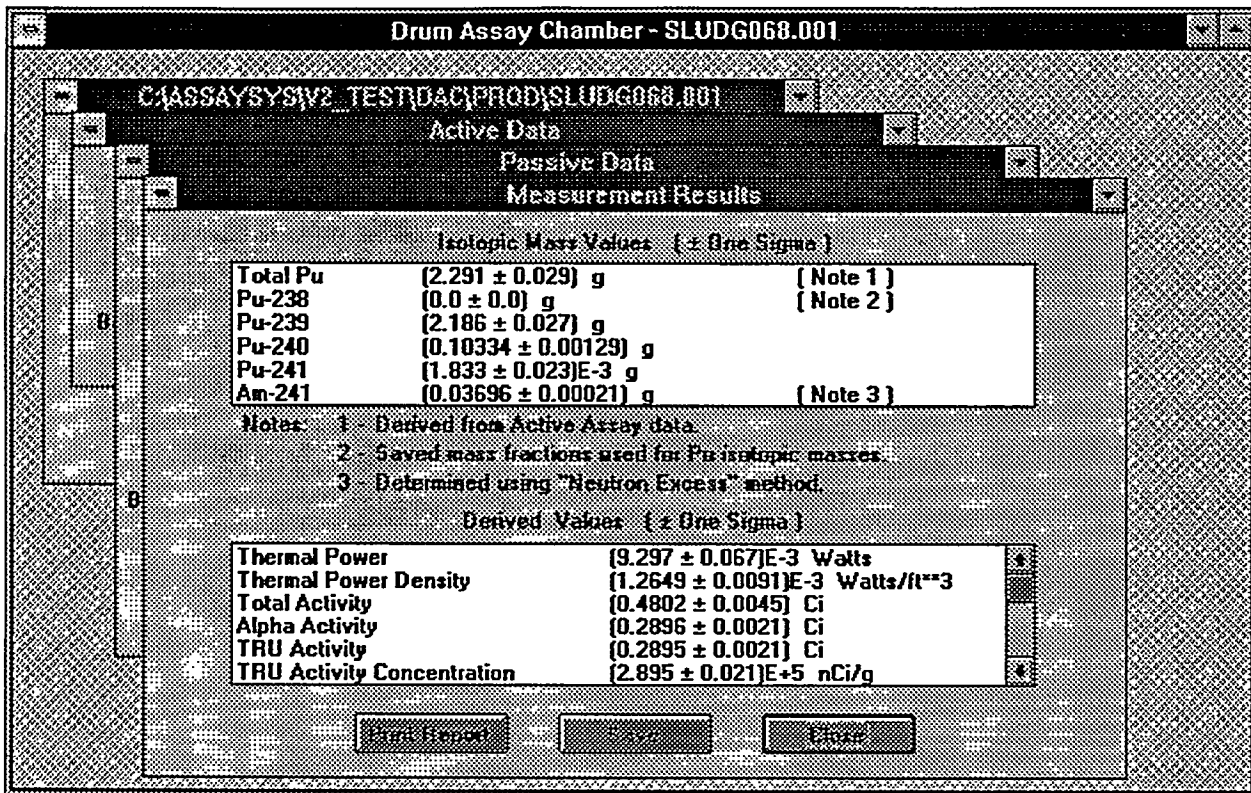


Figure 15. Assay chamber window and child windows.

5.15.2 Decomposition

This class is derived from CMDIChildWnd, a class in the MFC library.

5.15.3 Dependencies

The chamber window is created by the main window. Each chamber window has its own configuration file. The chamber window is associated with a chamber. The process options dialog provides the interface to set the pulse count limits and the count time limits for the active and passive data acquisitions respectively.

The chamber window is also associated with the dms dialog, when the dms interface is active.

5.15.4 Interfaces

Files: DAC.INI, OPTIONS.DAC, CHISQ.DAC, CAC.INI, OPTIONS.CAC, and CHISQ.CAC.

Windows: the chamber window.

5.16 clock_generator

5.16.1 Description

The clock_generator object models a generic CAMAC clock generator board. (See Figure 8.)

5.16.2 Decomposition

This class is derived from the camac class. It has camac_217 as a subclass.

5.16.3 Dependencies

The clock generator is associated with an array of clock ports. This provides the interface to get and set the clock rate for individual ports.

5.16.4 Interfaces

Files: none.

Windows: none.

5.17 clock_port

5.17.1 Description

The clock_port class models a port (register or address) on a CAMAC clock generator. (See Figure 9.) A clock port generates clock pulses at a prescribed rate.

5.17.2 Decomposition

This class is derived from the port class. There are no subclasses.

5.17.3 Dependencies

The clock port is created by a camac_217 object, which also sets the clock rates for designated clock ports. The clock rate is retrieved by a timer object.

5.17.4 Interfaces

Files: none.

Windows: none.

5.18 coincidence_detector

5.18.1 Description

The coincidence_detector class computes the coincidence rate, correcting the rate for accidental counts. (See Figure 7.)

5.18.2 Decomposition

This class is derived from the detector class. There are no subclasses.

5.18.3 Dependencies

A coincidence_detector is associated with a scaler_port, a detector that has the same name as the coincidence_detector, a gate, and a timer. It is created by the camac_crate object when specific devices are

associated with the scaler ports. The gate and the timer provide times that are used together with the count from the scaler port to compute a rate. Access to the rate is provided through the `camac_crate`.

5.18.4 Interfaces

Files: none.

Windows: none.

5.19 container

5.19.1 Description

The container object models the behavior of a generic container process at the assay station. (See Figure 3.) A container has an identification, such as a Rocky Flats ID, a type description, such as 55-gallon drum, and a volume in liters.

5.19.2 Decomposition

There is no superclass. There are no subclasses.

5.19.3 Dependencies

A container is associated with a matrix. The container provides the interface to get the volume used in data reduction. The container produces a report when requested.

5.19.4 Interfaces

Files: none.

Windows: none.

Reports: summary report.

5.20 container_dlg

5.20.1 Description

The container dialog class provides the user interface to modify the list of container types available for processing. (See Figures 3 and 16.)

5.20.2 Decomposition

This class is derived from `CDialog`, a class in the MFC library.

5.20.3 Dependencies

This class is created by main window when the menu item **Container List...** is selected. The container dialog is associated with the container list.

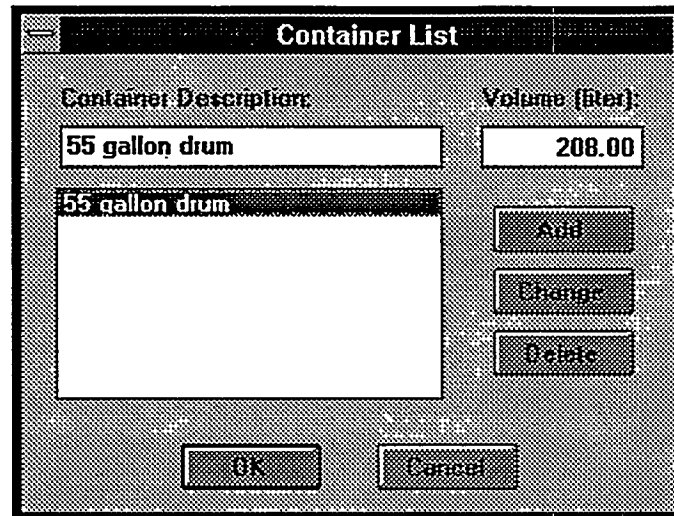


Figure 16. Container dialog window.

5.20.4 Interfaces

Files: none.

Windows: the container dialog.

5.21 container_list

5.21.1 Description

The container_list class models the list of containers used at the assay station. (See Figure 3.)

5.21.2 Decomposition

This class is derived from the list class. There are no subclasses.

5.21.3 Dependencies

The container list is created by the main window. It is used in the container window to select a waste container type for processing. It is also used in the container dialog to modify the list of available containers.

5.21.4 Interfaces

Files: none.

Windows: none.

5.22 controller

5.22.1 Description

The controller object encapsulates all the functionality for the CAMAC crate controller, regardless of the actual controller implementation. (See Figure 6.) This functionality includes transmitting an instruction to a controller and getting a response from a controller.

5.22.2 Decomposition

There is no superclass. There is one subclass, the `camac_8901`.

5.22.3 Dependencies

A controller is associated with each instance of a CAMAC module.

5.22.4 Interfaces

Files: none.

Windows: none.

5.23 delay_dlg

5.23.1 Description

A `delay_dlg` object is used to provide a delay between active and passive data acquisitions and indicate the time remaining to the user. The `delay_dlg` class was added in Version 2.0.

5.23.2 Decomposition

This class is derived from `CDialog`, a class in the MFC library. There are no subclasses.

5.23.3 Dependencies

A `delay_dlg` is created by a chamber object whenever passive data collection follows an active data collection and the delay time specified in the chamber configuration file is greater than zero.

5.23.4 Interfaces

Files: none.

Windows: Delay dialog window (Figure 17).

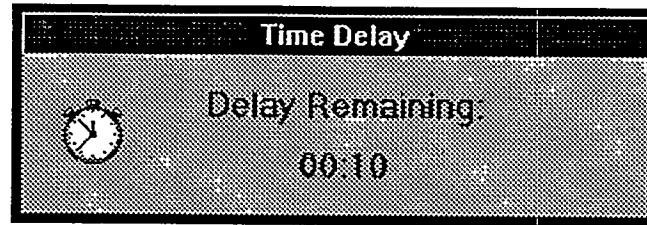


Figure 17. Delay dialog window.

5.24 detector

5.24.1 Description

The detector object models a neutron detector or group of detectors in the assay chamber. (See Figure 7.)

5.24.2 Decomposition

This object is derived from device. There is one subclasses, coincidence_detector.

5.24.3 Dependencies

A detector is associated with either a timer or a gate or both. The timer or gate provide an elapsed time, along with the counts from the associated scaler port, that the detector object uses to compute its rate.

5.24.4 Interfaces

Files: none.

Windows: none.

5.25 device

5.25.1 Description

The device object models a physical or logical device. (See Figure 7.) Derived classes, such as gate, timer, detector, or coincidence_detector, model behavior specific for each device. The device is associated with a scaler port.

5.25.2 Decomposition

This is no superclass. Subclasses are detector, gate, and timer.

5.25.3 Dependencies

The device is associated with a scaler port through the device list.

5.25.4 Interfaces

Files: none.

Windows: none.

5.26 device_list

5.26.1 Description

The device_list class models a list of logical or physical devices.

5.26.2 Decomposition

This class is derived from the list class. There are no subclasses.

5.26.3 Dependencies

One or more devices are associated with a scaler port through the device list. The scaler port may interrogate the list through the get_first, get_next, and find functions of the device list.

5.26.4 Interfaces

Files: none.

Windows: none.

5.27 dms_dlg

5.27.1 Description

The dms_dlg (dialog) class defines the interface with the DMS "remote" computer. (See Figure 1.) It opens a communications port, receives a request from the DMS, sends results to the DMS, and acknowledges that the DMS has received data. The DMS dialog window is shown in Figure 17.

5.27.2 Decomposition

This class is derived from CDialog, a class in the MFC library. There are no subclasses.

5.27.3 Dependencies

A single instance of dms_dlg is created by the main window, which it then associates with the chamber windows. The dms_dlg object passes requests for assays or recalculations to the main window. The main window determines the appropriate assay chamber to perform the request and sends the request to the chamber window.

The chamber window then asks the dms_dlg object to provide the details of the request, e.g., the container identification and the file identification. When processing is complete, the chamber window sends the dms_dlg object a message requesting that the assay results be forward to the DMS computer.

5.27.4 Interfaces

Files: none.

Windows: the dms dialog.

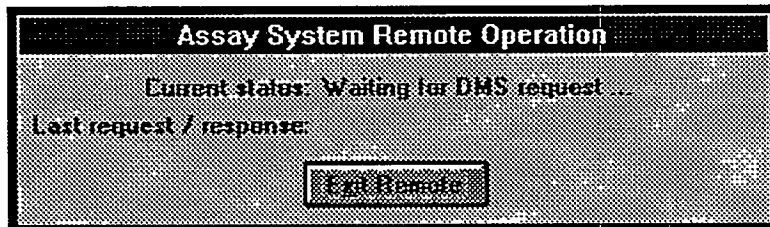


Figure 18. DMS dialog window.

5.28 error_bar

5.28.1 Description

The error_bar object class models an “error bar”, that is, a value plus or minus one standard deviation uncertainty. (See Figure 4.) Overloaded operators for addition, subtraction, multiplication and division for error_bar and error_bar in conjunction with float variables are defined to propagate the error through the computations. Overload operators for comparison operations were added in Version 2; these operators take into account the uncertainties associated with error_bar values at the 95 percent confidence level when performing comparisons between error_bar values; see Reference [14], Section 2-3.

5.28.2 Decomposition

There is no superclass. There are no subclasses.

5.28.3 Dependencies

This class is used by the matrix class in performing the data reduction. The chamber calibration parameters, mass fractions and mass ratios are also defined as error bars.

5.28.4 Interfaces

Files: none.

Windows: none.

5.29 gate

5.29.1 Description

The gate class models the functionality of a CAMAC gate module, which is a logical device. (See Figure 7.) A gate defines a window in time wherein certain events are counted.

5.29.2 Decomposition

This class is derived from the device class.

5.29.3 Dependencies

A gate is created by the `camac_crate` object when the configuration information is read. A gate is associated with a scaler port and a gate port. These provide the interface for a gate to get the count of the number of gates and the gate width from which the gate computes its delay time and its gate time.

5.29.4 Interfaces

Files: none.

Windows: none.

5.30 gate_generator

5.30.1 Description

The `gate_generator` class models the generic functionality of a CAMAC gate generator module. (See Figure 8.)

5.30.2 Decomposition

This class is derived from the `camac` class. There is one subclass, the `camac_2323`.

5.30.3 Dependencies

A `gate_generator` object has gate ports associated with it. The gate ports retain the information necessary to program the gate generator.

5.30.4 Interfaces

Files: none.

Windows: none.

5.31 gate_port

5.31.1 Description

The `gate_port` class models a port or address on a CAMAC crate gate generator. (See Figure 9.)

5.31.2 Decomposition

This class is derived from the port class. There is one subclass, the `port_2323`.

5.31.3 Dependencies

A `gate_port` is associated with a `gate_generator`.

5.31.4 Interfaces

Files: none.

Windows: none.

5.32 hstreambuf

5.32.1 Description

The hstreambuf models the Hewlett-Packard LaserJet printer as a stream buffer.

5.32.2 Decomposition

This class is derived from filebuf, a class in the iostream library provided with the C++ language.

5.32.3 Dependencies

This class is defined whenever a user requests a file to be printed or a report is generated. It is associated with the print file.

5.32.4 Interfaces

Files: none.

Windows: none.

5.33 istream

5.33.1 Description

The istream class models an input token stream.

5.33.2 Decomposition

This class is derived from ifstream, a class in the iostream library provided with the C++ language.

5.33.3 Dependencies

The istream class is created by the main window or a chamber window whenever a file must be read as an input token stream.

5.33.4 Interfaces

Files: none.

Windows: none.

5.34 link

5.34.1 Description

The link class models a link used in constructing a linked list. (See Figures 2, 3, and 4.)

5.34.2 Decomposition

There is no superclass. There are no subclasses.

5.34.3 Dependencies

The link class is created by a list.

5.34.4 Interfaces

Files: none.

Windows: none.

5.35 list

5.35.1 Description

The list class models a generic linked list. This provides the facility to add, remove, insert, replace, and find items in a list. (See Figures 2, 3, and 4.)

5.35.2 Decomposition

There are no superclasses. Subclasses are `container_list`, `device_list`, `matrix_list`, and `user_list`.

5.35.3 Dependencies

The list class is not intended to be created itself.

5.35.4 Interfaces

Files: none.

Windows: none.

5.36 login_dlg

5.36.1 Description

The login dialog class provides the user interface to log into the system. (See Figure 2.) The login dialog validates the user against the user list.

5.36.2 Decomposition

This class is derived from `CDialog`, a class in the MFC library. There are no subclasses.

5.36.3 Dependencies

A login dialog is created by the main window when the menu item **Login...** is selected.

5.36.4 Interfaces

Files: none.

Windows: the login dialog (see Figure 19).

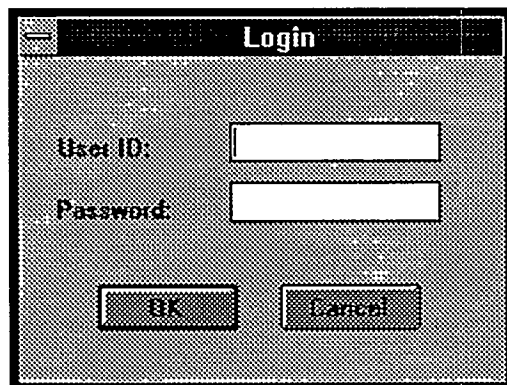


Figure 19. User login dialog window.

5.37 main_wnd

5.37.1 Description

The `main_wnd` class provides the overall management of the user interface and the operating system interface. (See Figure 1.) There is only one `main_wnd` object per program instance, and it creates and manages the main program window.

5.37.2 Decomposition

This class is derived from `CMDIFrameWnd`, a class in the MFC library. There are no subclasses.

5.37.3 Dependencies

This class creates the `user_list`, `container_list`, and the `matrix_list` objects as global resources. The `main_wnd` object creates chamber windows as child windows, as specified in the configuration (see Figure 19).

The main window is associated with the container dialog, the matrix dialog, and the user dialog. Also, the main window provides interface to the about dialog.

5.37.4 Interfaces

Files: AS.INI, the log file, the error file.

Windows: the main window (Figure 20).

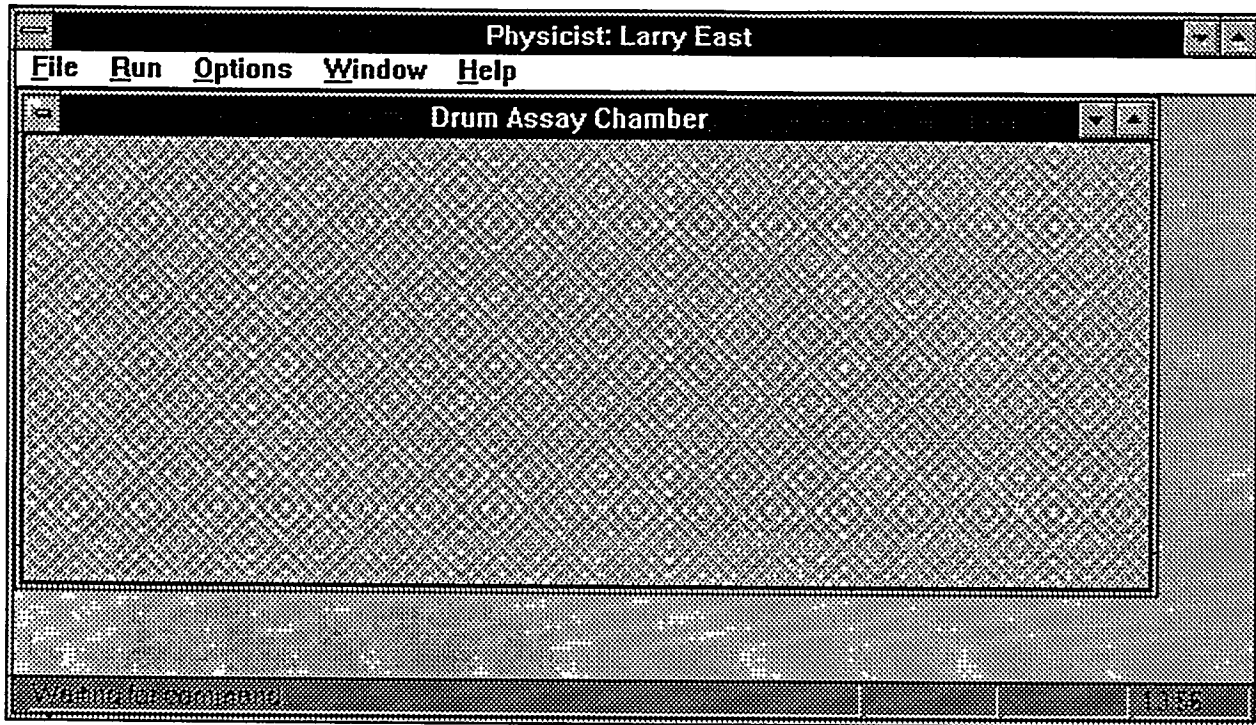


Figure 20. Main window with one chamber as a child window.

5.38 matrix

5.38.1 Description

The matrix class models the waste content of a container. (See Figure 4.) Active and passive signals are used to estimate the amount of plutonium, americium and (in some cases) uranium in the waste. Other derived attributes are computed based on the masses of plutonium, americium and uranium. These include thermal power, thermal power density, total alpha activity, and activity concentration.

The matrix object performs an assay by computing estimates for the plutonium mass based on the active and the passive data. A mass selection criteria determines which of these is reported as the plutonium mass.

5.38.2 Decomposition

There is no superclass. There are no derived classes.

5.38.3 Dependencies

A matrix object is created and managed by a chamber window. In addition, the matrix list and matrix dialog maintain the list of available matrixes.

The instances used in data reduction must be associated with a chamber in order to access the chamber calibration parameters. The matrix must read the correlation coefficients (for the chamber) from a

data file, the nuclide properties from a data file, and the mass fraction distribution of plutonium from a data file. Through the association with the chamber, the matrix has access to the active signals and the passive signals. Those signals are required to perform the data reduction.

The matrix is associated with a container that provides the volume used to compute the thermal power density.

The matrix generates a report of the assay results when requested.

5.38.4 Interfaces

Files: CORREL.DAC, NUCLIDE.DAT, gamma data file (default DEFAULT.GAM).

Windows: none.

Reports: summary report.

5.39 matrix_dlg

5.39.1 Description

The `matrix_dlg` object provides the user interface to modify the matrix list. (See Figure 4.) The key attributes of a matrix are the content code, type description, and whether it is a sludge (normal or americium bearing) or non-sludge.

5.39.2 Decomposition

This class is derived from `CDialog`, a class in the MFC library.

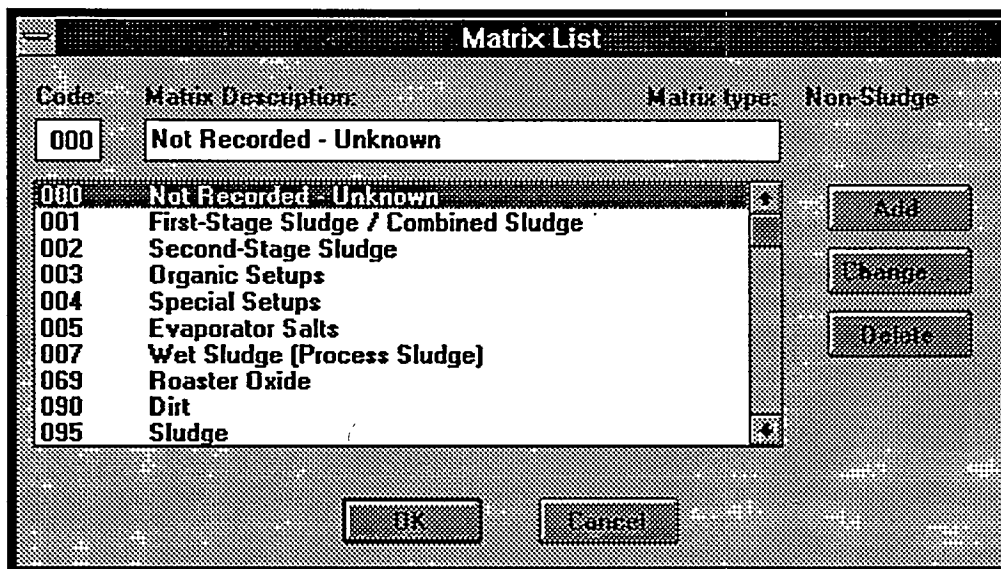


Figure 21. Matrix dialog window.

5.39.3 Dependencies

A `matrix_dlg` object is created by the main window when the menu item **Matrix List...** is selected from the "Physicist" menu. The matrix list provides the interface to add, remove, and find matrixes in the list.

5.39.4 Interfaces

Files: none.

Windows: the matrix dialog (Figure 21).

5.40 `matrix_list`

5.40.1 Description

The matrix list class models the list of matrixes that are processed at the assay station. (See Figure 4.)

5.40.2 Decomposition

This class is derived from the list class. There are no subclasses.

5.40.3 Dependencies

A single instance of the `matrix_list` is created by the main window. This `matrix_list` is used in the matrix dialog (Figure 20) to allow a Physicist to modify the list and in the waste dialog (Figure 27) to allow the user to select a particular matrix for processing.

5.40.4 Interfaces

Files: none.

Windows: none.

5.41 `matrix_parameter_dlg`

5.41.1 Description

The `matrix_parameter_dlg` class provides the user interface to modify selected parameters for the matrix class. (See Figure 4.) These parameters are the moderator index, the coefficients for the self absorption correction factor correlation, and the matrix correction factors (active and passive shielded coincidence and passive system coincidence) that are externally computed using a Monte Carlo simulation.

5.41.2 Decomposition

This class is derived from `CDialog`, a class in the MFC library.

5.41.3 Dependencies

This class is created by the matrix dialog whenever these attributes need to be changed.

5.41.4 Interfaces

Files: none.

Windows: the matrix parameters dialog (Figure 22).

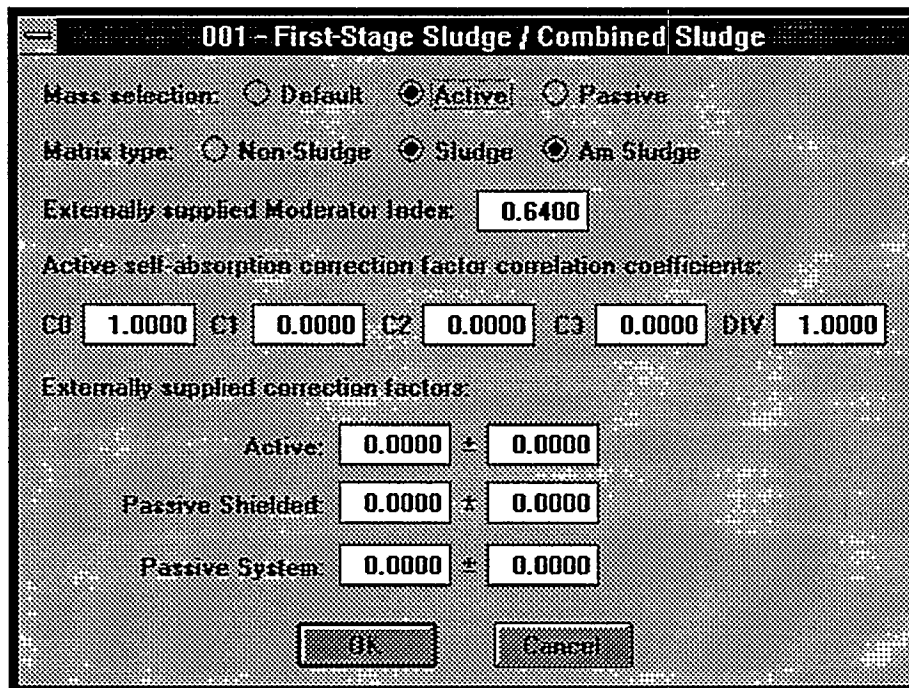


Figure 22. Matrix parameter dialog window.

5.42 message_dlg

5.42.1 Description

The message_dlg object is used to display a message associated with the detection of bad data. This class was added in Version 2.0. The standard MFC dialog box is not used in this case because it was desired to provide a custom icon and custom push buttons in the dialog window.

5.42.2 Decomposition

This class is derived from CDialog, a class in the MFC library.

5.42.3 Dependencies

A message_dlg object is created by a chamber in response to bad data detected by active_dlg or passive_dlg.

5.43.4 Interfaces

Files: none.

Windows: Message dialog window (Figure 23).

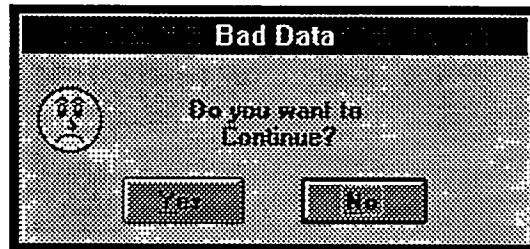


Figure 23. Message dialog window.

5.43 nuclide

5.43.1 Description

The nuclide object encapsulates the attributes and operations for nuclides. (See Figure 4.) The mass of a nuclide is defined as an error bar. Nuclide properties are its mass, mass fraction (for plutonium isotopes), mass ratio (relative to Pu-239), auxiliary mass ratio (nuclide dependent), alpha branching ratio, equivalent pu factor, fissile gram equivalent, specific activity and thermal power (several of these attributes were added in Version 2.0).

5.43.2 Decomposition

There is no superclass. There are no subclasses.

5.43.3 Dependencies

The matrix class creates instances of nuclide that describes the plutonium and americium isotopes.

5.43.4 Interfaces

Files: none.

Windows: none.

5.44 options_dlg

5.44.1 Description

The options_dlg class provides the user interface to change the options for data acquisition operations. (See Figure 10.) The options include the background count time, the active pulse count, and the passive count time.

5.44.2 Decomposition

This class is derived from CDialog, a class in the MFC library. There are no subclasses.

5.44.3 Dependencies

This class is created by the chamber window in response to selecting the menu item **Assay Times...**

5.44.4 Interfaces

Files: none.

Windows: the processing options dialog (Figure 24).

Drum Assay Chamber Operation Parameters			
Background Count Time		600	(secs)
Active Pulse Rate		50	(pulses/sec)
		Production	Verification (Pink Drum)
Passive Count Time	200	200	(secs)
Active Pulse	2000	2000	(counts)
OK		Defaults	Cancel

Figure 24. Processing options (assay times) dialog window.

5.45 passive_dlg

5.45.1 Description

The passive dialog shows the counts and rates from a passive data acquisition. (See Figure 10.) Both the bare and shielded counts are displayed for each of the six sides and the total. Rates for the shielded total, the system total, the shielded coincidence, and system coincidence are displayed with an error of one standard deviation. The sums of the six sides are checked against the CAMAC hardware totals. Additionally, the variation of the four vertical sides is checked against the average for both bare and shielded detectors.

5.45.2 Decomposition

This class is derived from CDialog, a class in MFC library. There are no subclasses.

5.45.3 Dependencies

This object is created by a chamber window when passive data are acquired or read from a file. The passive dialog creator function receives a pointer to a passive signals object through the argument list. The passive signals pointer provides the dialog with the interface to get the counts and rates that the dialog displays. The passive dialog window is a child window of the creating chamber window.

5.45.4 Interfaces

Files - none.

Windows - the passive dialog window (Figure 25) and, optionally, diagnostic message boxes.

Passive Data				
	Bare counts	Shielded counts	Net Rate $\pm$ One Sigma (counts/sec)	
Left:	5053	1173	Shielded Total:	1.883 $\pm$ 0.128
Back:	4909	1098	System Total:	13.03 $\pm$ 0.30
Right:	5692	1010	Shielded Coincidence:	-0.0059 $\pm$ 0.0147
Door:	6333	985	System Coincidence:	-0.050 $\pm$ 0.060
Top:	2922	579	Notes: Maximum side variation from average is 15 %. Bare Sum not equal to Bare Total.	
Bottom:	3555	1170		
Sum:	28464	6015		
Total:	28463	6015		

Figure 25. Passive dialog (data) window.

5.46 passive_signals

5.46.1 Description

The passive_signals class computes the net rates for the shielded total, the system total, the shielded coincidence, and the system coincidence signals. (See Figure 5.)

5.46.2 Decomposition

This class is derived from the signals class. There are no subclasses.

5.46.3 Dependencies

A passive_signals object is created by a chamber object. Pointers to the associated camac_crate object that maintains the passive gross data and the passive background data are passed as arguments to

the `passive_signals` creator function. The `camac_crate` provides the interface to get the gross counts, background counts, and count rates for the passive signals.

The shielded coincidence count is associated with the "short gate," and the system coincidence count is associated with the "long gate." These gates and the signal names are defined in the CAMAC crate configuration file. A change in the naming convention of the passive signals will require a change in the passive signals class.

The passive signals object is used by the `passive_dlg` object to display the passive data, and it is used by the matrix object in the data reduction.

5.46.4 Interfaces

Files - none.

Windows - none.

5.47 port

5.47.1 Description

The port class models a generic port of a CAMAC module. (See Figure 9.) Each port has a name.

5.47.2 Decomposition

There is no superclass. Subclasses are `clock_port`, `gate_port`, and `scaler_port`.

5.47.3 Dependencies

CAMAC objects create ports appropriate to the type of the module being modeled. A `camac_crate` can search the ports via the interface provided by each module type.

5.47.4 Interfaces

Files: none.

Windows: none.

5.48 port_2323

5.48.1 Description

The `port_2323` class models the gate port associated with a `camac_2323`. (See Figure 9.)

5.48.2 Decomposition

This class is derived from the `gate_port` class. There are no subclasses.

5.48.3 Dependencies

A `camac_2323` object creates an array of `port_2323` objects.

5.48.4 Interfaces

Files: none.

Windows: none.

5.49 results_dlg

5.49.1 Description

The results_dlg provides the user interface to view the results of a data analysis. (See Figure 10.) It displays the total plutonium and isotopic masses, uranium isotopic mass (if present) americium-241 mass, thermal power, thermal power density, activity values and fissile gram equivalent value. Each quantity is displayed with an error of one standard deviation.

The dialog provides push buttons to print the summary report, save the data file, and close the file. The summary report is automatically printed if the chamber window menu item summary report is checked. Data are automatically saved to a file at the end of an assay if the automatic save item is checked in the chamber window menu. The save and close functions are also provided under the chamber window's File menu.

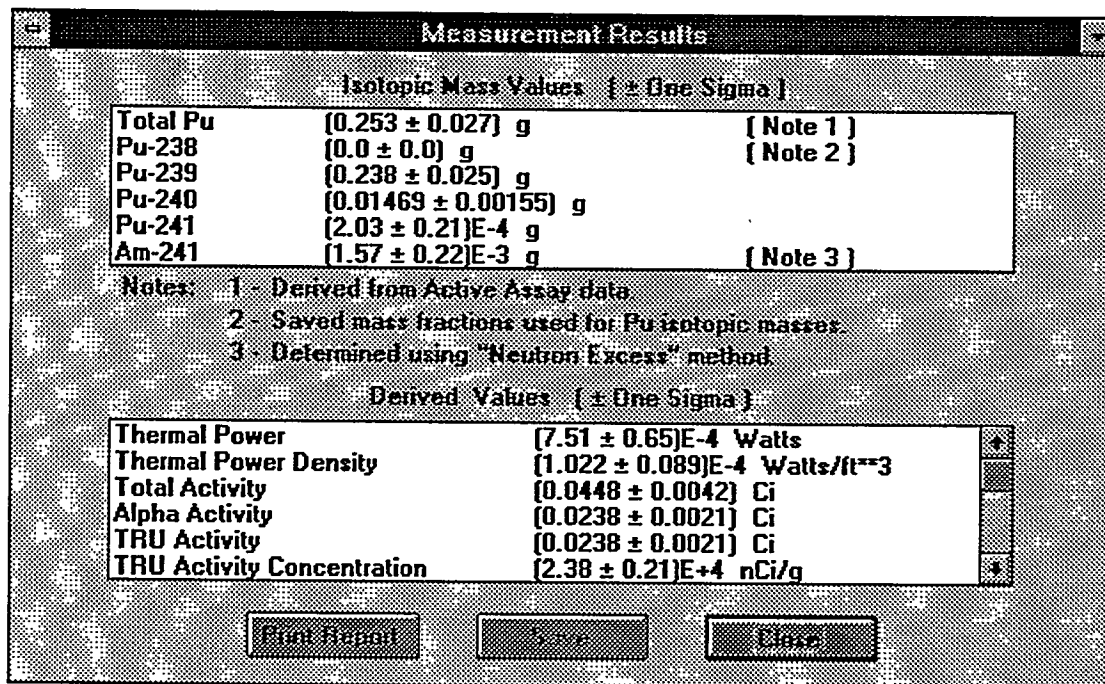


Figure 26. Results dialog window.

5.49.2 Decomposition

This class is derived from CDialog, a class in the MFC library. There are no subclasses.

5.49.3 Dependencies

A results_dlg object is created at the completion of a data reduction by the chamber window. This dialog is declared as a friend class in matrix so that it can access the quantities to be displayed.

5.49.4 Interfaces

Files: none.

Windows: the results dialog (Figure 26).

5.50 scaler

5.50.1 Description

The scaler class models a generic CAMAC scaler module. (See Figure 8.) It defines the interface to clear and read the scaler registers.

5.50.2 Decomposition

This class is derived from the camac class. Subclasses are camac_2551 and camac_450.

5.50.3 Dependencies

A scaler is associated with an array of scaler ports. This provides the interface to get and set the counts for individual ports.

5.50.4 Interfaces

Files: none.

Windows: none.

5.51 scaler_port

5.51.1 Description

The scaler_port class models the behavior of a port in a CAMAC scaler module. (See Figure 9.) The scaler port manages a device list.

5.51.2 Decomposition

This class is derived from the port class. There are no subclasses.

5.51.3 Dependencies

A scaler has an array of scaler ports. Each scaler port is associated with either a clock port or a gate port. The scaler_port manages a device list that provides the interface from which the camac_crate object can identify specific devices and retrieve counts, rates, and times.

The scaler_port provides the interface to find specific devices.

5.51.4 Interfaces

Files: none.

Windows: none.

5.52 signals

5.52.1 Description

The signals class models the generic functionality of the signals needed for data reduction. (See Figure 5.)

5.52.2 Decomposition

There is no superclass. Subclasses are active_signals and passive_signals.

5.52.3 Dependencies

There are no dependencies. The signals class is not intended to be instantiated. Rather, it abstracts certain behavior common to both the active and passive signals.

5.52.4 Interfaces

Files: none.

Windows: none.

5.53 timer

5.53.1 Description

The timer class defines the behavior of a (CAMAC) logical timer. (See Figure 7.)

5.53.2 Decomposition

This class is derived from the device class. There are no subclasses.

5.53.3 Dependencies

A timer object is associated with a scaler port by virtue of its inherited properties from device and with a clock port. The timer computes a time using the counts from the scaler port and the clock rate from the clock port.

5.53.4 Interfaces

Files: none.

Windows: none.

5.54 user

5.54.1 Description and Purpose

The user class defines the user name, ID, password, and access level for each user of the assay system. (See Figure 2.) The class reads and writes the user attributes in a data file.

5.54.2 Decomposition

There is no superclass. There are no derived classes.

5.54.3 Dependencies

The user object created by the main window is the login user, whose access level determines the functions that may be performed. The user list creates instances of user to define the list of all registered users. The users dialog creates instances of user to add to the user list.

5.54.4 Interfaces

Files: none.

Windows: none.

5.55 users_dlg

5.55.1 Description and Purpose

The users_dialog class provides the interface for the administrator to modify the list of users and user attributes. (See Figure 2.)

5.55.2 Decomposition

This class is derived from CDialog, a class in the MFC library. There are no subclasses.

5.55.3 Dependencies

The users dialog is created in the main window when a user with Administrator access level selects the menu item **User List** The dialog access users from the user list

5.55.4 Interfaces

Files: none.

Windows: the users dialog.

5.56 user_list

5.56.1 Description

The user_list class maintains the list of users who have access to the assay system. (See Figure 2.) This is not a "secure" list; primary system access control is accomplished by facility access procedures.

5.56.2 Decomposition

This class is derived from the list class. There are no subclasses.

5.56.3 Dependencies

The main window manages the global user list. The main window login function validates a login request against the user list.

Through the users dialog, the user list may be modified and written to its file, or it may be reread.

5.56.4 Interfaces

Files: USER.LST.

Windows: none.

5.57 waste_dlg

5.57.1 Description

The waste_dlg class provides the user interface to identify the waste container type, matrix type, and net weight of the container being processed. (See Figure 10.)

For new files, the user may input or select the attributes. The user selects the container type in a list selection box. The volume of a container is displayed, but cannot be modified through the waste dialog. The user selects the matrix type in another list selection box.

For an existing file, the users with Operator access privilege may not change the attributes, but they can be changed by a Physicist after selecting the "Edit" button at the bottom of the window (not shown to Operators)..

5.57.2 Decomposition

This class is derived from CDialog, a class in the MFC library.

5.57.3 Dependencies

The chamber window creates a waste dialog to process a new waste container or to perform recalculation on a previously processed container. The waste dialog calls the methods of both the container and the matrix classes to define a waste container for processing. The waste dialog also uses the global pointers to the container list and the matrix list.

5.57.4 Interfaces

Files: none.

Windows: the waste dialog (Figure 27).

C:\ASSAYSYSV2_TEST\DAC\PROD\SLUDG062.001

Identification:
sludg062.001

Container Type: 55 gallon drum Net Volume (L): 208.0

Matrix Type: 001 First-Stage Sludge / Combined Sludge Net Weight (kg): 101.0

Continue Exit Cancel

Figure 27. Waste dialog window.

6. REQUIREMENTS TRACEABILITY

The requirements traceability matrix in Table 3 identifies the requirement name and paragraph number from the software requirements specification (SRS), Reference [3]. The third column identifies the object where the requirement is designed and coded.

Table 3. Requirements traceability matrix.

SRS	Requirement	SDD / code
3.1.2.2 M.1	Login users	login_dlg, user, user_list, main_wnd
3.1.2.2 M.2	Present user menu functions	main_wnd
3.1.2.2 M.3	Logout users	main_wnd
3.1.2.2 M.4	Allow Administrator to modify user list	users_dlg, user, user_list, main_wnd
3.1.3.2 M.1	Read default processing options for assay system	main_wnd, chamber_wnd
3.1.3.3 M.1	Read default chamber processing options	chamber_wnd, options_dlg
3.1.3.3 M.2	Allow an Operator or Physicist to modify the processing options	chamber_wnd, options_dlg
3.1.3.3 M.3	Allow Physicist to modify the chamber calibration	calibration_dlg, chamber_wnd
3.1.3.3 M.4	Perform chi-square test	chamber_wnd
3.1.3.4 M.1	Read CAMAC crate configuration from file	camac_crate
3.1.3.4 M.2	Get time and date of data acquisition	camac_crate
3.1.3.4 M.3	Acquire active data from assay chamber	chamber, camac_crate
3.1.3.4 M.4	Store active data on file	camac_crate
3.1.3.4 M.5	Read active data from file	camac_crate
3.1.3.4 M.6	Acquire passive data from assay chamber	chamber, camac_crate
3.1.3.4 M.7	Store passive data on file	chamber, camac_crate
3.1.3.4 M.8	Read passive data from file	camac_crate
3.1.3.5 M.1	Compute flux monitor gross, background, and net rates	active_signals, detector
3.1.3.5 M.2	Compute flux monitor gross, background, and net rate errors	active_signals, detector, error_bar
3.1.3.5 M.3	Compute barrel flux monitor gross, background, and net rates	active_signals, detector
3.1.3.5 M.4	Compute barrel flux monitor gross, background, and net rate errors	active_signals, detector, error_bar
3.1.3.5 M.5	Compute shielded total gross, background, and net rates	active_signals, detector

SRS	Requirement	SDD / code
3.1.3.5 M.6	Compute shielded total gross, background, and net rate errors	active_signals, detector, error_bar
3.1.3.6 M.1	Compute shielded total rate	passive_signals, detector
3.1.3.6 M.2	Compute shielded total rate error	passive_signals, detector, error_bar
3.1.3.6 M.3	Compute system total rate	passive_signals, detector
3.1.3.6 M.4	Compute gross, background, and net system total rate error	passive_signals, detector, error_bar
3.1.3.6 M.5	Compute shielded coincidence rate	passive_signals, detector
3.1.3.6 M.6	Compute shielded coincidence rate error	passive_signals, detector, error_bar
3.1.3.6 M.7	Compute system coincidence rate	passive_signals, detector
3.1.3.6 M.8	Compute system coincidence rate error	passive_signals, detector, error_bar
3.1.4.2 M.1	Input container attributes	waste_dlg, container, chamber_wnd
3.1.4.2 M.2	Store container data on file	container
3.1.4.2 M.3	Read container data from file	container
3.1.4.2 M.4	Provide container summary report	container
3.1.4.2 M.5	Allow Physicist to modify the container list	container_dlg, container, container_list, main_wnd
3.1.4.3 M.1	Input matrix attributes	waste_dlg, matrix, chamber_wnd
3.1.4.3 M.2	Store matrix data on file	matrix
3.1.4.3 M.3	Read matrix data from file	matrix
3.1.4.3 M.4	Allow Physicist to modify the matrix list	matrix_dlg, matrix, matrix_list, main_wnd
3.1.4.3 M.5	Calculate active Pu mass	matrix
3.1.4.3 M.6	Calculate active Pu mass error	matrix, error_bar
3.1.4.3 M.7	Calculate passive Pu mass	matrix
3.1.4.3 M.8	Calculate passive Pu mass error	matrix, error_bar
3.1.4.3 M.9	Select either the active or passive Pu mass	matrix
3.1.4.3 M.10	Select either the active or passive Pu mass error	matrix, error_bar
3.1.4.3 M.11	Calculate plutonium isotopic masses	matrix
3.1.4.3 M.12	Calculate plutonium isotopic mass errors	matrix, error_bar
3.1.4.3 M.13	Calculate U-235 mass	matrix
3.1.4.3 M.14	Calculate U-235 mass error	matrix, error_bar
3.1.4.3 M.15	Calculate the Am-241 mass	matrix
3.1.4.3 M.16	Calculate the Am-241 mass error	matrix, error_bar
3.1.4.3 M.17	Calculate thermal power	matrix

SRS	Requirement	SDD / code
3.1.4.3 M.18	Calculate the thermal power error	matrix, error_bar
3.1.4.3 M.19	Calculate the thermal power density	matrix
3.1.4.3 M.20	Calculate the thermal power density error	matrix, error_bar
3.1.4.3 M.21	Calculate activity values	matrix
3.1.4.3 M.22	Calculate the activity errors	matrix, error_bar
3.1.4.3 M.23	Calculate the TRU activity concentration	matrix
3.1.4.3 M.24	Calculate the TRU activity concentration error	matrix, error_bar
3.1.4.3 M.25	Calculate Fissile Gram Equivalent	matrix
3.1.4.3 M.26	Calculate Fissile Gram Equivalent error	matrix, error_bar
3.1.4.3 M.27	Provide a measurement summary report for the container	matrix
3.1.5.2 M.1	Send "handshake" message to DMS	dms_dlg
3.1.5.2 M.2	Send measurement results to DMS	dms_dlg, chamber_wnd
3.1.5.2 M.3	Receive "handshake" message from DMS	dms_dlg
3.1.5.2 M.4	Receive new assay request from DMS	dms_dlg
3.1.5.2 M.5	Receive recalculation request from DMS	dms_dlg

7. DESIGN DETAIL

An overview of the software design can be seen by examining the header or "include" files that define C++ objects that make up the SAS software. A header file is associated with each class object developed for the program. Each header file identifies the public and private attributes and methods for each class. Associations with other classes are defined, typically through a C language pointer. A header file also identifies any superclass from which the class is derived, often a class defined in the Microsoft ® Foundation Class (MFC) Library that is part of the Visual C++™ development platform [16]. Appendix A lists each header file used in the SAS software.

8. FUTURE DEVELOPMENT DIRECTIONS

It is expected that this software will continue to evolve to support hardware changes planned for the SWEPP assay and DMS systems.

During the development of Version 2.0 of the SAS software, it became evident that some structural changes should be considered in order to reduce the complexity of the program objects. A "metrics study" performed on the source code about half way through the Version 2.0 upgrade development process indicated that the program was becoming less maintainable due to increasing complexity [15]. No restructuring was attempted due to time and resource constraints. Two obvious targets for complexity reduction are the "matrix" and "chamber_wnd" objects. The chamber_wnd object controls, directly or indirectly, most of the data acquisition and many of the user interface functions of the program. The matrix object contains all of the data reduction algorithms in addition to operations dealing directly with matrix properties of the waste container being measured. The feasibility of putting some of the functionality of these class objects into new (possibly derived) classes should be investigated. Program complexity could also be reduced by removing the option to emulate NEUT2 program behavior.

Functions that send output to a printer should be changed to use standard Windows™ printer support functions in order to take advantage of the high degree of device independence offered by these functions. At the present time, all printing is encapsulated in the "hstreambuf" object which was written specifically for HP LaserJet III and compatible printers.

Moving from the Windows platform to Windows-NT™ should be considered in order to take advantage of the security features and enhanced network and serial communications support available in Windows-NT.

Appendix A

Header Files

Appendix A

Header Files

Listings of the following program header files are contained in this appendix:

ABOUTDLG.H	ACQORDLG.H	ACTDLG.H	ACTSGNLS.H	ASSAYSYS.H
CALIBDLG.H	CAM217.H	CAM2323.H	CAM2551.H	CAM450.H
CAM8901.H	CAMAC.H	CAMCRATE.H	CHAMBER.H	CHAMBWND.H
CLOCKGEN.H	CLOKPORT.H	COINDET.H	CONTAIN.H	CONTDLG.H
CONTLIST.H	CONTROL.H	DELAYDLG.H	DETECTOR.H	DEVICE.H
DEVLIST.H	DMSDLG.H	ERRORBAR.H	FILELIM.H	GATE.H
GATEGEN.H	GATEPORT.H	HSTREAM.H	ITSTREAM.H	LINK.H
LIST.H	LOGINDLG.H	MAINWND.H	MAPARDLG.H	MATRDLG.H
MATRIX.H	MATRLIST.H	MENU.H	MSGDLG.H	NUCLIDE.H
PASDLG.H	PASSGNLS.H	PORT.H	PORT2323.H	PROPTDLG.H
RANGECHK.H	RESLTDLG.H	RESOURCE.H	RESRC1.H	SCALER.H
SCALPORT.H	SIGNALS.H	TIMER.H	USER.H	USERLIST.H
USERSDLG.H	WASTEDLG.H			

```

#ifndef INC_about_dlg_
#define INC_about_dlg_
////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile: aboutdlg.h $
// $Revision: 2.0 $
//
// 2. Function / Purpose
//
// The about dialog provides the user interface to identify the
// version and the build date of the application.
//
// 3. Interface Description
//
// The about dialog is derived from the Microsoft Foundation Class
// CDialog.
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:22 esm
// Initial release.
//
////////////////////////////////////////////////////////////////
class about_dlg : public CDialog
{
    char *program_id;
    char *build_date;
    char *copyright_date;

    COLORREF bkColor; // Background color.
    HBRUSH hbrBkgnd; // Background brush.

    afx_msg HBRUSH OnCtlColor( CDC*, CWnd*, UINT );
public:
    about_dlg( char *, char *, char * );
    ~about_dlg();

    BOOL OnInitDialog();

    DECLARE_MESSAGE_MAP()
};
#endif

```

```

#ifndef _INC_acqr_dlg_
#define _INC_acqr_dlg_

////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
//
// 1.2 Software Module
//
//      $Workfile:  acqrdlg.h  $
//      $Revision:  2.0  $
//
//
// 2. Function / Purpose
//
//      The acquire dialog provides the user interface to identify the
//      status of data acquisition.
//
//
// 3. Interface Description
//
//      The acquire dialog is derived from the Microsoft Foundation Class
//      CDialog.
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:26    esm
//      Initial revision.
//
////////////////////////////////////////////////////////////////

class chamber;
class motor;

class acquire_dlg : public CDialog
{
    char *title;
    chamber *pchamber;
    motor *pmotor;

    unsigned long pulse_rate;
    unsigned long num_secs;
    float total;
    float part_done;
    float last_part_done;
    float pct;

    int    wait_time;

```

```

    int update_time;

    CRect *rectStatus;
    CRect *rectBound;

    char avalue[8];
    BOOL cancel_flag;
    BOOL in_progress;
    BOOL do_background;

    void draw_background_bar( CDC * );
    void draw_status_bar( CDC * );
    void start_acquisition();
    void CancelAcquisition();
#ifdef LANL
    HWND m_hServer;
    CWnd *GetSmc();
    void StartServer();
#endif

    afx_msg void OnPaint();
    afx_msg void OnMove( int, int );
    afx_msg void OnDestroy();

public:
    acquire_dlg( chamber * );
    ~acquire_dlg();

    void set_title( char *title ) { this->title = title; }
    void set_pulse_rate( unsigned long pulse_rate )
        { this->pulse_rate = pulse_rate; }
    void check_progress();

    BOOL OnInitDialog();
    void OnCancel();

#ifdef LANL
    afx_msg LRESULT OnServerStarted( WPARAM wparam, LPARAM lparam );
    afx_msg LRESULT OnRotationStarted( WPARAM wparam, LPARAM lparam );
    afx_msg LRESULT OnRotationStopped( WPARAM wparam, LPARAM lparam );
#endif

    DECLARE_MESSAGE_MAP()
};

#endif

#ifdef _INC_active_dlg_
#define _INC_active_dlg_

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile: actdlg.h $
// $Revision: 2.0 $

```

INEL-96/0057

```
//
//
// 2. Function / Purpose
//
//   The active dialog provides the user interface to view the counts
//   and rates from the active data acquisition.
//
//
// 3. Interface Description
//
//
//
// 4. History
//
//
//   Version 2.0 Release, April, 1996 (L. V. East)
//   User interface improvements, additional data integrity tests and additional
//   calculational capabilities as outlined in INEL/RWMC System Change Request
//   No. 13623 (February, 1996). See Software Requirements Specification,
//   INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//   description of this program.
//
//   Rev 1.0   05 Oct 1993 15:25:20   esm
//   Initial release.
//
//
//
/////////////////////////////////////////////////////////////////

#include "filelim.h"

#define MAX_ACTIVE_BADVALUES      12

class active_signals;
class error_bar;
class chamber_wnd;

class active_dlg : public CDialog
{
    chamber_wnd *parent;

    active_signals *active;

    char buffer[ RECORD_LENGTH ];

    HBRUSH hbrBkgnd;    // Background brush to use for highlighted message boxes.
    COLORREF bkgColor;  // Color to use for background brush.

    BOOL bad_rate;
    BOOL bad_count;
    int bad_valueIDs[ MAX_ACTIVE_BADVALUES ];
    int bad_values;

    void set_count( char *, int, int );
    void set_rate( error_bar, int, int );

    afx_msg HBRUSH OnCtlColor( CDC*, CWnd*, UINT );
    afx_msg void OnClose();
    afx_msg void OnLButtonDown( UINT, CPoint );
    afx_msg void OnPaint();

public:
    active_dlg( active_signals *, chamber_wnd * );
    ~active_dlg();

    void rate_check();

    BOOL OnInitDialog();
};
```

```

        DECLARE_MESSAGE_MAP()
};

#endif

#ifndef _INC_active_signals_
#define _INC_active_signals_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:  actsgnls.h  $
//      $Revision:  2.0  $
//
// 2. Function / Purpose
//
//      The active signals provides the interface to the active data to
//      access the shielded total, the flux monitor, and the barrel flux
//      monitor signals.
//
// 3. Interface Description
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:32    esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

#pragma hdrstop

#include "signals.h"
#include "camcrate.h"
#include "errorbar.h"

class active_signals : public signals
{
    error_bar get_gross_rate( char *name )
        { return signals::get_gross_rate( name, "EARLY_GATE" ); }
    error_bar get_background_rate( char *name )
        { return signals::get_gross_rate( name, "LATE_GATE" ); }
}

```

```

        error_bar get_rate( char *name, float adj_factor = 1.0f )
        { return get_gross_rate( name ) - adj_factor * get_background_rate(
name ); }

public:
    active_signals( camac_crate *active_crate ) : signals( active_crate,
        active_crate )
    {;}
    virtual ~active_signals() {;}

    unsigned long get_gross_count( char *name )
    { return signals::get_gross_count( name, "EARLY_GATE" ); }
    unsigned long get_background_count( char *name )
    { return signals::get_background_count( name, "LATE_GATE" ); }

    unsigned long get_pulse_count()
    { return signals::get_gross_count( "ACTIVE_PULSE" ); }

    error_bar get_shielded_total_rate( float adj_factor = 1.0f )
    { return get_rate( "SHIELDED_TOTAL", adj_factor ); }
    error_bar get_flux_monitor_rate()
    { return get_rate( "FLUX_MONITOR" ); }
    error_bar get_barrel_flux_monitor_rate()
    { return get_rate( "BARREL_FLUX_MONITOR" ); }

};

#endif

```

```

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:  assaysys.h  $
//      $Revision:  2.0  $
//
// 2. Function / Purpose
//
//      Assay System is the application object.  Microsoft Windows runs
//      the application as an instance of this object.
//
// 3. Interface Description
//
//      The normal application object methods are provided to create,
//      destroy and initialize an instance of the object.
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request

```


INEL-96/0057

```

BOOL near_zero( float );           // Returns TRUE if arg is essentially equal
to zero.
int places( float );               // Determine number of significant digits.
void set_status_text( const char *, BOOL = TRUE ); // Set text in status bar.
char *system_time();               // Returns current system date/time as ASCII string.

// assay_system class definitions.

class assay_system : public CWinApp
{
    void InitMemoryLeakDetection();
    void ExitMemoryLeakDetection();

public:
    assay_system();
    ~assay_system() { }

    virtual BOOL InitInstance();
    virtual BOOL ExitInstance();
};

#ifdef _INC_calibration_dlg_
#define _INC_calibration_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    calibdlg.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The calibration dialog provides the user interface to change
//      the chamber calibration attributes.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:32    esm
//      Initial revision.
//
//

```

```

////////////////////////////////////
#include "errorbar.h"

class chamber;

class calibration_dlg : public CDialog
{
    chamber *pchamber;

    void set_error_bar( error_bar, int, int );
    error_bar get_error_bar( int, int );

    float get_float( int );

public:
    calibration_dlg( chamber * );
    ~calibration_dlg() {}

    BOOL OnInitDialog();

    void OnOK();
};

#endif

#ifndef _INC_camac_217_
#define _INC_camac_217_

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    cam217.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The CAMAC 217 models the Jorway 217 Gated Clock Generator.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//

```

INEL-96/0057

```
// Rev 1.0   05 Oct 1993 15:25:28   esm
// Initial release.
//
//
// 5. References
//
// 5.1 CAMAC Model 217, Gated Clock Generator with Option 1,
//      Jorway Corporation.
//
//
////////////////////////////////////

#define JORWAY_217_GATED_CLOCK_GENERATOR    217

class camac_217 : public clock_generator
{
    int control_word;

    void set_clock_on( unsigned char );
    // void set_clock_off( unsigned char );           // inactive, untested

    void set_mux_clock( unsigned char );

    unsigned int write_control_word();

public:
    camac_217( controller **, unsigned char );
    virtual ~camac_217();

    int initialize();

    // unsigned int enable_clock( unsigned char );    // inactive, untested
    // unsigned int disable_clock ( unsigned char ); // inactive, untested

    // unsigned int start_gate();                     // inactive, untested
    // unsigned int stop_gate();                      // inactive, untested

    virtual int test() { return 1; }

};

#endif

#ifndef _INC_camac_2323_
#define _INC_camac_2323_

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
//
// 1.2 Software Module
//
//      $Workfile:    cam2323.h  $
//      $Revision:    2.0  $
//
//
// 2. Function / Purpose
//
```

```

// The CAMAC 2323 models the LeCroy 2323 Dual Gate and Delay Generator.
//
//
// 3. Interface Description
//
//
// 4. History
//
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:26 esm
// Initial release.
//
//
// 5. References
//
// 5.1 Operator's Manual, CAMAC Model 2323A, Dual Gate and Delay Generator
// LeCroy, May 1987.
//
//
/////////////////////////////////////////////////////////////////

#define LECROY_2323_DUAL_GATE_GENERATOR2323

class camac_2323 : public gate_generator
{
    int write_channel( unsigned char );

//    unsigned int read_channel( unsigned char ); // inactive, untested
//    void set_gate_port( unsigned char ); // inactive, untested
    virtual void set_gate_mode( unsigned char, unsigned char );
    virtual unsigned char get_gate_mode( unsigned char );

public:
    camac_2323( controller **, unsigned char );
    virtual ~camac_2323();

    virtual int initialize();
    virtual void read_data( itstream * );

    unsigned int start( unsigned char );
    unsigned int stop( unsigned char );

    virtual int test() { return 1; }
};

#endif

#ifndef _INC_camac_2551__
#define _INC_camac_2551__

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097

```

```
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
//      1.2 Software Module
//
//          $Workfile:   cam2551.h  $
//          $Revision:   2.0  $
//
//
//      2. Function / Purpose
//
//          The CAMAC 2551 models the LeCroy 2551 12 Channel 100MHz Scaler.
//
//
//      3. Interface Description
//
//
//
//      4. History
//
//          Version 2.0 Release, April, 1996 (L. V. East)
//          User interface improvements, additional data integrity tests and additional
//          calculational capabilities as outlined in INEL/RWMC System Change Request
//          No. 13623 (February, 1996). See Software Requirements Specification,
//          INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//          description of this program.
//
//          Rev 1.0    05 Oct 1993 15:25:34    esm
//          Initial revision.
//
//
//      5. References
//
//          5.1 Operator's Manual, CAMAC Model 2551, 12 Channel 100MHz Scaler,
//              LeCroy, March 1983.
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#define LECROY_2551_SCALER        2551

class camac_2551 : public scaler
{
    int clear_module_and_lam();
    void enable_lam();

//    void increment_all_registers(); // inactive, untested
//    void disable_lam();             // inactive, untested

protected:
    virtual void accumulate_all_registers();
    virtual void clear_all_registers();

    virtual void clear_lam();

public:
    camac_2551( controller **, unsigned char );
    virtual ~camac_2551() {}

    virtual int initialize();
    virtual int test_lam();

    virtual int test() { return 1; }
};
```

```
#endif
```

```
#ifndef _INC_camac_450_
#define _INC_camac_450_
```

```

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   cam450.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The CAMAC 450 models the DSP 2551 Model QS-450, 100MHz Quad Scaler.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:26   esm
//      Initial release.
//
// 5. References
//
//      5.1 Operator's Manual, CAMAC Model QS-450, 100MHz Quad Scaler,
//      DSP.
//
////////////////////////////////////

#define DSP_450_SCALER   450

class camac_450 : public scaler
{
//      unsigned char read_module_id_number();           // inactive, untested
//      int test_register_lam( unsigned char );
//      void clear_register( unsigned char );

protected:
    virtual void accumulate_all_registers();
    virtual void clear_all_registers();
    virtual void clear_lam();

```

```

public:
    camac_450( controller **, unsigned char );
    virtual ~camac_450() {}

    virtual int test_lam();

    virtual int test() { return 1; }
};

#endif

#ifndef _INC_camac_8901_
#define _INC_camac_8901_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    cam8901.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The CAMAC 8901 models the LeCroy 8901 GPIB Interface crate controller.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:32    esm
//      Initial release.
//
// 5. References
//
//      5.1 Technical Reference Manual, Model 8901A GPIB Interface,
//          LeCroy.
//
/////////////////////////////////////////////////////////////////

#define LECROY_8901_CONTROLLER    8901

```

```

#define MIN_DEVICE 1
#define MAX_DEVICE 16

#define WRITE_BUF_LEN 6
#define READ_BUF_LEN 4
#define DEV_NAME_LEN 6

#include "control.h"

class camac_8901 : public controller
{
    short status;
    short int gpib_id;
    short int crate_id;
    char crate_name[ DEV_NAME_LEN ];
    char gpib_name[ DEV_NAME_LEN ];

    char *cmd;
    char write_buffer[ WRITE_BUF_LEN ];
    char read_buffer[ READ_BUF_LEN ];

private:
    void set_four_word_transfer();

    void set_station( unsigned char );
    void set_function( unsigned char );
    void set_address( unsigned char );

    void write_data( unsigned long );
    unsigned long read_data();
    void camac_cycle( unsigned );

public:
    camac_8901();
    virtual ~camac_8901() {};

    virtual int set_crate( unsigned char );

    // General CAMAC crate functions

    virtual void initialize();
    virtual void clear();
    virtual void set_inhibit();
    virtual void clear_inhibit();
    virtual int test_lam();

    // CAMAC module I/O interface

    virtual unsigned long read( unsigned char, unsigned char, unsigned char );
    virtual int write( unsigned char, unsigned char, unsigned char, long );
    virtual void execute( unsigned char, unsigned char, unsigned char );

    virtual int test() { return 1; }
};

#endif

#ifndef _INC_camac_
#define _INC_camac_

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//

```

[illegible]

```

// Reserved                29
// Nonstandard              30
// Reserved                 31

#define LAM_CHECK_OFF      0
#define LAM_CHECK_ON       1

#include "control.h"
#include "port.h"

class itstream;
class ostream;

class camac
{
    unsigned char i_port;      // index to port

protected:
    controller **cc;
    unsigned char station;     // valid 1 through 23
    unsigned int model;        // manufacturer's model number
    unsigned int type;         // generic board type
    unsigned char min_reg;
    unsigned char max_reg;

    unsigned char lam_check : 1;
    port *reg[MAX_ADDRESS+1]; // registers or subaddresses in camac

public:
    camac( controller **, unsigned char, unsigned int );
    camac( controller **, unsigned char, unsigned int, unsigned int, unsigned
          char, unsigned char );
    virtual ~camac() {}

    void set_register_name( unsigned char, char * );
    char *get_register_name( unsigned char );

    unsigned char find_register( char * );
    port *find_port( char * );
    port *get_first_port();
    port *get_next_port();

    void set_lam_check( unsigned char lam_check ) { this->lam_check = lam_check; }

    virtual int initialize() { return 0; }

    // return station number if this board matches the query

    unsigned char get_station() { return station; }
    int is_model( int model )
        { return( model == (int)this->model ? station : 0 ); }
    int is_type( int type )
        { return( type == (int)this->type ? station : 0 ); }

    // Interface to crate controller

    void write( unsigned char function, unsigned char address, long data )
        { (*cc)->write( station, function, address, data ); }
    void write( unsigned char function, unsigned char address, int data[2] )
        { (*cc)->write( station, function, address, (long) data ); }
    void write( unsigned char function, unsigned char address, int data )
        { (*cc)->write( station, function, address, (long) data ); }
    unsigned long read( unsigned char function, unsigned char address )
        { return (*cc)->read( station, function, address ); }
    void execute( unsigned char function, unsigned char address )
        { (*cc)->execute( station, function, address ); }
    void execute( unsigned char function )
        { (*cc)->execute( station, function, 0 ); }

```

```

    unsigned int get_response() { return (*cc)->get_response(); }
    unsigned int get_command_accepted() { return (*cc)->get_command_accepted(); }

    virtual void read_data( istream * ) {}
    virtual void write_data( ostream * );
    virtual int test() { return 1; } // each board defines diagnostic tests
};

#endif

#ifndef _INC_camcrate_
#define _INC_camcrate_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   camcrate.h   $
//      $Revision:   2.0   $
//
// 2. Function / Purpose
//
//      The camac_crate objec models the CAMAC crate.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:30   esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

#include <fstream.h>
#include <time.h>

#include "control.h"

class camac;
class clock_generator;
class gate_generator;
class scaler;
class device;

```

```

class error_bar;
class itstream;
class port;

class camac_crate
{
    char *name;

    unsigned char number;          // crate number
    controller *cc;                // crate controller
    camac *board[MAX_STATION]; // CAMAC boards

    // internally used
    time_t ltime;
    unsigned char check_station;
    unsigned char check_port;
    unsigned long check_limit;
    float scale_factor;
    float part_accumulated;

    void delete_configuration();
    void initialize_boards();
    int test_lam();

    int set_devices();
    port *find_port( char *, int );
    device **find_device( char *name, char *gate_name = NULL );

    void compute_device_values( int );
public:
    camac_crate( char * );
    ~camac_crate();

    char *get_name() { return name; }
    unsigned long get_check_limit();

    int read_data( itstream * );
    void write_data( ofstream * );

    int initialize( unsigned long );
    void compute_values();

    void start_acquisition();
    void stop_acquisition();

    float check_progress();
    void accumulate();

    time_t get_time() { return ltime; }
    unsigned long get_count( char *name, char *gate_name = NULL );
    error_bar get_rate( char *name, char *gate_name = NULL );

    int test();
};

#endif

#ifndef _INC_chamber_
#define _INC_chamber_

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//

```



```

camac_crate *passive_background;
camac_crate *passive_gross;
camac_crate *pcamac_crate;

public:
    active_signals *active;
    passive_signals *passive;
    passive_signals *background;

    chamber( char * );
    ~chamber();

    void set_file_name( char * );
    int can_write() { return has_write_permission; }
    int read_calibration_data();
    int read_calibration_data( istream * );
    void write_calibration_data();
    void write_calibration_data( ostream * );

    char *get_name() { return name; }

    error_bar get_shielded_detector_efficiency()
    { return shielded_detector_efficiency; }
    error_bar get_system_detector_efficiency()
    { return system_detector_efficiency; }
    error_bar get_active_zero_matrix_calibration_factor()
    { return active_zmcf; }
    error_bar get_shielded_coincidence_zero_matrix_calibration_factor()
    { return shielded_coincidence_zmcf; }
    error_bar get_system_coincidence_zero_matrix_calibration_factor()
    { return system_coincidence_zmcf; }

    error_bar get_fraction_Pu_239() { return fraction_Pu_239; }
    error_bar get_fraction_Pu_240() { return fraction_Pu_240; }

    time_t get_time() { return pcamac_crate->get_time(); }
    time_t get_active_time() { return active_crate->get_time(); }
    time_t get_passive_time() { return passive_gross->get_time(); }
    time_t get_background_time() { return passive_background->get_time(); }

    char *get_crate_name() { return pcamac_crate->get_name(); }
    unsigned long get_check_limit() { return pcamac_crate->get_check_limit(); }

    int initialize_active( unsigned long );
    int initialize_passive( unsigned long );
    int initialize_background( unsigned long );

    void start_acquisition() { pcamac_crate->start_acquisition(); }
    float check_progress() { return pcamac_crate->check_progress(); }
    void stop_acquisition() { pcamac_crate->stop_acquisition(); }
    void accumulate() { pcamac_crate->accumulate(); }

    void set_background_correction( unsigned int background_correction )
    { passive->set_background_correction( background_correction ); }

    int read_active_data( istream *data )
    { return active_crate->read_data( data ); }
    int read_passive_data( istream *data )
    { return passive_gross->read_data( data ); }
    int read_background_data( istream *data )
    { return passive_background->read_data( data ); }

    void compute_active_values()
    { active_crate->compute_values(); }
    void compute_passive_values()
    {
        passive_gross->compute_values();
    }

```

```

        passive_background->compute_values();
    }
    void compute_background()
    { passive_background->compute_values(); }

    void write_background( ofstream *out_file )
    { passive_background->write_data( out_file ); }
    void write_passive( ofstream *out_file )
    { passive_gross->write_data( out_file ); }
    void write_active( ofstream *out_file )
    { active_crate->write_data( out_file); }

    int test_crate() { return pcamac_crate->test(); }
};

#endif

#ifndef _INC_chamber_wnd_
#define _INC_chamber_wnd_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
//     1.1 Project
//
//         Title: SWEPP Assay System
//         Project ID: Task Baseline Agreement SDM-92-097
//         Developed by: Scientific Computing Unit, INEL
//         Funded by: Transuranic Waste Programs Unit, INEL
//
//     1.2 Software Module
//
//         $Workfile:   chambwnd.h  $
//         $Revision:   2.0  $
//         $Date:       24 Jan 1994 11:53:50  $
//
// 2. Function / Purpose
//
//     The "chamber window" is the parent for child windows showing assay
//     data and results for a specific hardware assay chamber configuration.
//
// 3. Interface Description
//
//     The normal application object methods are provided to create,
//     destroy and initialize an instance of the object.
//
// 4. History
//
//     Version 2.0 Release, April, 1996 (L. V. East)
//     User interface improvements, additional data integrity tests and additional
//     calculational capabilities as outlined in INEL/RWMC System Change Request
//     No. 13623 (February, 1996). See Software Requirements Specification,
//     INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//     description of this program.
//
//     Rev 1.2   24 Jan 1994 11:53:50   esm
//     Modified attributes needed for chi-square report per SPR 014.
//
//     Rev 1.1   22 Dec 1993 11:47:30   esm
//     Added funtion to check for disk free space. SPR #005
//

```

```

// Rev 1.0   05 Oct 1993 15:25:30   esm
// Initial release.
//
//
/////////////////////////////////////////////////////////////////

#include <fstream.h>
#include <stdlib.h>
#include "errorbar.h"

// Private message definition -- used to notify waste_dlg of a menu item change.
#define STATUS_CHANGED (WM_USER + 501 )

// auto state values
#define AUTO_OFF      0
#define AUTO_RUN      1
#define AUTO_PAUSE -1

class auto_dlg;
class chamber;
class container;
class dms_dlg;
class error_bar;
class itstream;
class matrix;

class chamber_wnd : public CMDIChildWnd
{
    friend class process_options_dlg;
    friend class waste_dlg;
    friend class active_dlg;
    friend class passive_dlg;
    friend class results_dlg;

    static dms_dlg *pdms_dlg;
    static auto_dlg *pauto_dlg;
    static BOOL NEAR is_remote_operation;
    static int NEAR auto_mode_state;
    static int NEAR chamber_count;           // Number of active chamber windows.
    static int NEAR arch_flag;              // Last archive drive requested.

    CWnd *parent_wnd;

    chamber *pchamber;
    matrix *content;

    itstream *ac_ini;

    unsigned long bkg_secs;
    unsigned long num_pulses;
    unsigned long num_secs;
    unsigned long ver_pulses;
    unsigned long ver_secs;
    unsigned long pulse_rate;

    int    passive_delay;    // Delay before starting passive after active assay.
    int last_code;          // Most recently used content code.

    BOOL active_first;      // Determines order of active & passive data
                           // acquisition.
    BOOL auto_save;         // If TRUE, automatically save data after an assay.
    BOOL background_correction; // Passive background correction flag.
    BOOL good_background;   // TRUE only if background data-read successful.
    BOOL include_U233;      // If TRUE, report U-233 mass (based in mass ratio info)
    BOOL include_U235;      // If TRUE, report U-235 mass (based in mass ratio info)
    BOOL include_Pu242;     // If TRUE, report Pu-242 mass.
    BOOL summary_printed;   // TRUE if summary report printed.

```

```

    BOOL summary_report;        // Print ON/OFF flag.

    COLORREF fill_color;

    CString str_profile_heading;

    char file_name[_MAX_FNAME];
    char last_file_name[_MAX_FNAME];
    char *path_name;
    char cw_dir[_MAX_PATH];    // Current default directory for this chamber.
    int  cw_drive;             // Current default disk drive for this chamber.

    char container_ID[32];     // ID applied to waste container by waste generator.
    char archive_ID[16];      // Administrative label for archive diskette.

    waste_dlg *p waste_dlg;
    active_dlg *p active_dlg;
    passive_dlg *p passive_dlg;
    results_dlg *p results_dlg;

    // directories
    char *root_dir;
    int  root_drive;

    char *arch_dir;
    int  arch_drive;

    char *gamma_dir;
    char *backg_dir;
    char *verif_dir;
    char *calib_dir;
    char *prod_dir;

    // files
    char *options_name;
    char *backg_name;
    char *verif_name;
    char *passive_cfg;
    char *active_cfg;
    char *correl_dat;
    char *chi_sq_dat;
    char *nuclide_dat;
    char *default_mass_dat;
    char *gamma_dat;
    char *regress_dat;

    // other stuff...
    int state;                // See chambwnd.cpp for "state" definitions.
    BOOL bad_data;            // TRUE if bad data discovered during assay or
recalc.
    BOOL is_file_saved;
    BOOL is_active_done;
    BOOL is_passive_done;
    BOOL is_background_done;
    BOOL is_verification;

    unsigned int chi_square_sample_size;
    unsigned int chi_square_count_time;
    float chi_square_acceptance_min;
    float chi_square_acceptance_max;
    float chi_square_confidence;
    float chi_sq_shielded_total;
    float chi_sq_system_total;

    unsigned long *shielded_total;
    unsigned long *system_total;

```

```

double av_shielded_total;
double av_system_total;

BOOL accept_shielded_total;
BOOL accept_system_total;

error_bar nominal_Pu240_fraction;
error_bar nominal_Pu241_fraction;

int gamma_flag;
BOOL use_gamma_Pu_data;

time_t gamma_date;          // Date/time gamma-ray measurement performed.
char SGRS_fileID[12];       // Unique part of SGRS file name.

void extract_gamma_date( itstream * ); // Extracts date/time from SGRS file
                                         // name.

void set_menu();
void set_path_name( char *, char * );

BOOL read_chi_square_parameters( char * );
void chi_square_report();

BOOL in_range( float min, float max, float value )
    { return min <= value && value <= max; }
BOOL valid_eb_value( error_bar &value )
    { return value.datum != -1.0f && value.error != -1.0f; }

char *read_name( char * );
char *new_path_name( char *, char * );
void check_space( const unsigned );

int select_file_name( BOOL, const char *, char * );
void find_gamma_file();
void find_old_files();
BOOL copy_file( char *, int, ofstream *, char * = NULL );
BOOL set_dir( int, char *, char * = NULL );
void make_temp_name( char * );

BOOL create_content( char * );
void prepare_calc();
BOOL prepare_recalc();
BOOL ok_to_continue();

BOOL get_gamma_data( char *, int = 0 );

void on_new_file();
void on_open_file();
void on_close_file();

void on_continue();

void recalculate();
void calculate();
void dms_recalc();

void assay();
void acquire_active();
void acquire_passive();
void active_only();
void passive_only();
void acquire_background();

void on_verification();
void on_chi_square();
void on_accept_bkgnd();

```

```

void on_reject_bkgnd();

void do_delay( int );

void set_color();
void set_process_options();
void toggle_summary_report();
void toggle_auto_save();
void toggle_background_correction();
void toggle_auto_mode();
BOOL toggle_menu_item( unsigned int, BOOL );

void on_calibration();

void regression_test();
void test_crate();
int crate_error_message( int );
void get_file_name( itstream *, char *, char * );
char *log_time();

BOOL read_passive_bkgnd();
BOOL is_bkgnd_too_old();

BOOL write_data( const char *, BOOL = FALSE, BOOL = FALSE, BOOL = FALSE );
void print_summary_report();

void restore_focus();

BOOL show_rmenu();

afx_msg void OnClose();
afx_msg void OnDestroy();
afx_msg void OnMDIActivate( BOOL, CWnd *, CWnd * );
afx_msg void OnPaint();
afx_msg void OnSetFocus( CWnd * );
afx_msg void OnUpdateStatus( CCmdUI* );// Private, not system...

public:
    chamber_wnd( itstream * );
    ~chamber_wnd();

    static const char *wait_text;

    BOOL bactive_wnd;           // TRUE if this is the currently active window
    BOOL use_current_bkgnd;     // TRUE if user wants to use current rather than
                                // saved background for a recalculation.
    BOOL use_current_pmtrs;     // TRUE if user wants to use current matrix, and
                                // calibration parameters for a recalculation.
    float nominal_SG_bkgnd;     // Nominal coincidence backgrounds -- used in
    float nominal_LG_bkgnd;     // background data integrity tests.
    float max_bkgnd_age;        // Max hours before new background required.
    float max_var_base;         // Base value of maximum allowed fractional
                                // variation of individual detector bank counts
                                // from the average. 2.5 * (S.D. of average)
                                // gets added to this base value.

#ifdef LANL
    BOOL motor_control;        // If TRUE, chamber has software controlled
truntable
#endif

    static void set_auto_state( int state )
        { auto_mode_state = state; }

    char *get_name();
    void set_dms_dlg( dms_dlg *pdms_dlg );

    BOOL read_data();

```

```

    BOOL read_process_options();
    BOOL set_window();

    void remote_assay();
    void remote_recalc();

    void set_bad_data( BOOL arg ) { bad_data = arg; }

    BOOL Create( const RECT &rect = rectDefault, CMDIFrameWnd *parent = NULL );
    virtual BOOL DestroyWindow();

    afx_msg void on_save_file();

    DECLARE_MESSAGE_MAP()
};

#endif

#ifndef _INC_clockgen_
#define _INC_clockgen_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    clockgen.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The clock generator models a generic CAMAC clock generator board.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:30    esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

#define CLOCK_GENERATOR 101

#include "camac.h"

```

INEL-96/0057

```
class itstream;

class clock_generator : public camac
{
    void set_clock_rate( unsigned char, unsigned long );

public:
    clock_generator( controller **cc, unsigned char station,
        unsigned int model, unsigned char min_reg,
        unsigned char max_reg )
        : camac( cc, station, model, CLOCK_GENERATOR, min_reg, max_reg )
    {}

    virtual ~clock_generator() {}

    virtual void read_data( itstream * );
    virtual int initialize() { return 0; }

    virtual int test() { return 1; }
};

#endif

#ifndef _inc_clokport_
#define _inc_clokport_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   klokport.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The clock_port class models a port or address on a CAMAC
//      clock generator. A clock port generates clock pulses as a
//      prescribed rate.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
// Rev 1.0   05 Oct 1993 15:25:24   esm
```

```

// Initial revision.
//
//
////////////////////////////////////////////////////////////////
#include "port.h"

class ostream;

class clock_port : public port
{
    unsigned long rate;

public:
    clock_port() { rate = 0L; }
    virtual ~clock_port() {}

    void set_rate( unsigned long rate ) { this->rate = rate; }
    unsigned long get_rate() { return rate; }

    virtual void write_data( ostream * );
};

#endif

#ifndef _INC_coindet
#define _INC_coindet_

////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   coindet.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The coincidence_detector object models a coincidence detector device.
//
// 3. Interface Description
//
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:30   esm
//      Initial release.

```

INEL-96/0057

```
//
//
////////////////////////////////////////////////////////////////

#define COINCIDENCE_DETECTOR      401

#include "detector.h"

class coincidence_detector : public detector
{
    detector **hdetector;
public:
    coincidence_detector( scaler_port *, gate **, timer **, detector ** );
    virtual ~coincidence_detector() {};

    virtual void compute_values();
};

#endif

#ifndef _INC_contain_
#define _INC_contain_

////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    contain.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The container class models the behavior of a generic container
//      process at the assay station.
//
// 3. Interface Description
//
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:20   esm
//      Initial release.
//
//
////////////////////////////////////////////////////////////////
```

```

#include <iostream.h>

class itstream;

class container
{
    char *type;
    float volume;
    char *name;           // identification

public:
    container();
    container( char * );
    container( char *, float );
    container( const container & );
    ~container();

    void set_type( char * );
    void set_volume( float volume ) { this->volume = volume; }
    void set_name( const char * );

    char *get_type() const { return type; }
    float get_volume() const { return volume; }
    char *get_name() const { return name; }

    int read_data( itstream * );
    void write_data( ostream * );
    void report( ostream * );
};

#endif

#ifndef _INC_container_dlg_
#define _INC_container_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   contdlg.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The container dialog provides the user interface to modify the list
//      of container types available for processing.
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)

```

INEL-96/0057

```
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
```

```
//
// Rev 1.0   05 Oct 1993 15:25:32   esm
// Initial release.
```

```
//
//
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
class container;
```

```
class container_dlg : public CDialog
```

```
{
    char *title;

    char *text;
    float volume;

    container *pcontainer;

    int sel;
    BOOL changed;          // Flag to indicate whether list has been changed.
    BOOL can_write;        // TRUE if underlying file is write enabled.
```

```
    CListBox *list_box;
```

```
    BOOL get_dlg_data( int );
    void set_dlg_data();
    BOOL is_duplicate();
    void set_button_state();
    void reset_focus( int );
```

```
    afx_msg void on_add();
    afx_msg void on_change();
    afx_msg void on_delete();
    afx_msg void on_selchange();
    afx_msg void OnPaint();
```

```
public:
```

```
    container_dlg();
    ~container_dlg();
```

```
    BOOL OnInitDialog();
    void OnOK();
    void OnCancel();
```

```
    DECLARE_MESSAGE_MAP();
```

```
};
```

```
#endif
```

```
#ifndef _INC_container_list_
#define _INC_container_list_
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//
```

```
// 1. Identification
```

```
//
```

```
// 1.1 Project
```

```
//
```

```
// Title: SWEPP Assay System
```

```
// Project ID: Task Baseline Agreement SDM-92-097
```

```
// Developed by: Scientific Computing Unit, INEL
```

[illegible]

```

1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
//
1.2 Software Module
//
// $Workfile: control.h $
// $Revision: 2.0 $
//
//
2. Function / Purpose
//
// The controller class models the generic functionality of a CAMAC crate
// controller.
//
//
3. Interface Description
//
//
//
4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:40 esm
// Initial release.
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#define MIN_STATION 1
#define MAX_STATION 25
#define MIN_ADDRESS 0
#define MAX_ADDRESS 15
#define MIN_FUNCTION 0
#define MAX_FUNCTION 31
#define NO_POWER -1
#define OFF_LINE 1

class controller
{
protected:
    unsigned char crate_number;
    unsigned int command_accepted; // CAMAC X
    unsigned int response; // CAMAC Q

public:
    controller() {}
    virtual ~controller() {}

    virtual int set_crate( unsigned char ) { return 0; }

    // Common control functions

    virtual void initialize() {}

```

```

virtual void clear() {}
virtual void set_inhibit() {}
virtual void clear_inhibit() {}
virtual int test_lam() { return 0; }

// CAMAC module I/O interface

virtual unsigned long read( unsigned char, unsigned char, unsigned char )
    { return 0L; }
virtual int write( unsigned char, unsigned char, unsigned char, long )
    { return 0; }
virtual void execute( unsigned char, unsigned char, unsigned char ) {}

unsigned int get_response()
    { return response; } // get last generated response (CAMAC Q)
unsigned int get_command_accepted() { return command_accepted; }

virtual int test() { return 1; }
};

#endif

#ifndef _INC_delay_dlg_
#define _INC_delay_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
//
// 1.2 Software Module
//
// delaydlg.h
//
// Written November 1995 by L. V. East, INEL
//
// 2. Function / Purpose
//
// Delays continuation of program for specified number of seconds.
// Displays modal dialog box showing number of seconds remaining.
// Intended to be used after an Active and before Passive assay.
//
// 3. Interface Description
//
// Derived from the Microsoft Foundation Class CDialog.
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 10 Nov 1995 lve
// Initial implementation - part of Version 2 modifications.
//
/////////////////////////////////////////////////////////////////

class delay_dlg : public CDialog
{

```

INEL-96/0057

```
private:
    int time_remaining;
    int m_remaining;    // minutes
    int s_remaining;    // seconds
    char buffer[32];
    char format[16];

    void time_cnvt();

    void OnCancel();
    BOOL OnInitDialog();

public:
    delay_dlg( int );
    ~delay_dlg();

    void update_time();
};

#endif // _INC_delay_dlg_

#ifndef _INC_detector_
#define _INC_detector_

////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    detector.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The detector class models the neutron detector in the assay chamber.
//
// 3. Interface Description
//
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:32    esm
//      Initial revision.
//
//
////////////////////////////////////////////////////////////////
```

```

#define DETECTOR    301

#include "device.h"
#include "errorbar.h"

class timer;
class gate;

class detector : public device
{
protected:
    error_bar rate;

public:
    timer **htimer;
    gate **hgate;

    detector( scaler_port *, timer ** );
    detector( scaler_port *, gate ** );
    detector( scaler_port *, timer **, gate **, int );
    virtual ~detector() {};

    virtual void compute_values();
    error_bar& get_rate() { return rate; }
};

#endif

#ifndef _inc_device_
#define _inc_device_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    device.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      This class, device, models a physical or logical device.  Derived
//      classes, such as gate, timer, detector, or coincidence detector, model
//      behavior specific for each device.  The device is associated with a
//      scaler port.
//
// 3. Interface Description
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request

```

INEL-96/0057

```
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
```

```
//
// Rev 1.0 05 Oct 1993 15:25:26 esm
//Initial revision.
```

```
//
//
//////////////////////////////////////////////////////////////////
```

```
class scaler_port;
```

```
class device
```

```
{
    char *name;
    int type;
```

```
protected:
    scaler_port *pscaler_port;
```

```
public:
```

```
    device( scaler_port *, int );
    device( scaler_port * );
    virtual ~device();

    void set_name( char * );
    char *get_name() { return name; }
    int is_type( int type ) { return this->type == type; }
```

```
    unsigned long get_count();
    virtual void compute_values() {};
```

```
};
```

```
#endif
```

```
#ifndef _INC_device_list_
#define _INC_device_list_
```

```
////////////////////////////////////////////////////////////////
```

```
//
// 1. Identification
```

```
//
// 1.1 Project
```

```
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
```

```
// 1.2 Software Module
```

```
//
// $Workfile: devlist.h $
// $Revision: 2.0 $
//
```

```
// 2. Function / Purpose
```

```
//
// The device list models the list of devices associated with
// a scaler port. In the usual case there is only a single
// device.
//
```

```
// 3. Interface Description
```

```
//
//
```

```

//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0   05 Oct 1993 15:25:40   esm
// Initial release.
//
//
//
/////////////////////////////////////////////////////////////////

#include "list.h"

class device;

class device_list : public list
{
public:
    device_list() {}
    ~device_list();

    void reset();

    device **find( char * );

    void append( device *pdev ) { list::append( (void *) pdev ); }
    void insert( device *pdev ) { list::insert( (void *) pdev ); }
    void remove( device *pdev ) { list::remove( (void *) pdev ); }
    void replace( device *, device * );

    device *get_first() { return (device *) list::get_first(); }
    device *get_next() { return (device *) list::get_next(); }

    device **get_hitem() { return (device **) list::get_hitem(); }
};

#endif

#ifdef _INC_dms_dlg_
#define _INC_dms_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile: dmsdlg.h $
// $Revision: 2.0 $
//
//
// 2. Function / Purpose

```



```

int dms_rev_flag;          // Determines how data are to be sent to/from DMS --
                           // will be either OLD_DMS_FORMAT or NEW_DMS_FORMAT
int num_recalc_bytes;      // Number of bytes in recalculation request from DMS
int num_assay_bytes;       // Number of bytes in assay request from DMS
int num_hs_bytes;          // Number of bytes in handshake between PC and DMS
int nbytes_read;           // Number of bytes received from DMS

BOOL request_active;

char *comm_attr;
CWnd *parent;
matrix *content;

unsigned int buffer_size;
char *buffer;

BOOL open_comm();
int close_comm() { return CloseComm( id_comm_dev ); }

void listen();

int receive_bytes( int );
BOOL receive_handshake();
void receive_new_assay_data();
void receive_recalc_data();

void send_handshake( BOOL );

void show_error( char * );
BOOL send_buffer();
void send_value( float );
void send_eb_value( error_bar );
void send_date( const time_t * );
void send_unknown();

void OnCancel() { ; }      // Ignore 'Cancel' requests.
BOOL OnInitDialog();

afx_msg void exit_remote();
afx_msg void send_results();
afx_msg void send_cancel();
afx_msg void OnDestroy();

public:
dms_dlg( char *, CWnd *, int = NEW_DMS_FORMAT );
~dms_dlg();

void set_content( matrix *content ) { this->content = content; }
void set_PAN_date( time_t datecode ) { PAN_date = datecode; }
void set_SGRS_date( time_t datecode ) { SGRS_date = datecode; }
void set_PAN_fileID( char *string ) { strcpy( PAN_fileID, string ); }
void set_SGRS_fileID( char *string ) { strcpy( SGRS_fileID, string ); }

unsigned int get_code() { return code; }
float get_weight() { return weight; }
char *get_file_id() { return file_id; }
char *get_volume_id() { return volume_id; }
int get_run_no() { return run_no; }
char *get_chamber_id() { return chamber_id; }
char *get_container_id() { return container_id; }

void receive_request();
void wait_for_send();

DECLARE_MESSAGE_MAP()
};

```

INEL-96/0057

#endif

#ifndef _INC_error_bar_
#define _INC_error_bar_

```
////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile: errorbar.h $
// $Revision: 2.0 $
//
// 2. Function / Purpose
//
// The error_bar class models an error bar, that is a value and its one
// standard deviation error.
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:28 esm
// Initial release.
//
// 5. References
//
// 5.1 References regarding error propagation and manipulation:
//
// 5.1.1 G. F. Knoll, "Radiation Detection and Measurement,"
// John Wiley and Sons, New York, NY, 1979, pp 131-135.
//
// 5.1.2 P. R. Bevington, "Data Reduction and Error Analysis for the
// Physical Sciences," McGraw Hill, New York, 1969, Chapter 4.
//
// 5.1.3 J. R. Taylor, "An Introduction to Error Analysis," Oxford
// University Press, 1982, Part I.
//
// 5.1.4 H. W. Coleman and W. G. Steele, "Experimentation and Uncertainty
// for Engineers," John Wiley & Sons, 1989, Section 3-1.
//
////////////////////////////////////

#include <fstream.h>
#include <iomanip.h>
```

```

#pragma warning( disable : 4270 )      // Keeps compiler from complaining about
                                        // generating temporary error_bar classes...
class error_bar
{
    float propagate( const error_bar &, const error_bar & );
    int is_zero( float );

public:
    float datum;
    float error;

    error_bar();
    error_bar( float );
    error_bar( float, float );
    ~error_bar() { };

    error_bar& operator=( const error_bar & );
    error_bar& operator=( float );

    friend error_bar operator+( const error_bar &, float );
    friend error_bar operator+( const error_bar &, const error_bar & );
    friend error_bar operator-( const error_bar &, float );
    friend error_bar operator-( const error_bar &, const error_bar & );
    friend error_bar operator*( const error_bar &, float );
    friend error_bar operator*( const error_bar &, const error_bar & );
    friend error_bar operator*( float arg, const error_bar & eb );
    friend error_bar operator/( const error_bar &, float );
    friend error_bar operator/( const error_bar &, const error_bar & );
    friend ostream& operator<<( ostream &, const error_bar & );
    friend istream& operator>>( istream &, error_bar & );

    error_bar& operator+=( float );
    error_bar& operator-=( float );
    error_bar& operator+=( const error_bar & );
    error_bar& operator-=( const error_bar & );
    error_bar& operator*=( float );
    error_bar& operator*=( const error_bar & );
    error_bar& operator/=( float );
    error_bar& operator/=( const error_bar & );

    int operator>( float );
    int operator<( float );
    int operator>=( float );
    int operator<=( float );
    int operator==( float );
    int operator!=( float );
    int operator==( const error_bar & );
    int operator!=( const error_bar & );
    int operator>( const error_bar & );
    int operator<( const error_bar & );
};

#endif

#ifndef _INC_file_lim_
#define _INC_file_lim_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL

```

INEL-96/0057

```
//      Funded by: Transuranic Waste Programs Unit, INEL
//
//
//      1.2 Software Module
//
//      $Workfile:   filelim.h  $
//      $Revision:   2.0  $
//
//
// 2. Function / Purpose
//
//      This file defines limits for files.
//
//
// 3. Interface Description
//
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      30 June 95   Modified during Version 2 development to use
//      _MAX_FNAME if defined./lve
//
//      Rev 1.0    05 Oct 1993 15:25:38   esm
//      Initial release.
//
////////////////////////////////////////////////////////////////
#include <stdlib.h>

#if defined( _MAX_FNAME )
#define FILE_NAME      _MAX_FNAME
#else
#define FILE_NAME      128
#endif

#define RECORD_LENGTH  128

#endif

#ifndef _INC_gate_
#define _INC_gate_

////////////////////////////////////////////////////////////////
//
// 1. Identification
//
//
//      1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
//
//      1.2 Software Module
//
//      $Workfile:   gate.h  $
//      $Revision:   2.0  $
```

```
//
// 2. Function / Purpose
//
// The gate class models the functionality of a gate, which is logical
// device. A gate is a window in time during which certain events are
// counted.
//
// 3. Interface Description
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0    05 Oct 1993 15:25:28    esm
// Initial release.
//
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#define GATE 101

#include "device.h"
#include "gateport.h"

class gate : public device
{
    gate_port *pgate_port;
    float time;
    float delay_time;

public:
    gate( scaler_port *, gate_port * );
    virtual ~gate() {};

    virtual void compute_values();

    float get_time() { return time; }
    float get_delay_time() { return delay_time; }
};

#endif

#ifndef _INC_gategen_
#define _INC_gategen_

////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
```

[illegible]

```
//      1.1 Project
//
//          Title: SWEPP Assay System
//          Project ID: Task Baseline Agreement SDM-92-097
//          Developed by: Scientific Computing Unit, INEL
//          Funded by: Transuranic Waste Programs Unit, INEL
//
//      1.2 Software Module
//
//          $Workfile:    gateport.h   $
//          $Revision:    2.0   $
//
//      2. Function / Purpose
//
//          The gate_port class models a port or address on a CAMAC gate generator.
//
//      3. Interface Description
//
//      4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:26    esm
//      Initial release.
//
//
///////////////////////////////////////////////////////////////////
#define NANOSEC_per_MICROSEC        1000L
#include "port.h"

class ostream;

class gate_port : public port
{
protected:
    unsigned long delay;
    unsigned long width;

    char *input_name;

public:
    gate_port();
    virtual ~gate_port();

    void set_input_name( char * );
    char *get_input_name() { return input_name; }

    virtual void set_delay( unsigned long delay ) { this->delay = delay; }
    virtual unsigned long get_delay() { return delay; }

    virtual void set_width( unsigned long width ) { this->width = width; }
    virtual unsigned long get_width() { return width; }

    virtual void write_data( ostream * ) {}
};
```

INEL-96/0057

```
#endif

#include <string.h>
#include <stdio.h> // for sprintf

class hstreambuf : public filebuf
{
    char *title;

public:
    hstreambuf( char * );
    virtual int sync();
    virtual int overflow( int ch );
    ~hstreambuf();

    void new_doc( char *title );
    void end_doc();

private:
    int column, line, page;
    char* buffer;
    void convert( long cnt );
    void newline( char*& pd, int& jj );
    void heading( char*& pd, int& jj );
    void pstring( char* ph, char*& pd, int& jj );
};

ostream &und( ostream &os );    // underline
ostream &reg( ostream &os );    // regular
ostream &pom( ostream &os );    // plus/minus character
ostream &formf( ostream &os );  // form feed character
ostream &tab( ostream &os );    // tab character
ostream &col( ostream &os );    // move column specified by next value in buffer

#ifndef _INC_itstream_
#define _INC_itstream_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:  itstream.h  $
//      $Revision:  2.0  $
//
// 2. Function / Purpose
//
//      The itstream class models an input token stream.
//
// 3. Interface Description
//
//
//
```

```

// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0   05 Oct 1993 15:25:24   esm
// Initial release.
//
//
//
/////////////////////////////////////////////////////////////////

#include <fstream.h>
#include "errorbar.h"
#include "fileelim.h"

#define NO_BUFFER 1

class itstream : public ifstream
{
    char buffer[RECORD_LENGTH];
    int section;
    streampos last_pos;

public:
    itstream( char *, int = 0 );

    void rewind( const char delim[] = " \t" );
    char* get_token( const char delim[] = " \t" );
    int find( const char *, ios::seek_dir dir = ios::cur,
              const char delim[] = " \t" );
    int find_section( const char *, ios::seek_dir dir = ios::cur,
                      const char delim[] = " \t" );
    void end_section();

    int get_val( const char *, unsigned char & );
    void get_val( unsigned char & );
    int get_val( const char *, int & );
    void get_val( int & );
    int get_val( const char *, unsigned int & );
    void get_val( unsigned int & );
    int get_val( const char *, long & );
    void get_val( long & );
    int get_val( const char *, unsigned long & );
    void get_val( unsigned long & );
    int get_val( const char *, float & );
    void get_val( float & );
    int get_val( const char *, error_bar & );
    void get_val( error_bar & );

    char *get_text( const char delim[] = " \t" );
};

#endif

#ifndef _INC_link_
#define _INC_link_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project

```

```
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
//
// 1.2 Software Module
//
//      $Workfile:    link.h  $
//      $Revision:    2.0   $
//
//
// 2. Function / Purpose
//
//      The link class models a link used in constructing a linked list.
//
//
// 3. Interface Description
//
//      Access to the item and the next link are through get and set
//      member functions.
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:22    esm
//      Initial release.
//
//
// //////////////////////////////////////
#include <stdlib.h>

class link
{
    void *item;
    link *next;

public:
    link( void *item = NULL, link *next = NULL )
    {
        this->item = item;
        this->next = next;
    }
    ~link() {}

    void *get_item() { return item; }
    link *get_next() { return next; }
    void set_item( void *an_item ) { item = an_item; }
    void set_next( link *a_link ) { next = a_link; }

    void **get_hitem() { return &item; }
};

#endif

#ifndef _INC_list_
#define _INC_list_
```

```

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
//
// 1.2 Software Module
//
//      $Workfile:    list.h  $
//      $Revision:    2.0  $
//
//
// 2. Function / Purpose
//
//      The list class models a generic singly linked list.
//
//
// 3. Interface Description
//
//      Functions are defined to append items to, insert items into,
//      replace items, or remove items from the list.
//      Access to the first, current, and last
//      links are through get functions.
//
//      The interface is protected, since only derived classes should
//      access these functions.
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:24    esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

#include "filelim.h"          // for FILE_NAME (max length of a file name).

class link;

class list
{
    link *first;
    link *last;
    link *current;

    int count;

    void initialize();
};

protected:
    char file_name[ FILE_NAME ];
    int has_write_permission;

```

```

    list();
    list( char * );
    ~list();

    void reset();

    void insert( void * );
    void append( void * );
    void remove( void * );
    void *replace( void * );

    void *get_first();
    void *get_next();
    void *get_current();
    void *get_last();

    void **get_hitem();

public:
    int can_write() { return has_write_permission; }
    virtual int read_data() { return 0; }
    virtual void write_data() {};

    int get_count() { return count; }
};

#endif

#ifndef _INC_login_dlg_
#define _INC_login_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:  logindlg.h  $
//      $Revision:  2.0  $
//
// 2. Function / Purpose
//
//      The login dialog provides the user interface to log in to the system.
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.

```

```

//
// Rev 1.0   05 Oct 1993 15:25:28   esm
// Initial release.
//
//
////////////////////////////////////////////////////////////////

class user;

class login_dlg : public CDialog
{
    user *login_user;

public:
    login_dlg( user * );
    ~login_dlg();

    void check_progress();
    BOOL OnInitDialog();
    void OnOK();
};

#endif

#ifndef _INC_main_wnd
#define _INC_main_wnd_

////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   mainwnd.h   $
//      $Revision:   2.0   $
//
// 2. Function / Purpose
//
//      The main window class provides the overall management of the user
//      interface and the operating system interface.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
// Rev 1.0   05 Oct 1993 15:25:34   esm
// Initial release.

```

```

//
//
/////////////////////////////////////////////////////////////////

#include <afxext.h>          // For status bar object definitions.

class user;
class matrix;
class dms_dlg;
class itstream;

class main_wnd : public CMDIFrameWnd
{
    BOOL is_initialized;
    BOOL is_exiting;
    BOOL login_ok;           // Will be TRUE if user login successful.
    BOOL set_child_position;

    CString str_profile_heading;

    static const char *wait_message;

    itstream *as_ini;
    char *comm_attr;
    dms_dlg *pdms_dlg;
    CMenu *pmenu_init;
    CStatusBar status_bar;

    char *root_dir;
    int root_drive;
    char *log_file_name;
    char *err_file_name;

    //
    void read_chamber_data();

    // utilities
    int new_chamber( itstream * );
    char *set_file_path( char *, char *, char *default_name = NULL );
    void set_menu();
    void restore_window();

    // menu activated functions
    afx_msg void on_new_log();
    afx_msg void on_print_log();
    afx_msg void on_print();
    afx_msg void on_users();
    afx_msg void on_exit();
    afx_msg void on_container();
    afx_msg void on_matrix();
    afx_msg void on_remote();
    afx_msg void end_remote();
    afx_msg void on_about();
    afx_msg void on_login();
    afx_msg void on_logout();

    // system message activated functions
    afx_msg int OnCreate( LPCTSTR );
    afx_msg void OnClose() { on_exit(); }
    afx_msg void OnDestroy();
    afx_msg void OnSize( UINT, int, int );

    // override functions
    void MDIIconArrange();

public:
    main_wnd();

```

```

~main_wnd();

user *login_user;

int read_data();
void do_assay();
void do_recalc();
BOOL check_printer();
void no_resources( unsigned long arg = 0L );
void set_window( int );

afx_msg void OnUpdateStatus( CCmdUI* );// Private, not system...

DECLARE_MESSAGE_MAP()
};

#endif

#ifndef _INC_matrix_parameter_dlg_
#define _INC_matrix_parameter_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:  mapardlg.h  $
//      $Revision:  2.0  $
//
// 2. Function / Purpose
//
//      The matrix parameters dialog provides the user interface to modify
//      selected parameters for the matrix class.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:22   esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

#include "errorbar.h"

class matrix;

```

```

class matrix_parameter_dlg : public CDialog
{
    matrix *pmatrix;
    BOOL can_change;

    int sludge;
    int Am_sludge;

    CButton *sludge_button;
    CButton *Am_sludge_button;
    CButton *non_sludge_button;

    void toggle( int &, CButton * );
    void set_error_bar( error_bar, int, int );
    error_bar get_error_bar( int, int );

    afx_msg void on_button_select();
    afx_msg void on_sludge();
    afx_msg void on_Am_sludge();
    afx_msg void on_non_sludge();
    afx_msg void OnPaint();

public:
    matrix_parameter_dlg( matrix *, BOOL );
    ~matrix_parameter_dlg() {};

    BOOL OnInitDialog();
    void OnOK();

    DECLARE_MESSAGE_MAP()
};

#endif

#ifndef _INC_matrix_dlg_
#define _INC_matrix_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   matrdlg.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The matrix dialog class provides the user interface to modify the list
//      of matrices processed at the assay station.
//
// 3. Interface Description
//
//
//

```

```

// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0   05 Oct 1993 15:25:26   esm
// Initial release.
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
class matrix;
class main_wnd;

class matrix_dlg : public CDialog
{
    char *title;
    main_wnd *parent;

    char *text;
    unsigned int content_code;

    matrix *pmatrix;

    int sel;
    BOOL changed;          // Flag to indicate whether list has been changed
    BOOL can_write;        // TRUE if underlying matrix file is write enabled.

    CListBox *list_box;

    BOOL get_dlg_data( int );
    void set_dlg_data();
    BOOL is_duplicate();
    void reset_focus( int );
    int add_string();
    void set_button_state();
    void set_matrix_parameters();

    afx_msg void on_add();
    afx_msg void on_change();
    afx_msg void on_delete();
    afx_msg void on_selchange();
    afx_msg void OnPaint();

public:
    matrix_dlg( main_wnd * );
    ~matrix_dlg();

    BOOL OnInitDialog();
    void OnOK();
    void OnCancel();

    DECLARE_MESSAGE_MAP()
};

#endif

#ifdef _INC_matrix_
//
// 1. Identification

```

```
//      1.1 Project
//
//          Title: SWEPP Assay System
//          Project ID: Task Baseline Agreement SDM-92-097
//          Developed by: Scientific Computing Unit, INEL
//          Funded by: Transuranic Waste Programs Unit, INEL
//
//      1.2 Software Module
//
//          $Workfile:   matrix.h   $
//          $Revision:   2.0    $
//
//
//  2. Function / Purpose
//
//      The matrix class models the waste content of a container. Active and
//      passive signals are used to estimate the amount of Plutonium, Uranium and
//      Americium in the waste. Other derived attributes are computed based on
//      the masses of Pu, U and Am.
//
//
//  3. Interface Description
//
//
//  4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program. Extensive modifications for V2.0,
//
//      Rev 1.1    24 Jan 1994 15:25:38     esm
//      Added code to test for high/low count rates. SPR #015
//
//      Rev 1.0    05 Oct 1993 15:25:20     esm
//      Initial release.
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#define PU238  0
#define PU239  PU238 + 1
#define PU240  PU239 + 1
#define PU241  PU240 + 1
#define PU242  PU241 + 1
#define NUM_PU_ISOTOPES PU242 + 1

#define PASSIVE_SG 1
#define PASSIVE_LG 2
#define PASSIVE_SRC 3

// 'gamma_flag' can be logical OR of (at least some of) the following values.
#define FILE_FOUND         1
#define FILE_NOT_FOUND    2
#define DEFAULTS_USED     4
#define GAMMA_DATA_USED   8
#define SAVED_DATA_USED   16
#define OUTSIDE_NOMINAL   32

#include <iostream.h>
#include <math.h>
#include "errorbar.h"
```

```

#include "nuclide.h"

class chamber;
class container;
class error_bar;
class nuclide;
class itstream;

class matrix
{
friend class matrix_dlg;           // allow matrix_dlg to set attributes
friend class matrix_parameter_dlg; // allow matrix_parameter_dlg to set attributes
friend class results_dlg;         // allow results_dlg to retrieve summary values
friend class dms_dlg;             // allow dms_dlg to retrieve computed values
friend class chamber_wnd;         // allow chamber_wnd to access nuclides

public:
    static enum Criteria           // (gets referenced in private section...)
    {
        Default,
        Active,
        Passive
    };

private:
    unsigned int code;
    char *type;

    float moderator_index;
    float active_self_absorption;

    Criteria mass_selection_criteria;

    int gamma_flag;                // Gives status of getting mass fraction info from
gamma file.
    int passive_type;              // Will be set to PASSIVE_SG, PASSIVE_LG, or PASSIVE_SRC.

    BOOL from_active;
    BOOL bad_passive_data;

    // self absorption correction factor correlation coefficients
    float c0;
    float c1;
    float c2;
    float c3;
    float div;

    // externally supplied correction factors
    error_bar active_cf;
    error_bar passive_shielded_cf;
    error_bar passive_system_cf;

    float weight;
    chamber *pchamber;

    BOOL sludge;
    BOOL Am_sludge;

    // primary derived attributes
    nuclide Pu[NUM_PU_ISOTOPES];    // 'nuclide' defined in nuclide.h
    nuclide U233;
    nuclide U235;
    nuclide Am241;
    error_bar total_Pu_mass;

    // secondary derived attributes
    error_bar power;

```

```

error_bar power_density;
error_bar alpha_activity;
error_bar Pu_equiv_activity;
error_bar total_activity;
error_bar TRU_activity;
error_bar TRU_activity_concentration;
error_bar fissile_gram_equivalent;

// internally used, derived attributes
float absorber_index;
error_bar active_Pu_mass;
error_bar passive_Pu_mass;
error_bar Am241_NE_mass; // Am-241 mass derived from "neutron excess"

error_bar U_cf;

error_bar active_shielded_total_rate;
error_bar active_flux_monitor_rate;
error_bar active_barrel_flux_monitor_rate;

error_bar passive_shielded_total_rate;
error_bar passive_system_total_rate;
error_bar passive_shielded_coincidence_rate;
error_bar passive_system_coincidence_rate;

BOOL rate_limited;

// constants for absorber index equation
float AI_DEFAULT;
float AI_MIN;

// constants for moderator index equation
float A0, A1, A2, A3;
float MI_MIN;
float MI_MAX;

// active matrix correction factor parameters
float AI_THRESHOLD;
float AMCF_AI_C0;
float AMCF_AI_C1;
float MI_THRESHOLD;
float AMCF_MI_C0;
float AMCF_MI_C1;

// mass calculation coefficients for active calculation
float AI_MI_C0;
float AI_MI_C1;
float ACTIVE_BACKGROUND_SCALE_FACTOR;
float ACTIVE_MASS_ERR_C0;
float ACTIVE_MASS_ERR_C1;
float ACTIVE_MASS_ERR_C2;

// constants for passive mass calculations
float SYSTEM_COINCIDENCE_CF_C0;
float SYSTEM_COINCIDENCE_CF_C1;
float SHIELDED_COINCIDENCE_CF_C0;
float SHIELDED_COINCIDENCE_CF_C1;

// final mass calc paramters
float MASS_THRESHOLD;

// americium sludge mass equation coefficients
float AM_SLUDGE_MASS_C0;
float AM_SLUDGE_MASS_C1;
float AM_SLUDGE_MASS_C2;
float AM_SLUDGE_MASS_C3;

```

```

// internal methods
void initialize();

BOOL flux_rates_ok()
{
    return (active_barrel_flux_monitor_rate > 0.0f &&
           active_flux_monitor_rate > 10.0f);
}

void calc_absorber_index();
void calc_moderator_index();
float calc_active_matrix_cf();

error_bar calc_active_Pu_mass(); // calc active mass
error_bar calc_passive_Pu_mass(); // calc passive mass
error_bar PAN_passive_Pu_mass( const char * ); // does the calculations
void calc_U_mass(); // calc U-23x masses based on gamma data
error_bar calc_Am_mass( BOOL = FALSE ); // calc Am-241 mass, not Am sludge
error_bar calc_sludge_Am_mass(); // calc Am-241 mass if sludge is Am laced
int Am_mass_from_ratios(); // Am-241 mass from isotopic ratio data

float LANL_error_estimate( float matrix_cf, float active_mass ) // New in V2
{
    return (float)fabs( ACTIVE_MASS_ERR_C0 + ( ACTIVE_MASS_ERR_C1 +
        ACTIVE_MASS_ERR_C2 * matrix_cf ) * active_mass );
}

void calc_power();
void calc_activity();
void calc_FGE();

void print_error_bar( ostream *, error_bar );

public:
matrix();
matrix( unsigned int, char * );
~matrix();

char Pu_mass_note[64]; // String describing how total Pu mass was derived.
char Am_mass_note[64]; // String describing how Am-241 mass was derived.
char Isotopic_note[64]; // String describing how isotopic masses determined
char U_corr_note_1[48]; // Strings describing U correction to active mass.
char U_corr_note_2[48];

container *pcontainer;

void set_chamber( chamber *pchamber ) { this->pchamber = pchamber; }
void set_gamma_flag( int arg ) { gamma_flag = arg; }
void set_weight( float weight ) { this->weight = weight; }
float get_weight() { return weight; }

BOOL is_rate_limited() { return rate_limited; }
BOOL is_active_ok();
BOOL is_passive_ok();
BOOL is_from_active() { return from_active; }
BOOL is_from_passive() { return !from_active; }

BOOL read_data( itstream * );
BOOL read_correlation_data( itstream * );
BOOL read_nuclide_data( itstream * );
BOOL read_Pu_fractions( itstream * );
void read_mass_ratios( itstream * );

BOOL valid_mass( error_bar &value )
{ return (value.datum != -1.0f && value.error != -1.0f); }

void write_data( ostream * );

```

```

void write_Pu_fractions( ostream * );
void write_values( ostream *, BOOL, BOOL, BOOL );

void reset_nuclide_data( nuclide * );

void set_type( char * );
void set_code( unsigned int code ) { this->code = code; }
unsigned int get_code() { return code; }
char *get_type(){ return type; }
float self_absorption_cf( float, float );

void copy( const matrix & );
void assay();
unsigned long get_active_pulse_count();
void report( ostream * );
void set_passive_bad() { bad_passive_data = TRUE; }
};

#endif

#ifndef _INC_matrix_list_
#define _INC_matrix_list_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile:  matrlist.h  $
// $Revision:  2.0  $
//
// 2. Function / Purpose
//
// The matrix list object models the list of matrixes that are
// processed at the assay station. It is derived from the "list"
// class (see list.h).
//
// 3. Interface Description
//
// Functions are defined for appending, inserting, and
// removing matrixes. A find function searches the list for
// a matrix with a specified code.
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:38 esm
// Initial release.

```

```

//
//
////////////////////////////////////////////////////////////////

#include "list.h"

class matrix;

class matrix_list : public list
{
    void reset();

public:
    matrix_list( char * );
    ~matrix_list();

    virtual int read_data();
    virtual void write_data();

    void append( matrix *pm ) { list::append( (void *) pm ); }
    void insert( matrix *pm ) { list::insert( (void *) pm ); }
    void remove( matrix *pm ) { list::remove( (void *) pm ); }

    matrix *find( unsigned int );

    matrix *get_first() { return (matrix *) list::get_first(); }
    matrix *get_next() { return (matrix *) list::get_next(); }

};

#endif

// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0    05 Oct 1993 15:25:24    esm
// Initial release.
//
//
////////////////////////////////////////////////////////////////

// Define the menu bar position for each "Window" menu.
// See assaysys.rc for complete menu declarations.
#define OPER_WINDOW_POS          3
#define PHYS_WINDOW_POS          3
// No "Window" menu item in the Initial and Administrator menus, so
// set to invalid position. This will result in a NULL pointer being
// used in calls to CreateClient() and MDISetMenu(), but this is OK.
#define ADMIN_WINDOW_POS         99
#define INIT_WINDOW_POS          999

// Menu bar

#define UI_LOGIN                  10

// File Menu

#define UI_NEW                    110
#define UI_OPEN                  120
#define UI_SAVE                  130
#define UI_CLOSE                 140
#define UI_PRINT                 150
#define UI_USERS                 160

```

INEL-96/0057

```
#define UI_NEW_LOG          170
#define UI_PRINT_LOG        180
#define UI_EXIT              190

#define UI_LOGOUT            20

// Run Menu

#define UI_ASSAY             210
#define UI_ACTIVE            220
#define UI_PASSIVE           230
#define UI_BACKGROUND        240
#define UI_VERIFICATION      250
#define UI_CHI_SQUARE        260
#define UI_REMOTE            270
#define UI_REGRESSION        280
#define UI_TEST_CRATE        290

// Options Menu

#define UI_SUMMARY_REPORT    310
#define UI_BACKGROUND_CORRECTION 320
#define UI_SET_COLOR         330
#define UI_SET_DEFAULTS      340
#define UI_CALIBRATION       350
#define UI_CONTAINER         360
#define UI_MATRIX            370

// Window Menu

#define UI_CASCADE           410
#define UI_TILE              420
#define UI_NEXT              430
#define UI_ARRANGE           440

// Help Menu

#define UI_ABOUT             990

// Non-menu commands

#define DO_ASSAY             9010
#define DO_RECALC            9020
#define END_REMOTE           9030
#define PRINT_REPORT         9040
#define CONTINUE             9050
#define ACCEPT_BKGND         9060
#define REJECT_BKGND         9070

#ifndef _INC_message_dlg_
#define _INC_message_dlg_

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
//
// 1.2 Software Module
//
// msgdlg.h
//
// Written October 1995 by L. V. East as part of V2.0 modifications.
```

```
//
// 2. Function / Purpose
//
//      Dialog box to display a message to user.  Intended for use when bad
//      data or other problem is encountered and allow user to choose to
//      about current operation.
//
// 3. Interface Description
//
//      Derived from the Microsoft Foundation Class CDialog.
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
```

```
// Rev 1.0   13 Oct 1995 lve
//      Initial implementation.
//
```

```
////////////////////////////////////
```

```
class message_dlg : public CDialog
{
private:
    char *title;
    char *message;

public:
    message_dlg( char *, char * );
    ~message_dlg();

    BOOL OnInitDialog();

    afx_msg void OnDestroy();

    DECLARE_MESSAGE_MAP()
};

#endif // _INC_message_dlg_
```

```
#ifndef _INC_nuclide_
#define _INC_nuclide_
```

```
////////////////////////////////////
```

```
//
// 1. Identification
//
//      1.1 Project
//
//          Title: SWEPP Assay System
//          Project ID: Task Baseline Agreement SDM-92-097
//          Developed by: Scientific Computing Unit, INEL
//          Funded by: Transuranic Waste Programs Unit, INEL
//
//      1.2 Software Module
//
//          $Workfile:   nuclide.h  $
//          $Revision:   2.0  $
//
```

```

//
// 2. Function / Purpose
//
// The nuclide class encapsulates the nuclide data required for assay analysis
// It also includes the calculated nuclide mass (value +/- uncertainty).
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Version 2 modifications -- December 1995 lve
// Class members "mass_ratio", "aux_ratio" "equivalent_Pu_factor",
// "fissile_gram_equivalent", "x_section" and "nu_bar" added.
// Removed unused constructors.
// All "float" member variables initialized to zero in constructor.
//
// Rev 1.0 05 Oct 1993 15:25:40 esm
// Initial release.
//
//
//
////////////////////////////////////////////////////////////////
#include "errorbar.h"

class nuclide
{
public:
    error_bar mass; // Calculated value.
    error_bar mass_fraction; // 0.0 to 1.0
    error_bar mass_ratio; // Mass ratio relative to Pu-239 (usually...)
    error_bar aux_ratio; // Mass ratio relative to something else
    float alpha_branching_ratio; // 0.0 to 1.0
    float equivalent_Pu_factor; // Weighting factor for Pu equivalent act.
    float fissile_gram_equivalent; // Positive value; relative to 1.0 for Pu-239
    float specific_activity; // Total Curies per gram (Ci/g)
    float thermal_power; // Watts per gram (Watts/g)
    float x_section; // (Effective) fission cross-section (barns)
    float nu_bar; // Average number of neutrons/fission

// Class constructor(s)

    nuclide() { initialize(); }
    // The following constructors are not currently used
    // nuclide( float d, float e) { mass.datum = d; mass.error = e; initialize(); }
    // nuclide( error_bar eb ) { mass = eb; initialize(); }

// Class destructor

    ~nuclide() {}

private:

// Initializes member float's to 0; error_bar's get initialized on construction.

    void initialize()
    { alpha_branching_ratio = equivalent_Pu_factor = fissile_gram_equivalent =

```

```

        specific_activity = thermal_power = x_section = nu_bar = 0.0f; }
};

#endif

#ifndef _INC_passive_dlg_
#define _INC_passive_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   pasdlg.h   $
//      $Revision:   2.0   $
//
// 2. Function / Purpose
//
//      The passive dialog provides the user interface to view the
//      counts and rates from a passive data acquisition.
//
// 3. Interface Description
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:36   esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

#define MAX_PASSIVE_BADVALUES    24

class passive_signals;
class error_bar;
class chamber_wnd;
class matrix;

class passive_dlg : public CDialog
{
    chamber_wnd *parent;

    matrix *content;
    passive_signals *passive;

```

INEL-96/0057

```
    BOOL background_run;          // Will be TRUE only for passive background runs.

    char buffer[ RECORD_LENGTH ];

    HBRUSH hbrBkgnd;              // Background brush to use for highlighted message boxes.
    COLORREF bkgColor;            // Color to use for background brush.

    BOOL bad_counts;
    BOOL bad_rate;
    int bad_valueIDs[ MAX_PASSIVE_BADVALUES ];
    int bad_values;
    float fuzzy;

    unsigned long set_count( char *, int );
    unsigned long set_count( unsigned long, int );
    void set_rate( error_bar, int, int );

    afx_msg HBRUSH OnCtlColor( CDC*, CWnd*, UINT );
    afx_msg void OnClose();
    afx_msg void OnLButtonDown( UINT, CPoint );
    afx_msg void OnPaint();
    afx_msg void on_accept();
    afx_msg void on_reject();

public:
    passive_dlg( passive_signals *, chamber_wnd *,
                matrix *content = NULL, BOOL bkgnd_run = FALSE );
    ~passive_dlg();

    void rate_check();
    void enable_buttons();

    BOOL OnInitDialog();

    DECLARE_MESSAGE_MAP()
};

#endif

#ifdef _INC_passive_signals_
////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   passgnls.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The passive signals provides the interface to the passive data to
//      access the shielded total, the system total, the shielded coincidence,
//      and the system coincidence signals.
//
```

```

//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.1   24 Jan 1994 15:27:20   esm
// Added code to check high/low count rates. SPR #015
//
// Rev 1.0   05 Oct 1993 15:25:30   esm
// Initial release.
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#include <stdlib.h>
#include "signals.h"
#include "camcrate.h"

class error_bar;

class passive_signals : public signals
{
    unsigned int background_correction : 1;

    unsigned long get_count( char *name, char *gate_name = NULL )
    {
        return background_correction ?
            get_gross_count( name, gate_name ) - get_background_count( name,
                gate_name ) :
            get_gross_count( name, gate_name );
    }

    error_bar get_rate( char *name, char *gate_name = NULL );
    error_bar get_rate_ratio( char *name, char *gate_name );
public:
    passive_signals( camac_crate *gross, camac_crate *background ) : signals(
gross, background )
    { background_correction = 1; }
    virtual ~passive_signals() { }

    void set_background_correction( unsigned int background_correction )
    { this->background_correction = background_correction; }

    unsigned int get_background_correction()
    { return this->background_correction; }

    unsigned long get_shielded_total_count()
    { return get_count( "SHIELDED_TOTAL" ); }
    unsigned long get_system_total_count()
    { return get_count( "SYSTEM_TOTAL" ); }

    unsigned long get_shielded_coincidence_count()
    { return get_count( "SHIELDED_TOTAL", "SHORT_GATE" ); }
    unsigned long get_system_coincidence_count()
    { return get_count( "SYSTEM_TOTAL", "LONG_GATE" ); }

    error_bar get_shielded_total_rate()

```

```

        { return get_rate( "SHIELDED_TOTAL" ); }
error_bar get_system_total_rate()
        { return get_rate( "SYSTEM_TOTAL" ); }

error_bar get_shielded_coincidence_rate()
        { return get_rate( "SHIELDED_TOTAL", "SHORT_GATE" ); }
error_bar get_system_coincidence_rate()
        { return get_rate( "SYSTEM_TOTAL", "LONG_GATE" ); }

error_bar get_shielded_rate_ratio()
        { return get_rate_ratio( "SHIELDED_TOTAL", "SHORT_GATE" ); }
error_bar get_system_rate_ratio()
        { return get_rate_ratio( "SYSTEM_TOTAL", "LONG_GATE" ); }
};

#endif

#ifndef _INC_passive_signals_
#define _INC_passive_signals_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:  passgnls.h  $
//      $Revision:  2.0  $
//
// 2. Function / Purpose
//
//      The passive signals provides the interface to the passive data to
//      access the shielded total, the system total, the shielded coincidence,
//      and the system coincidence signals.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.1   24 Jan 1994 15:27:20   esm
//      Added code to check high/low count rates. SPR #015
//
//      Rev 1.0   05 Oct 1993 15:25:30   esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

```

```

#include <stdlib.h>
#include "signals.h"
#include "camcrate.h"

class error_bar;

class passive_signals : public signals
{
    unsigned int background_correction : 1;

    unsigned long get_count( char *name, char *gate_name = NULL )
    {
        return background_correction ?
            get_gross_count( name, gate_name ) - get_background_count( name,
                gate_name ) :
            get_gross_count( name, gate_name );
    }

    error_bar get_rate( char *name, char *gate_name = NULL );
    error_bar get_rate_ratio( char *name, char *gate_name );

public:
    passive_signals( camac_crate *gross, camac_crate *background ) :
        signals( gross, background )
        { background_correction = 1; }
    virtual ~passive_signals() { }

    void set_background_correction( unsigned int background_correction )
        { this->background_correction = background_correction; }

    unsigned int get_background_correction()
        { return this->background_correction; }

    unsigned long get_shielded_total_count()
        { return get_count( "SHIELDED_TOTAL" ); }
    unsigned long get_system_total_count()
        { return get_count( "SYSTEM_TOTAL" ); }

    unsigned long get_shielded_coincidence_count()
        { return get_count( "SHIELDED_TOTAL", "SHORT_GATE" ); }
    unsigned long get_system_coincidence_count()
        { return get_count( "SYSTEM_TOTAL", "LONG_GATE" ); }

    error_bar get_shielded_total_rate()
        { return get_rate( "SHIELDED_TOTAL" ); }
    error_bar get_system_total_rate()
        { return get_rate( "SYSTEM_TOTAL" ); }

    error_bar get_shielded_coincidence_rate()
        { return get_rate( "SHIELDED_TOTAL", "SHORT_GATE" ); }
    error_bar get_system_coincidence_rate()
        { return get_rate( "SYSTEM_TOTAL", "LONG_GATE" ); }

    error_bar get_shielded_rate_ratio()
        { return get_rate_ratio( "SHIELDED_TOTAL", "SHORT_GATE" ); }
    error_bar get_system_rate_ratio()
        { return get_rate_ratio( "SYSTEM_TOTAL", "LONG_GATE" ); }
};

#endif

#ifdef _inc_port_
////////////////////////////////////

```

```
// 1. Identification
//
//      1.1 Project
//
//          Title: SWEPP Assay System
//          Project ID: Task Baseline Agreement SDM-92-097
//          Developed by: Scientific Computing Unit, INEL
//          Funded by: Transuranic Waste Programs Unit, INEL
//
//      1.2 Software Module
//
//          $Workfile:    port.h  $
//          $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The port class models a port or address on a CAMAC board. Ports
//      are named in association with the internal or external device
//      signals wired into the CAMAC boards.
//
// 3. Interface Description
//
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:38   esm
//      Initial release.
//
////////////////////////////////////
class ostream;

class port
{
protected:
    char *name;

public:
    port();
    virtual ~port();

    virtual void set_name( char * );
    char *get_name(){ return name; }
    unsigned char is_name( char * );

    virtual void write_data( ostream * ) {;}
};

#endif

#ifdef _inc_port_2323_
#define inc port 2323
```

```

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile: port2323.h $
// $Revision: 2.0 $
//
// 2. Function / Purpose
//
// The port_2323 class models the ports on a LeCroy 2323 Dual Gate and
// Delay Generator.
//
// 3. Interface Description
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:36 esm
// Initial release.
//
// 5. References
//
// 5.1 Operator's Manual, CAMAC Model 2323A, Dual Gate and Delay Generator
// LeCroy, May 1987
//
/////////////////////////////////////////////////////////////////

// The programming of each port is managed through a 16 bit word that is
// partitioned into 2 bits for delay, 1 bit for latch, 3 bits for exponent,
// and 10 bits for the mantissa. The internal delay may be set for 10ns, 30ns,
// 100ns, or 300ns. The latch determines the mode, whether internal
// or external. The exponent and mantissa define the gate width.

#define DELAY_10ns 0x00 // delay bits - defines delay width
#define DELAY_30ns 0x01
#define DELAY_100ns 0x02
#define DELAY_300ns 0x03
#define MIN_DELAY DELAY_10ns
#define MAX_DELAY DELAY_300ns

#define INTERNAL 0x00 // latch bit = 0; gate duration defined by control word
#define EXTERNAL 0x01 // latch bit = 1; gate duration defined by START and STOP

#define MIN_EXPONENT 0x00
#define MAX_EXPONENT 0x07 // 3 bits wide

```

INEL-96/0057

```
#define MIN_MANTISSA 0x00
#define MAX_MANTISSA 0x3ff // 10 bits wide

#include "gateport.h"

class port_2323 : public gate_port
{
    unsigned char internal_delay;
    unsigned char mode;
    unsigned char exponent;
    unsigned int mantissa;

public:
    port_2323();
    virtual ~port_2323() {}

    void set_internal_delay( unsigned char );
    void set_mode( unsigned char );
    virtual void set_width( unsigned long );

    int get_data();
    void set_data( int ); // inactive, untested

    unsigned int get_internal_delay(); // inactive, untested
    unsigned char get_mode() { return mode; }
    virtual unsigned long get_width();

    virtual void write_data( ostream * );
};

#endif

#ifndef _INC_propt_dlg_
#define _INC_propt_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   proptdlg.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The process options dialog class provides the user interface to
//      modify the acquisition parameters for background, verification,
//      and production processing.
//
// 3. Interface Description
//
//
// 4. History
//
```

```

// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0   05 Oct 1993 15:25:24   esm
// Initial release.
//
//
/////////////////////////////////////////////////////////////////

class chamber_wnd;

class process_options_dlg : public CDialog
{
    chamber_wnd *pchamber_wnd;

    void set_dlg_data();
    afx_msg void on_defaults();

public:
    process_options_dlg( chamber_wnd * );
    ~process_options_dlg() {}

    BOOL OnInitDialog();
    void OnOK();

    DECLARE_MESSAGE_MAP();
};

#endif

#ifndef _INC_range_check
#define _INC_range_check_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   rangechk.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      This file provides the macro to perform a range check. (DEBUG use only)
//
// 3. Interface Description
//
//
//
// 4. History
//

```

INEL-96/0057

```
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
//      30 June 95      Changed to use "ASSERT" (MFC macro) rather than "assert". /lve
//
// Rev 1.0   05 Oct 1993 15:25:22   esm
// Initial release.
//
//
/////////////////////////////////////////////////////////////////

#include <afx.h>

#define RANGE_CHECK( min, max, value )  ASSERT( min <= value && value <= max )

#endif

#ifndef _INC_results_dlg_
#define _INC_results_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
//      1.1 Project
//
//          Title: SWEPP Assay System
//          Project ID: Task Baseline Agreement SDM-92-097
//          Developed by: Scientific Computing Unit, INEL
//          Funded by: Transuranic Waste Programs Unit, INEL
//
//      1.2 Software Module
//
//          $Workfile:   resltdlg.h  $
//          $Revision:   2.0  $
//
// 2. Function / Purpose
//
//      The results dialog provides the user interface to view the results
//      of the assay calculations.
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0   05 Oct 1993 15:25:36   esm
// Initial release.
//
//
/////////////////////////////////////////////////////////////////
```

```

#define MAX_RESULT_BADVALUES      4

class matrix;
class error_bar;
class chamber_wnd;

class results_dlg : public CDialog
{
    matrix *content;
    chamber_wnd *parent;

    BOOL bad_mass;

    char *insert_eb( char *, error_bar ); // Inserts error_bar value into string.

    afx_msg void OnClose();
    afx_msg void OnPaint();
    afx_msg void OnLButtonDown( UINT, CPoint );

public:
    results_dlg( matrix *, chamber_wnd * );
    ~results_dlg();

    void disable_print();
    void enable_save();

    void mass_check();

    afx_msg void on_print_report();
    afx_msg void on_save();
    afx_msg void on_close();

    BOOL OnInitDialog();

    DECLARE_MESSAGE_MAP()
};

#endif

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    resource.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      This file defines (most of) the resources for the dialogs
//      See also menu.h and resrc1.h.
//
// 3. Interface Description
//

```

INEL-96/0057

```
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0    05 Oct 1993 15:25:38    esm
// Initial release.
//
//
//
//
// NOTE: Resources defined by AppStudio are in resrc1.h

#define Unused                -1

// User Dialog Boxes
#define ID_USER_NAME          101
#define IDD_LOGIN             300
#define ID_USER_ID            301
#define ID_USER_PW            302
#define IDB_OPERATOR          310
#define IDB_PHYSICIST         311
#define IDB_ADMINISTRATOR     312

// Container Attributes Dialog
#define ID_TEXT_ID            1110
#define ID_CONTENT_CODE       1120
#define ID_WEIGHT             1130

// Set Defaults Dialog
#define ID_NUM_PULSES         1230
#define ID_NUM_SECS           1240

// Acquire Dialog
#define IDD_ACQUIRE           1500
#define IDT_REQUESTED         1511
#define ID_REQUESTED          1512
#define IDT_ESTIMATED         1521
#define ID_ESTIMATED          1522
#define ID_STATUS_BAR         1530

#define IDD_REMOTE             400
#define IDD_CALIBRATION       600
#define IDT_BUILD_DATE        702
#define IDB_DEFAULT           705
#define IDB_ACTIVE            706
#define IDB_PASSIVE           707
#define IDD_ACTIVE            800
#define IDD_PASSIVE           900
#define IDD_RESULTS           1000
#define IDD_OPTIONS           1200
#define ID_VER_PULSES         1250
#define ID_VER_SECS           1260
#define ID_SHIELDED           604
#define ID_SHIELDED_ERROR     605
#define ID_ACTIVE             703
#define ID_DESCRIPTION         704
#define ID_ACTIVE_ERROR       708
#define ID_C0                 710
#define ID_C1                 711
#define ID_C2                 712
```

```

#define ID_C3 713
#define ID_BKG_SECS 1204
#define IDD_ABOUT 1300
#define ID_SHIELDED_COINCIDENCE 608
#define ID_SHIELDED_COINCIDENCE_ERROR 609
#define ID_SYSTEM_COINCIDENCE 610
#define ID_SYSTEM_COINCIDENCE_ERROR 611
#define ID_NAME 1401
#define ID_VOLUME 1403
#define IDL_CONTAINERS 1405
#define IDB_ADD 1406
#define IDB_CHANGE 1407
#define IDB_DELETE 1408
#define IDL_MATRIX 715
#define IDD_MATRIX_PARAMETERS 1600
#define ID_MI 1601
#define ID_DIV 1602
#define IDD_MATRIX_LIST 700
#define IDD_CONTAINER_LIST 1400
#define IDD_WASTE_ID 1100
#define IDCB_CONTAINER 1101
#define IDCB_MATRIX 1102
#define IDT_TEXT 1702
#define IDT_TEXT2 1703
#define ID_BARE_LEFT 502
#define ID_SHIELDED_RATE_ERROR 504
#define ID_FLUX_RATE_ERROR 505
#define ID_BARREL_RATE_ERROR 506
#define ID_BARE_RIGHT 508
#define ID_BARE_DOOR 510
#define ID_BARE_TOP 511
#define ID_BARE_BOTTOM 512
#define ID_SHIELDED_LEFT 513
#define ID_SHIELDED_BACK 514
#define ID_SHIELDED_RIGHT 515
#define ID_SHIELDED_DOOR 516
#define ID_SHIELDED_TOP 517
#define ID_SHIELDED_BOTTOM 518
#define ID_GROSS_SHIELDED 520
#define ID_GROSS_FLUX 521
#define ID_GROSS_BARREL 522
#define ID_SYSTEM_RATE 535
#define ID_SYSTEM_RATE_ERROR 541
#define ID_SHIELDED_RATE 546
#define ID_FLUX_RATE 547
#define ID_BARREL_RATE 548
#define ID_BARE_BACK 902
#define IDOK 1
#define ID_POWER 527
#define ID_POWER_DENSITY 528
#define ID_ACTIVITY 529
#define ID_ACTIVITY_CONCENTRATION 530
#define ID_PU_MASS 531
#define ID_AM_MASS_ERROR 532
#define ID_POWER_ERROR 1001
#define ID_POWER_DENSITY_ERROR 1002
#define ID_ACTIVITY_ERROR 1003
#define ID_ACTIVITY_CONCENTRATION_ERROR 1004
#define ID_PU_MASS_ERROR 1005
#define ID_AM_MASS 1007
#define ID_BACKG_BARREL 523
#define ID_BACKG_SHIELDED 524
#define ID_BACKG_FLUX 525
#define IDB_SLUDGE 716
#define IDB_AM_SLUDGE 717
#define ID_SYSTEM 602
#define ID_SYSTEM_ERROR 603

```

INEL-96/0057

```
#define ID_BARE_TOTAL 901
#define ID_SHIELDED_TOTAL 903
#define ID_PU_238 606
#define ID_PU_239 607
#define ID_PU_240 612
#define ID_PU_241 613
#define ID_PU_242 614
#define ID_PASSIVE_SYSTEM 1603
#define ID_PASSIVE_SYSTEM_ERROR 1604
#define ID_PASSIVE_SHIELDED 1605
#define ID_PASSIVE_SHIELDED_ERROR 1606
#define IDD_CHI_SQUARE 500
#define ID_PU_238_ERROR 615
#define ID_PU_239_ERROR 616
#define ID_PU_240_ERROR 617
#define ID_PU_241_ERROR 618
#define ID_PU_242_ERROR 619
#define IDL_USERS 201
#define IDB_SAVE 1009
#define IDB_CLOSE 1010
#define IDB_PRINT 1011
#define IDB_CONTINUE 1103
#define IDT_STATUS 401
#define IDD_USER_LIST 1800
#define IDB_SEND_RESULTS 402
#define IDB_SEND_CANCEL 403
#define IDB_EXIT_REMOTE 404
#define IDT_PROGRAM_ID 701
#define ID_BARE_SUM 904
#define ID_SHIELDED_SUM 905
#define IDB_DEFAULTS 1201
```

```
////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile: resrc1.h $
// $Revision: 2.0 $
//
// 2. Function / Purpose
//
// This file contains resource definitions generated by
// Microsoft AppStudio. See also menu.h and resrc1.h.
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
//
////////////////////////////////////
```

```

//{(NO_DEPENDENCIES)}
// App Studio generated include file.
// Used by ASSAYSYS.RC
//
#define IDD_MSG_BOX 103
#define IDD_BATCH 107
#define IDD_AUTO 108
#define IDD_DELAY 109
#define IDI_TIMER 110
#define IDC_PULSE_RATE 1205
#define IDB_DUMMY 1207
#define IDB_EDIT 1209
#define IDT_RSLT_NOTE1 1212
#define IDT_RSLT_NOTE2 1213
#define ID_U_MASS 1214
#define ID_U_MASS_ERROR 1215
#define IDB_ACCEPT 1217
#define IDB_REJECT 1218
#define IDB_NON_SLUDGE 1221
#define ID_TYPE 1222
#define IDC_CHECK 1224
#define IDC_CHECK1 1226
#define IDB_PAUSE 1228
#define IDB_GO 1229
#define IDC_LIST1 1232
#define IDC_LIST2 1233
#define IDT_RSLT_NOTE3 1234
#define IDT_COPYRIGHT 1236
#define UI_BATCH_ENABLE 40002
#define UI_NEWF 40003
#define UI_AUTO_ENABLE 40004
#define UI_DATA_SAVE 40005

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifdef APSTUDIO_READONLY_SYMBOLS

#define _APS_NEXT_RESOURCE_VALUE 111
#define _APS_NEXT_COMMAND_VALUE 40006
#define _APS_NEXT_CONTROL_VALUE 1237
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

#ifdef _INC_scaler_
#define _INC_scaler_

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL
// Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
// $Workfile: scaler.h $
// $Revision: 2.0 $
//
//

```

```
// 2. Function / Purpose
//
// The scaler class models the functionality of a generic
// CAMAC scaler board.
//
//
// 3. Interface Description,
//
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0    05 Oct 1993 15:25:30    esm
// Initial release.
//
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#define SCALER 301

#include "camac.h"

class device;
class itstream;
class ostream;

class scaler : public camac
{
    int has_data;
    int is_check_port_set;
    unsigned char check_port;

    void set_gate_name( unsigned char, char * );
    void set_clock_name( unsigned char, char * );
    void set_count( unsigned char, unsigned long );

    void set_device_alias( unsigned char, char * );

protected:
    virtual unsigned long read_and_clear_register( unsigned char );

    virtual void clear_all_registers() {};
    virtual void accumulate_all_registers() {};

    virtual void clear_lam() {};

public:
    scaler( controller **, unsigned char, unsigned int,
            unsigned char, unsigned char );
    virtual ~scaler();

    void set_has_data() { has_data = 1; }
    int has_check_port() { return is_check_port_set; }
    unsigned char get_check_port() { return check_port; }

    virtual unsigned long read_register( unsigned char );

    device **find_device( char *, char * );

    virtual int initialize();
```

```

    virtual int accumulate();
    virtual int test_lam() { return 0; }

    virtual void read_data( itstream * );
    virtual void write_data( ostream * );

    virtual int test() { return 0; }
};

#endif

#ifndef _inc_scalport_
#define _inc_scalport_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    scalport.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The scaler_port class models a port or address on a CAMAC scaler
//      board.  The scaler board hardware accumulates counts in each port.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:24    esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

#include "port.h"

class gate_port;
class clock_port;
class device;
class gate;
class timer;
class detector;
class device;

```

INEL-96/0057

```
class device_list;
class ostream;

class scaler_port : public port
{
    unsigned long count;

    char *gate_name;
    char *clock_name;

    device *pdevice;
    device_list *pdevice_list;

public:
    scaler_port();
    virtual ~scaler_port();

    virtual void set_name( char * );
    void set_device_alias( char * );
    void set_gate_name( char * );
    char *get_gate_name() { return gate_name; }

    void set_clock_name( char * );
    char *get_clock_name() { return clock_name; }

    device *get_first_device();
    device *get_next_device();

    void set_device( gate_port * );
    void set_device( clock_port * );
    void set_device( gate ** );
    void set_device( timer ** );
    void set_device( gate **, timer ** );
    void set_device( gate **, timer **, detector ** );

    device **find_device( char *, char * );

    void set_count( unsigned long count) { this->count = count; }
    unsigned long get_count() { return count; }

    virtual void write_data( ostream * );
};

#endif

#ifndef _INC_signals_
#define _INC_signals_

////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    signals.h  $
//      $Revision:    2.0  $
//
//
```

```

// 2. Function / Purpose
//
// The signals class models the generic functionality to get signal
// rates and counts.
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:38 esm
// Initial release.
//
//
/////////////////////////////////////////////////////////////////

#include <stdlib.h>
#include "camcrate.h"
#include "errorbar.h"

class signals
{
    camac_crate *gross;
    camac_crate *background;

protected:
    error_bar get_gross_rate( char *name, char *gate_name = NULL )
        { return gross->get_rate( name, gate_name ); }
    error_bar get_background_rate( char *name, char *gate_name = NULL )
        { return background->get_rate( name, gate_name ); }

public:
    signals( camac_crate *gross, camac_crate *background )
        { this->gross = gross; this->background = background; }
    virtual ~signals() {}

    unsigned long get_gross_count( char *name, char *gate_name = NULL )
        { return gross->get_count( name, gate_name ); }
    unsigned long get_background_count( char *name, char *gate_name = NULL )
        { return background->get_count( name, gate_name ); }

};

#endif

#ifndef _INC_timer_
#define _INC_timer_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
// Title: SWEPP Assay System
// Project ID: Task Baseline Agreement SDM-92-097
// Developed by: Scientific Computing Unit, INEL

```

```
//      Funded by: Transuranic Waste Programs Unit, INEL
//
//
//      1.2 Software Module
//
//          $Workfile:   timer.h  $
//          $Revision:   2.0    $
//
//
//      2. Function / Purpose
//
//          The timer class models a timer device.
//
//
//      3. Interface Description
//
//
//
//      4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0   05 Oct 1993 15:25:28   esm
//      Initial release.
//
//
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
#define TIMER           201

#include "device.h"
#include "clockport.h"

class timer : public device
{
    clock_port *pclock_port;
    float time;

public:
    timer( scaler_port *, clock_port * );
    virtual ~timer() {};

    virtual void compute_values();
    float get_time() { return time; }
};

#endif

#ifndef _INC_user_
#define _INC_user_

////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
//      1. Identification
//
//
//      1.1 Project
//
//          Title: SWEPP Assay System
//          Project ID: Task Baseline Agreement SDM-92-097
//          Developed by: Scientific Computing Unit, INEL
//          Funded by: Transuranic Waste Programs Unit, INEL
```

•

INEL-96/0057

};

#endif

#ifndef _INC_user_list_

#define _INC_user_list_

//

//

// 1. Identification

//

// 1.1 Project

//

// Title: SWEPP Assay System

// Project ID: Task Baseline Agreement SDM-92-097

// Developed by: Scientific Computing Unit, INEL

// Funded by: Transuranic Waste Programs Unit, INEL

//

//

// 1.2 Software Module

//

// \$Workfile: userlist.h \$

// \$Revision: 2.0 \$

//

//

// 2. Function / Purpose

//

// The user list class models the list of users who have access to
// the assay system.

//

//

// 3. Interface Description

//

// Functions are defined for appending, inserting, and
// removing users. A find function searches the list for a user
// with a specified id.

//

//

// 4. History

//

// Version 2.0 Release, April, 1996 (L. V. East)

// User interface improvements, additional data integrity tests and additional

// calculational capabilities as outlined in INEL/RWMC System Change Request

// No. 13623 (February, 1996). See Software Requirements Specification,

// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed

// description of this program.

//

// Rev 1.0 05 Oct 1993 15:25:40 esm

// Initial release.

//

//

//

#include "list.h"

class user;

class user_list : public list

{

void reset();

public:

user_list(char *);

~user_list();

virtual int read_data();

```

virtual void write_data();

void append( user *pc ) { list::append( (void *) pc ); }
void insert( user *pc ) { list::insert( (void *) pc ); }
void remove( user *pc ) { list::remove( (void *) pc ); }

user *find( char * );

user *get_first() { return (user *) list::get_first(); }
user *get_next() { return (user *) list::get_next(); }
};

#endif

#ifndef _INC_users_dlg_
#define _INC_users_dlg_

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:    usersdlg.h  $
//      $Revision:    2.0  $
//
// 2. Function / Purpose
//
//      The user dialog class provides the administrator interface to
//      modify the list of users who have access to the assay system.
//
// 3. Interface Description
//
//
// 4. History
//
//      Version 2.0 Release, April, 1996 (L. V. East)
//      User interface improvements, additional data integrity tests and additional
//      calculational capabilities as outlined in INEL/RWMC System Change Request
//      No. 13623 (February, 1996). See Software Requirements Specification,
//      INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
//      description of this program.
//
//      Rev 1.0    05 Oct 1993 15:25:40    esm
//      Initial release.
//
/////////////////////////////////////////////////////////////////

class user;

class users_dlg : public CDialog
{
    char *title;
    user *current_user;

```

INEL-96/0057

```
user *puser;

char *name;
char *id;
char *pw;

int access_level;
int sel;

BOOL changed;      // Flag to indicate whether list has been changed.
BOOL can_write;    // TRUE if underlying file is write enabled.

CListBox *list_box;

BOOL get_dlg_data( int );
void set_dlg_data();
BOOL is_duplicate();

void set_button_state();
void reset_focus( int );

afx_msg void on_add();
afx_msg void on_change();
afx_msg void on_delete();
afx_msg void on_selchange();
afx_msg void on_button_select();
afx_msg void OnPaint();

public:
    users_dlg( user * );
    ~users_dlg();

    BOOL OnInitDialog();
    void OnOK();
    void OnCancel();

    DECLARE_MESSAGE_MAP();
};

#endif

#ifndef _INC_waste_dlg_
#define _INC_waste_dlg_

#include "filelim.h"

/////////////////////////////////////////////////////////////////
//
// 1. Identification
//
// 1.1 Project
//
//      Title: SWEPP Assay System
//      Project ID: Task Baseline Agreement SDM-92-097
//      Developed by: Scientific Computing Unit, INEL
//      Funded by: Transuranic Waste Programs Unit, INEL
//
// 1.2 Software Module
//
//      $Workfile:   wastedlg.h  $
//      $Revision:   2.0  $
//
// 2. Function / Purpose
//
```

```

// The waste dialog provides the user interface to identify the waste
// container type, the matrix type, and the net weight of the container
// being processed.
//
//
// 3. Interface Description
//
//
// 4. History
//
// Version 2.0 Release, April, 1996 (L. V. East)
// User interface improvements, additional data integrity tests and additional
// calculational capabilities as outlined in INEL/RWMC System Change Request
// No. 13623 (February, 1996). See Software Requirements Specification,
// INEL-96/0056, and Software Design Description, INEL-96/0057, for a detailed
// description of this program.
//
// Rev 1.0 05 Oct 1993 15:25:36 esm
// Initial release.
//
////////////////////////////////////

class matrix;
class chamber_wnd;

class waste_dlg : public CDialog
{
    char *title;
    matrix *pmatrix;
    chamber_wnd *parent;

    char buffer[RECORD_LENGTH];

    BOOL focus_to_default; // OnPaint() will set focus to default button if TRUE.
    BOOL use_current_background;
    BOOL use_current_parameters;

    void disable_buttons();

    afx_msg void OnClose();
    afx_msg void OnPaint();
    afx_msg void OnLButtonDown( UINT, CPoint );
    afx_msg void on_sel_container();
    afx_msg void on_sel_matrix();
    afx_msg void on_continue();
    afx_msg void on_close();
    afx_msg void on_edit();
    afx_msg void on_box_toggled();
    afx_msg LRESULT status_changed( WPARAM, LPARAM );

public:
    waste_dlg( char *, matrix *, chamber_wnd * );
    ~waste_dlg();

    void enable_continue( BOOL = FALSE );
    void enable_update( BOOL, BOOL = TRUE );
    void set_dlg_weight();
    void set_buttons();
    void set_use_current_flag( BOOL arg = TRUE )
        { use_current_parameters = arg; }

    BOOL OnInitDialog();

    DECLARE_MESSAGE_MAP()
};

#endif

```

Appendix B

Data Files

Appendix B

Data Files

Listings of the following SAS data files are contained in this Appendix:

ACTCRATE.DAC
ACTCRATE.CAC ⁽¹⁾
AS.INI
ASSAYSYS.INI ⁽²⁾
CAC.INI ⁽¹⁾
CALIB.DAC
CALIB.CAC ⁽¹⁾
CHISQ.DAC
CHISQ.CAC ⁽¹⁾
CONTAIN.LST
CORREL.DAC
CORREL.CAC ⁽¹⁾
DAC.INI
DEFAULT.GAM
MATRIX.LST
NUCLIDE.DAT
OPTIONS.DAC
OPTIONS.CAC ⁽¹⁾
PASCRATE.DAC
PASCRATE.CAC ⁽¹⁾
REGRESS.DAC ⁽³⁾
REGRESS.CAC ^(1,3)
SAMPLE.001 ⁽⁴⁾

⁽¹⁾ This file is associated with the Crate Assay Chamber which is not currently in use.

⁽²⁾ This file is updated each time the program exits; the listing in this appendix is an example only.

⁽³⁾ This file is not currently used.

⁽⁴⁾ Sample "save" file; included as an example only.

B.1 ACTCRATE.DAC

```
; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/ACTCRATE.DAV 1.0 11 Jan 1994 13:10:28 esm
$
```

Active

```
TIME T 0
CRATE 2

STATION 01      2323      ;LeCroy 2323A Programmable Dual Gate Generator
CHANNEL 0      EARLY_GATE_DELAY 700_us ACTIVE_PULSE
CHANNEL 1      EARLY_GATE      2000_us EARLY_GATE_DELAY
END
STATION 03      2323      ;LeCroy 2323A Programmable Dual Gate Generator
CHANNEL 0      LATE_GATE_DELAY 5700_us ACTIVE_PULSE
CHANNEL 1      LATE_GATE      10000_us LATE_GATE_DELAY
END
STATION 05      2551      ;LeCroy 2551 Scaler
GATE          EARLY_GATE
CHANNEL 0      SHIELDED TOTAL
CHANNEL 1      FLUX MONITOR
CHANNEL 2      BARREL_FLUX_MONITOR
END
END
STATION 06      2551      ;LeCroy 2551 Scaler
GATE          LATE_GATE
CHANNEL 0      SHIELDED TOTAL
CHANNEL 1      FLUX MONITOR
CHANNEL 2      BARREL_FLUX_MONITOR
END
END
STATION 11      2551      ;LeCroy 2551 Scaler
IAM_TEST      OFF
CHANNEL 6      ACTIVE_PULSE
CHECK_PORT    6
ALIAS 6      EARLY_GATE
ALIAS 6      LATE_GATE
END
STATION 25      8901      ;LeCroy 8901 GPIB Interface
END
```

END

B2. ACTCRATE.CAC

```
; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/ACTCRATE.CAV 1.0 11 Jan 1994 13:12:18 esm
$
```

Active

```
TIME T 0
CRATE 2

STATION 01      2323      ;LeCroy 2323A Programmable Dual Gate Generator
CHANNEL 0      EARLY_GATE_DELAY 700_us ACTIVE_PULSE
CHANNEL 1      EARLY_GATE      2000_us EARLY_GATE_DELAY
END
STATION 03      2323      ;LeCroy 2323A Programmable Dual Gate Generator
CHANNEL 0      LATE_GATE_DELAY 5700_us ACTIVE_PULSE
CHANNEL 1      LATE_GATE      10000_us LATE_GATE_DELAY
END
STATION 05      2551      ;LeCroy 2551 Scaler
GATE          EARLY_GATE
CHANNEL 3      SHIELDED TOTAL
CHANNEL 4      FLUX MONITOR
```

```

        CHANNEL 5 BARREL_FLUX_MONITOR
    END
END
STATION 06      2551      ;LeCroy 2551 Scaler
    GATE      LATE_GATE
        CHANNEL 3 SHIELDED_TOTAL
        CHANNEL 4 FLUX_MONITOR
        CHANNEL 5 BARREL_FLUX_MONITOR
    END
END
STATION 18      2551      ;LeCroy 2551 Scaler
    LAM_TEST   OFF
        CHANNEL 6 ACTIVE_PULSE
        CHECK_PORT 6
        ALIAS 6 EARLY_GATE
        ALIAS 6 LATE_GATE
    END
END
STATION 25      8901      ;LeCroy 8901 GPIB Interface
END

```

B3. AS.INI

```

; SWEPP Assay System Version 2.0
; Modified 1 Apr 1996 for Version 2.0 Release

; Customization parameters, run time files and directories
; for the SWEPP Passive/Active Neutron Assay System
; ** NOTE: Entries are case sensitive! **

NEUT2_CONVERSION      OFF              ; ON to use NEUT2 algorithms (Default = OFF)

Root_directory        "."

Log_file              "AS.LOG"          ; Assay System log file
Error_file            "AS.ERR"          ; Assay System error file
Printer               "LPT1"            ; System printer name
Remote                COM1:300,N,8,1    ; communication attributes

User_list             "USER.LST"         ; Login info, users, access levels, etc.
Container_list        "CONTAIN.LST"     ; List of containers and volumes
Matrix_list           "MATRIX.LST"      ; List of matrices and properties

; Define as many chambers as necessary

Chamber               "DAC.INI"          ; Drum Assay Chamber configuration
; Chamber             "CAC.INI"          ; Crate Assay Chamber

```

B4. ASSAYSYS.INI

```

[Drum Assay Chamber]
rect=0000 0000 0687 0436
max=0
icon=0
color=16777088

[Main Window]
rect=0063 0029 0758 0530
max=0
acq=0232 0282
dms=0224 0297
msg=0293 0397

[Crate Assay Chamber]
rect=0021 0041 0707 0451

```

```
max=0
icon=0
color=8454016
```

B5. CAC.INI

```
; SWEPP Assay System Version 2.0
; Modified 1 Apr 1996 for Version 2.0 release

Chamber_name                "Crate Assay Chamber"

; Menu options
SUMMARY_REPORT              ON
BACKGROUND_CORRECTION       ON
AUTO_SAVE                   ON

; Customization parameters

; Passive background check parameters

; Nominal Pu mass fractions (value followed by standard deviation)

; Directory and File locations
DIRECTORIES
    Root_directory          "CAC"                ; chamber's root directory
    Gamma_directory         "E:\"                ; directory for gamma data
    Archive_directory       "B:\CAC"            ; chamber's archive directory
    Background_directory    "BACKG"
    Verification_directory  "VERIF"
    Production_directory    "PROD"
END

FILES
    ; The nuclide data and mass fractions in the root directory for the system.
    Nuclide                 "NUCLIDE.DAT"        ; Nuclide properties
    Default_mass_data       "DEFAULT.GAM"        ; Default mass fractions from gamma

    ; The remaining files, specific to the assay chamber, are in the root
    ; directory for the chamber, a subdirectory of the system, specified above.
    Process_options         "OPTIONS.CAC"        ; chamber options
    Background_data         "BACKGRND"          ; last background data
    Verification_data       "000496"            ; last verification data
    Passive_CAMAC_config    "PASCRATE.CAC"       ; passive CAMAC crate configuration
    Active_CAMAC_config     "ACTCRATE.CAC"       ; active CAMAC crate configuration
    Calibration             "CALIB.CAC"         ; chamber calibration coefficients
    Correlation             "CORREL.CAC"         ; equation correlation coefficients
    Chi_Square              "CHISQ.CAC"         ; Chi-Square parameter file
END
```

B6. CALIB.DAC

```
CHAMBER
;Detector efficiencies
SHIELDED 0.029 0
SYSTEM 0.1265 0
;Zero matrix calibration factors
ACTIVE 4.35 0
SHIELDED_COINCIDENCE 28.1 0
SYSTEM_COINCIDENCE 1.28 0
;Mass fractions
PU_238 0.00012 0
```

INEL-96/0057

```

      PU_239 0.93826 0
      PU_240 0.0582 0
      PU_241 0.0034 0
      PU_242 0.00024 0
END
```

B7. CALIB.CAC

```

; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/CALIB.CAV 1.0 11 Jan 1994 13:12:18 esm $

CHAMBER
      ; Detector efficiencies
      SHIELDED      0.029  0.0
      SYSTEM        0.1265 0.0

      ; Zero matrix calibration factors
      ACTIVE         18.5    0.0
      SHIELDED_COINCIDENCE 84.0    0.0
      SYSTEM_COINCIDENCE  2.92    0.0

      ; Pu mass fractions
      PU_238 0.00012 0.0
      PU_239 0.93826 0.0
      PU_240 0.0582 0.0
      PU_241 0.0034 0.0
      PU_242 0.00024 0.0
END
```

B8. CHISQ.DAC

```

; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/CHISQ.DAV 1.2 03 Mar 1994 11:08:44 esm $

CHI_SQUARE
      SAMPLE_SIZE      20
      PASSIVE_COUNT_TIME 10
      ACCEPTANCE_MIN    6.844
      ACCEPTANCE_MAX    38.582
      CONFIDENCE        .99
END
```

B9. CHISQ.CAC

```

; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/CHISQ.CAV 1.1 18 Jan 1994 09:45:24 esm $

CHI_SQUARE
      SAMPLE_SIZE      20
      PASSIVE_COUNT_TIME 100
      ACCEPTANCE_MIN    8.907
      ACCEPTANCE_MAX    32.852
      CONFIDENCE        .95
END
```

B10. CONTAIN.LST

```

CONTAINER
      TYPE 55 gallon drum
      VOLUME 208
END
```

B11. CORREL.DAC

```
; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/CORREL.DAV 1.0 11 Jan 1994 13:10:42 esm
$
```

```
; Correlation coefficients from the NEUT2 FORTRAN program and LA-10774-MS,
; "The Los Alamos Second-Generation System for Passive and Active
; Neutron Assays of Drum-Size Containers", by J.T. Caldwell, et al.,
; September 1986
```

ABSORBER_INDEX_CORRELATION

```
AI_DEFAULT 1.0
AI_MIN 1.0
```

END

MODERATOR_INDEX_CORRELATION

```
; LA-10774-MS, Equation (3), Page 13
A0 1.0
A1 0.226
A2 1.04
A3 0.2924
MI_MIN 0.0
MI_MAX 0.75
```

END

ACTIVE_MATRIX_CORRECTION_FACTOR

```
; factor correlated to absorption index
; Equations (8) and (9), Page 23
AI_THRESHOLD 2.72
AMCF_AI_C0 0.5404
AMCF_AI_C1 0.612
```

```
; factor correlated to moderator index
; Equations (10) and (11), Page 24
MI_THRESHOLD 0.4
AMCF_MI_C0 0.483
AMCF_MI_C1 1.817
```

END

ACTIVE_MASS

```
; interrogation background correlation
; Equation (17), Page 35
AI_MI_C0 0.00359
AI_MI_C1 0.00139
```

```
; background scale factor
; value from NEUT2
ACTIVE_BACKGROUND_SCALE_FACTOR 0.977
```

```
; coefficients for active mass error calculation
; value from NEUT2
ACTIVE_MASS_ERR_C0 0.003
; Equation (22), Page 41
ACTIVE_MASS_ERR_C1 0.05
ACTIVE_MASS_ERR_C2 0.05
```

END

PASSIVE_MASS

```
; system coincidence correction factor correlation coefficients
; Equation (14), Page 26
SYSTEM_COINCIDENCE_CF_C0 0.5967
SYSTEM_COINCIDENCE_CF_C1 0.4187
; system coincidence correction factor correlation coefficients
; Equation (14), Page 26
SHIELDED_COINCIDENCE_CF_C0 0.8092
SHIELDED_COINCIDENCE_CF_C1 0.2337
```

END

PLUTONIUM MASS SELECTION CRITERIA
 MASS_THRESHOLD 10.0

END

AMERICIUM MASS

; Americium sludge mass computation
 ; coefficients from NEUT2
 AM_SLUDGE_MASS_C0 1.12
 AM_SLUDGE_MASS_C1 76.0
 AM_SLUDGE_MASS_C2 2.93e-4
 AM_SLUDGE_MASS_C3 0.30

END

B12. CORREL.CAC

; SWEPP Assay System
 ; \$Header: C:/DEVELOP/ASSAYSYS/VCS/CORREL.CAV 1.0 11 Jan 1994 13:12:18 , esm
 \$

; Correlation coefficients from the NEUT2 FORTRAN program and LA-10774-MS,
 ; "The Los Alamos Second-Generation System for Passive and Active
 ; Neutron Assays of Drum-Size Containers", by J.T. Caldwell, et al.,
 ; September 1986

ABSORBER INDEX CORRELATION

AI_DEFAULT 0.18
 AI_MIN 0.1

END

MODERATOR INDEX CORRELATION

; LA-10774-MS, Equation (3), Page 13
 A0 3.3
 A1 0.235
 A2 1.0
 A3 -0.048
 MI_MIN 0.0
 MI_MAX 0.70

END

ACTIVE_MATRIX_CORRECTION_FACTOR

; factor correlated to absorption index
 ; Equations (8) and (9), Page 23
 AI_THRESHOLD 2.72
 AMCF_AI_C0 0.5404
 AMCF_AI_C1 0.612

; factor correlated to moderator index
 ; Equations (10) and (11), Page 24
 MI_THRESHOLD 0.4
 AMCF_MI_C0 0.483
 AMCF_MI_C1 1.817

END

ACTIVE_MASS

; interrogation background correlation
 ; Equation (17), Page 35
 AI_MI_C0 0.00359
 AI_MI_C1 0.00139

 ; background scale factor
 ; value from NEUT2
 ACTIVE_BACKGROUND_SCALE_FACTOR 1.0

```

; coefficients for active mass error calculation
; value from NEUT2
ACTIVE_MASS_ERR_C0    0.003
; Equation (22), Page 41
ACTIVE_MASS_ERR_C1    0.05
ACTIVE_MASS_ERR_C2    0.05
END

PASSIVE_MASS
; system coincidence correction factor correlation coefficients
; Equation (14), Page 26
SYSTEM_COINCIDENCE_CF_C0    0.5967
SYSTEM_COINCIDENCE_CF_C1    0.4187
; system coincidence correction factor correlation coefficients
; Equation (14), Page 26
SHIELDED_COINCIDENCE_CF_C0    0.8092
SHIELDED_COINCIDENCE_CF_C1    0.2337
END

PLUTONIUM_MASS_SELECTION_CRITERIA
MASS_THRESHOLD        10.0
END

AMERICIUM_MASS
; Americium sludge mass computation
; coefficients from NEUT2
AM_SLUDGE_MASS_C0    1.12
AM_SLUDGE_MASS_C1    76.0
AM_SLUDGE_MASS_C2    2.93e-4
AM_SLUDGE_MASS_C3    0.30
END

```

B13. DAC.INI

```

; SWEPP Assay System Version 2.0
; Modified 1 Apr 1996 for Version 2.0 release

; Assay chamber specific run time files and directories for the SWEPP Assay System.
; NOTE: Names are case sensitive!

Chamber_name          "Drum Assay Chamber"

; Menu options
SUMMARY_REPORT        ON          ; ON/OFF - Default = ON
AUTO_SAVE              ON          ; ON/OFF - Default = OFF
BACKGROUND_CORRECTION ON          ; ON/OFF - Default = ON

; Customization parameters
ACTIVE_FIRST           YES         ; YES/NO - Default = YES
PASSIVE_DELAY          15          ; Seconds delay after active assay -
                                   ; Default = 10 sec
INCLUDE_PU242          NO          ; YES/NO - Default = NO
INCLUDE_U233           NO          ; YES/NO - Default = NO
INCLUDE_U235           YES         ; YES/NO - Default = YES
USE_GAMMA_PU_DATA      NO          ; YES/NO - Default = YES
MAX_SIDE_VAR_BASE      0.20        ; Maximum fractional variation expected
                                   ; for side detector variation under good
                                   ; statistics conditions - Default = 0.15

; Passive background check parameters
NOMINAL_SG_BKGND       0.090      ; Nominal shielded background coincidence rate.
NOMINAL_LG_BKGND       1.10       ; Nominal system background coincidence rate.
MAXIMUM_BKGND_AGE      10.0       ; Maximum time (hours) before new bkgnd run
required.

```

; Nominal Pu mass fractions (value followed by standard deviation)

NOMINAL_PU240_FRACTION 0.0587 0.010
 NOMINAL_PU241_FRACTION 0.0014 0.010

; Directory and file names

DIRECTORIES

Root_directory "DAC" ; chamber's root directory
 Gamma_directory "E:\ " ; directory for gamma data
 Archive_directory "B:\DAC" ; chamber's archive directory
 Background_directory "BACKG" ;
 Verification_directory "VERIF" ;
 Production_directory "PROD" ;

END

FILES

; The nuclide data and mass fractions in the root directory for the system.
 Nuclide "NUCLIDE.DAT" ; Nuclide properties
 Default_mass_data "DEFAULT.GAM" ; Default mass fractions from gamma

; The remaining files, specific to the assay chamber, are in the root
 ; directory for the chamber, a subdirectory of the system, specified above.
 Process_options "OPTIONS.DAC" ; chamber options
 Background_data "BACKGRND" ; last background data
 Verification_data "000496" ; last verification data
 Passive_CAMAC_config "PASCRATE.DAC" ; passive CAMAC crate configuration
 Active_CAMAC_config "ACTCRATE.DAC" ; active CAMAC crate configuration
 Calibration "CALIB.DAC" ; chamber calibration coefficients
 Correlation "CORREL.DAC" ; equation correlation coefficients
 Chi_Square "CHISQ.DAC" ; Chi-Square parameter file

END

B14. DEFAULT.GAM

; SWEPP Assay System Version 2.0
 ; Modified 16 Apr 96 for Version 2.0 release.

PU_MASS_FRACTIONS

238 0.00015 0.00008
 239 0.93940 0.00110
 240 0.05870 0.00070
 241 0.00140 0.00060
 242 0.00025 0.00005

END

B15. MATRIX.LST

MATRIX 000

TYPE Not Recorded - Unknown
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 001

TYPE First-Stage Sludge / Combined Sludge
 MASS_SELECTION Active
 SLUDGE_TYPE Americium Sludge

```

MODERATOR_INDEX 0.64
C0      1
C1      0
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 002
TYPE      Second-Stage Sludge
MASS_SELECTION Active
SLUDGE_TYPE      Sludge
MODERATOR_INDEX 0.56
C0      1
C1      0
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 003
TYPE      Organic Setups
MASS_SELECTION Active
SLUDGE_TYPE      Sludge
MODERATOR_INDEX 0.456
C0      1
C1      0
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 004
TYPE      Special Setups
MASS_SELECTION Active
SLUDGE_TYPE      Sludge
MODERATOR_INDEX 0.51
C0      1
C1      0
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 005
TYPE      Evaporator Salts
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0

```

```

      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0
END
MATRIX 007
      TYPE      Wet Sludge (Process Sludge)
      MASS_SELECTION Active
      SLUDGE_TYPE      Sludge
      MODERATOR_INDEX 0.714
      C0      1
      C1      0
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0

```

```

END
MATRIX 090
      TYPE      Dirt
      MASS_SELECTION Passive
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0

```

```

END
MATRIX 095
      TYPE      Sludge
      MASS_SELECTION Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0

```

```

END
MATRIX 152
      TYPE      Pu Neutron Sources
      MASS_SELECTION Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0

```

```

END
MATRIX 241
      TYPE      Americium Process Residue
      MASS_SELECTION Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0

```

```

C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 290
TYPE     Sludge Filters
MASS_SELECTION Default
MODERATOR_INDEX 0
C0       1
C1       0.015
C2       0
C3       0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 292
TYPE     Cemented Sludge
MASS_SELECTION Default
MODERATOR_INDEX 0
C0       1
C1       0.0358
C2       0
C3       0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 300
TYPE     Graphite Molds
MASS_SELECTION Passive
MODERATOR_INDEX 0
C0       1
C1       0.015
C2       0
C3       0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 301
TYPE     Graphite Cores
MASS_SELECTION Passive
MODERATOR_INDEX 0
C0       1
C1       0.015
C2       0
C3       0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 302
TYPE     Benelex and Plexiglas
MASS_SELECTION Passive
MODERATOR_INDEX 0

```

```

C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 303
TYPE    Scarfed Graphite Chunks
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 310
TYPE    Graphite Scarfings
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 311
TYPE    Graphite Heels
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 320
TYPE    Heavy Non-SS Metals
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      2.5
C1      0.05
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX 330

```

```

TYPE      Paper and Rags - Dry
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV        1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 335

```

TYPE      Absolute Filters (8x8)
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV        1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 336

```

TYPE      Paper and Rags - Moist
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV        1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 337

```

TYPE      Plastic and Nonleaded Rubber
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV        1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 338

```

TYPE      Insulation and CWS Filter Media
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV        1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0

```

```

      WEIGHT  0
END
MATRIX  339
      TYPE    Leaded Rubber Gloves and Aprons
      MASS_SELECTION  Passive
      MODERATOR_INDEX 0
      C0      2.6
      C1      0
      C2      0
      C3      0
      DIV     1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT  0

```

```

END
MATRIX  360
      TYPE    Insulation
      MASS_SELECTION  Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV     1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT  0

```

```

END
MATRIX  361
      TYPE    Insulation Code 361
      MASS_SELECTION  Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV     1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT  0

```

```

END
MATRIX  370
      TYPE    Leco Crucibles
      MASS_SELECTION  Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV     1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT  0

```

```

END
MATRIX  371
      TYPE    Firebrick
      MASS_SELECTION  Passive
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV     1

```

ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 372

TYPE Grit
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 374

TYPE Blacktop, Concrete, Dirt, and Sand
 MASS_SELECTION Passive
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 375

TYPE Oil-Dri Residue from Incinerator
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 376

TYPE Cemented Insulation and Filter Media
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 2.54
 C1 0
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 391

TYPE Crucible and Sand
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015

```

C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0

```

END

MATRIX 392

```

TYPE      Sand, Slag, and Crucibles
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0

```

END

MATRIX 393

```

TYPE      Sand, Slag, and Crucible Heels
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0        2.82
C1        0
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0

```

END

MATRIX 409

```

TYPE      Molten Salts, 30% Unpulverized
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0        10.8
C1        0
C2        0
C3        0
DIV       0
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0

```

END

MATRIX 410

```

TYPE      Molten Salts, 30% Pulverized
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0

```

END

MATRIX 411

```

TYPE      Electrorefining Salts
MASS_SELECTION  Default

```

```

MODERATOR_INDEX 0
C0      13
C1      0
C2      0
C3      0
DIV     0
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0
END
MATRIX 412
TYPE      Gibson Salts
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0
END
MATRIX 416
TYPE      Zinc-Magnesium Alloy Metal
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0
END
MATRIX 420
TYPE      Ash, Incinerator (Virgin)
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0
END
MATRIX 421
TYPE      Heels, Ash (>2% G/G)
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0
END

```

MATRIX 422

```

TYPE      Soot
MASS_SELECTION Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF 0 0
PASSIVE_SYSTEM_CF 0 0
WEIGHT      0

```

END

MATRIX 425

```

TYPE      Fluid Bed Ash
MASS_SELECTION Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF 0 0
PASSIVE_SYSTEM_CF 0 0
WEIGHT      0

```

END

MATRIX 430

```

TYPE      Resin, Ion Column - Unleached
MASS_SELECTION Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF 0 0
PASSIVE_SYSTEM_CF 0 0
WEIGHT      0

```

END

MATRIX 431

```

TYPE      Resin, Leached
MASS_SELECTION Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF 0 0
PASSIVE_SYSTEM_CF 0 0
WEIGHT      0

```

END

MATRIX 432

```

TYPE      Resin, Leached and Cemented
MASS_SELECTION Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV       1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF 0 0

```

```

    PASSIVE_SYSTEM_CF      0 0
    WEIGHT 0
END
MATRIX 440
    TYPE      Glass
    MASS_SELECTION  Passive
    MODERATOR_INDEX 0
    C0        1
    C1        0
    C2        0
    C3        0
    DIV       1
    ACTIVE_CF      0 0
    PASSIVE_SHIELDED_CF  0 0
    PASSIVE_SYSTEM_CF  0 0
    WEIGHT 0
END
MATRIX 441
    TYPE      Unleached / Unleached Raschig Rings
    MASS_SELECTION  Default
    MODERATOR_INDEX 0
    C0        1
    C1        0.015
    C2        0
    C3        0
    DIV       1
    ACTIVE_CF      0 0
    PASSIVE_SHIELDED_CF  0 0
    PASSIVE_SYSTEM_CF  0 0
    WEIGHT 0
END
MATRIX 442
    TYPE      Leached Raschig Rings
    MASS_SELECTION  Passive
    MODERATOR_INDEX 0
    C0        1
    C1        0.015
    C2        0
    C3        0
    DIV       1
    ACTIVE_CF      0 0
    PASSIVE_SHIELDED_CF  0 0
    PASSIVE_SYSTEM_CF  0 0
    WEIGHT 0
END
MATRIX 460
    TYPE      Washables, Rubber, Plastics
    MASS_SELECTION  Default
    MODERATOR_INDEX 0
    C0        1
    C1        0.015
    C2        0
    C3        0
    DIV       1
    ACTIVE_CF      0 0
    PASSIVE_SHIELDED_CF  0 0
    PASSIVE_SYSTEM_CF  0 0
    WEIGHT 0
END
MATRIX 463
    TYPE      Gloves, Drybox / Leaded Rubber Gloves and Aprons
    MASS_SELECTION  Passive
    MODERATOR_INDEX 0
    C0        1
    C1        0.015
    C2        0
    C3        0

```

```

DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 464

```

TYPE      Benelex and Plexiglas
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 480

```

TYPE      Unleached Light Non-SS Metal
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        2.9
C1        0
C2        0
C3        0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 481

```

TYPE      Leached Light Non-SS Metal
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 490

```

TYPE      CWS Filters
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV      1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT      0

```

END

MATRIX 900

```

TYPE      LSA Plastics, Paper, etc.
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0        1

```

```

C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX  950
TYPE    LSA Metal, Glass, etc.
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX  960
TYPE    Concrete, Asphalt, etc.
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX  970
TYPE    Wood
MASS_SELECTION  Passive
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX  976
TYPE    Building 776 Process Sludge
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT  0
END
MATRIX  978
TYPE    Laundry Sludge

```

```

MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0

```

END

MATRIX 990

```

TYPE      Dirt
MASS_SELECTION Passive
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0

```

END

MATRIX 995

```

TYPE      Sewer Sludge
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0

```

END

MATRIX 069

```

TYPE      Roaster Oxide
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0

```

END

MATRIX 295

```

TYPE      Sewage Sludge
MASS_SELECTION Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF      0 0
PASSIVE_SYSTEM_CF      0 0
WEIGHT 0

```

END

MATRIX 299
 TYPE Miscellaneous Sludge
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 312
 TYPE Graphite Coarse
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 328
 TYPE Filters, Full Flow, Incinerator
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 331
 TYPE Filters, Fulflow
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 332
 TYPE Sludge, Oily
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0

PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 333

TYPE Metal, Calcium
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 334

TYPE Blankets, Fire
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 368

TYPE Mg Oxide Crucibles
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 373

TYPE Fire Brick, Leco Heels
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0
 C3 0
 DIV 1
 ACTIVE_CF 0 0
 PASSIVE_SHIELDED_CF 0 0
 PASSIVE_SYSTEM_CF 0 0
 WEIGHT 0

END

MATRIX 377

TYPE Course Firebrick
 MASS_SELECTION Default
 MODERATOR_INDEX 0
 C0 1
 C1 0.015
 C2 0

```

      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0
END
MATRIX 378
      TYPE      Fire Brick, Pulverized / Fines
      MASS_SELECTION Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0
END
MATRIX 390
      TYPE      Slag
      MASS_SELECTION Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0
END
MATRIX 413
      TYPE      Impure Salts, Cell Cleanout
      MASS_SELECTION Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0
END
MATRIX 414
      TYPE      Direct Oxide Reduction Salt
      MASS_SELECTION Default
      MODERATOR_INDEX 0
      C0      1
      C1      0.015
      C2      0
      C3      0
      DIV      1
      ACTIVE_CF      0 0
      PASSIVE_SHIELDED_CF      0 0
      PASSIVE_SYSTEM_CF      0 0
      WEIGHT 0
END
MATRIX 423
      TYPE      Resin - Soot, Calcinated
      MASS_SELECTION Default
      MODERATOR_INDEX 0

```

```

C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT  0

```

END

MATRIX 470

```

TYPE     Heels, Ash ( <2% G/G )
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT  0

```

END

MATRIX 482

```

TYPE     Metal, Scrap
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT  0

```

END

MATRIX 483

```

TYPE     Sand - Gravel
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT  0

```

END

MATRIX 700

```

TYPE     Oases Waste
MASS_SELECTION  Default
MODERATOR_INDEX 0
C0      1
C1      0.015
C2      0
C3      0
DIV     1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF  0 0
PASSIVE_SYSTEM_CF  0 0
WEIGHT  0

```

END

MATRIX 955

```

TYPE      Roaster Oxide
MASS_SELECTION Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV        1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF 0 0
PASSIVE_SYSTEM_CF 0 0
WEIGHT      0

END
MATRIX 980
TYPE      Equipment
MASS_SELECTION Default
MODERATOR_INDEX 0
C0        1
C1        0.015
C2        0
C3        0
DIV        1
ACTIVE_CF      0 0
PASSIVE_SHIELDED_CF 0 0
PASSIVE_SYSTEM_CF 0 0
WEIGHT      0

END

```

B16. NUCLIDE.DAT

```

; Nuclide data for SWEPP Assay System
; Created: 11 Jan 1994 13:08:10 /esm
; Updated: 02 Feb 1996 for Version 2 software /lve

; NOTE: Key words are case and order sensitive!

; ALPHA_BRANCHING_RATIO      // 0.0 to 1.0
; EQUIVALENT_PU_FACTOR       // Weighting factor for Pu equivalent activity
; FISSILE_GRAM_EQUIVALENT    // Relative to 1.0 for Pu-239 (positive value)
; SPECIFIC_ACTIVITY          // Total Curies per gram (Ci/g)
; THERMAL_POWER              // Watts per gram (Watts/g)
; X_SECTION                  // "Effective" fission cross-section (barns)
; NU_BAR                     // Average number of neutrons per fission

: NOTE: Pu data must be in increasing isotopic mass order!
PU238
  ALPHA_BRANCHING_RATIO      1.00
  EQUIVALENT_PU_FACTOR       1.10
  FISSILE_GRAM_EQUIVALENT    0.113
  SPECIFIC_ACTIVITY          17.3
  THERMAL_POWER              0.573
  X_SECTION                  0.0
  NU_BAR                     0.0
END
PU239
  ALPHA_BRANCHING_RATIO      1.00
  EQUIVALENT_PU_FACTOR       1.00
  FISSILE_GRAM_EQUIVALENT    1.00
  SPECIFIC_ACTIVITY          6.29e-2
  THERMAL_POWER              1.95e-3
  X_SECTION                  690.0
  NU_BAR                     2.89
END
PU240
  ALPHA_BRANCHING_RATIO      1.00

```

	EQUIVALENT_PU_FACTOR	1.00
	FISSILE_GRAM_EQUIVALENT	2.25E-2
	SPECIFIC_ACTIVITY	2.30e-1
	THERMAL_POWER	7.16e-3
	X_SECTION	0.0
	NU_BAR	0.0
END		
PU241		
	ALPHA_BRANCHING_RATIO	2.45e-5
	EQUIVALENT_PU_FACTOR	52.0
	FISSILE_GRAM_EQUIVALENT	2.25
	SPECIFIC_ACTIVITY	1.04e2
	THERMAL_POWER	3.31e-3
	X_SECTION	0.0
	NU_BAR	0.0
END		
PU242		
	ALPHA_BRANCHING_RATIO	1.00
	EQUIVALENT_PU_FACTOR	1.10
	FISSILE_GRAM_EQUIVALENT	7.50e-3
	SPECIFIC_ACTIVITY	3.97e-3
	THERMAL_POWER	1.17e-4
	X_SECTION	0.0
	NU_BAR	0.0
END		
AM241		
	ALPHA_BRANCHING_RATIO	1.00
	EQUIVALENT_PU_FACTOR	1.00
	FISSILE_GRAM_EQUIVALENT	1.87e-2
	SPECIFIC_ACTIVITY	3.47
	THERMAL_POWER	1.16e-1
	X_SECTION	0.0
	NU_BAR	0.0
END		
U235		
	ALPHA_BRANCHING_RATIO	1.00
	EQUIVALENT_PU_FACTOR	0.00
	FISSILE_GRAM_EQUIVALENT	1.00
	SPECIFIC_ACTIVITY	2.19E-6
	THERMAL_POWER	6.04E-8
	X_SECTION	501.0
	NU_BAR	2.42
END		

B17. OPTIONS.DAC

; SWEPP Assay System
; Modified 1 Apr 1996 for Version 2.0 release

BACKGROUND_COUNT_TIME 600

PRODUCTION

ACTIVE_PULSES 2000
PASSIVE_COUNT_TIME 200

END

VERIFICATION

ACTIVE_PULSES 2000
PASSIVE_COUNT_TIME 200

END

B18. OPTIONS.CAC

```
; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/OPTIONS.CAV 1.0 11 Jan 1994 13:12:18 esm
$
```

```
BACKGROUND_COUNT_TIME      200
```

PRODUCTION

```
    ACTIVE_PULSES          5000
```

```
    PASSIVE_COUNT_TIME     200
```

```
END
```

VERIFICATION

```
    ACTIVE_PULSES          2000
```

```
    PASSIVE_COUNT_TIME     200
```

```
END
```

B19. PASCRA.DAC

```
; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/PASCRA.DAV 1.0 11 Jan 1994 13:10:42 esm
$
```

```
; This CAMAC Crate configuration file defines the passive acquisition.
; It is used for both the background and the gross.
```

Passive Background**Passive Gross**

```
    TIME T 0
```

```
    CRATE 2
```

```
STATION 07      2323      ;LeCroy 2323A Programmable Dual Gate Generator
    CHANNEL 0  SHORT_GATE_DELAY  6_us  SHIELDED_TOTAL
    CHANNEL 1  SHORT_GATE        35_us  SHORT_GATE_DELAY
```

```
END
```

```
STATION 09      2323      ;LeCroy 2323A Programmable Dual Gate Generator
    CHANNEL 0  LONG_GATE_DELAY   6_us  SYSTEM_TOTAL
    CHANNEL 1  LONG_GATE         250_us LONG_GATE_DELAY
```

```
END
```

```
STATION 11      2551      ;LeCroy 2551 Scaler
    CLOCK      10KHz_CLOCK
```

```
    CHANNEL 0  SHIELDED_TOTAL
```

```
    CHANNEL 1  SYSTEM_TOTAL
```

```
    CHANNEL 2  LONG_GATE
```

```
    CHANNEL 3  SHORT_GATE
```

```
    CHANNEL 4  10KHz_CLOCK
```

```
    CHECK_PORT 4
```

```
    CHANNEL 5  1KHz_CLOCK
```

```
    CHANNEL 10 FLUX_MONITOR
```

```
    CHANNEL 11 BARREL_FLUX_MONITOR
```

```
END
```

```
END
```

```
STATION 12      2551      ;LeCroy 2551 Scaler
    CLOCK      10KHz_CLOCK
```

```
    CHANNEL 0  BARE_DOOR
```

```
    CHANNEL 1  BARE_RIGHT
```

```
    CHANNEL 2  BARE_BACK
```

```
    CHANNEL 3  BARE_LEFT
```

```
    CHANNEL 4  BARE_TOP
```

```
    CHANNEL 5  BARE_BOTTOM
```

```
    CHANNEL 6  SHIELDED_DOOR
```

```
    CHANNEL 7  SHIELDED_RIGHT
```

```
    CHANNEL 8  SHIELDED_BACK
```

```
    CHANNEL 9  SHIELDED_LEFT
```

```

        CHANNEL 10  SHIELDED_TOP
        CHANNEL 11  SHIELDED_BOTTOM
    END
END
STATION 13  450          ;DSP 450 Gated Quad Scaler
    GATE      SHORT_GATE
    CLOCK     1MHz_CLOCK
    CHANNEL   0  SHIELDED_TOTAL
    END
    CHANNEL   1  1MHz_CLOCK
    END
    GATE      LONG_GATE
    CLOCK     1MHz_CLOCK
    CHANNEL   2  SYSTEM_TOTAL
    END
    CHANNEL   3  1MHz_CLOCK
    END
END
STATION 21  217          ;Jorway 217 Gated Clock Generator
    CHANNEL   1  1MHz_CLOCK      1000000 ; /s  passive
    CHANNEL   3  10KHz_CLOCK     10000  ; /s  passive
    CHANNEL   4  1KHz_CLOCK      1000   ; /s  passive
    END
STATION 25  8901         ;LeCroy 8901 GPIB Interface
END

```

B20. PASCRATE.CAC

```

; SWEPP Assay System
; $Header: C:/DEVELOP/ASSAYSYS/VCS/PASCRATE.CAV  1.0  11 Jan 1994 13:12:18  esm
$

```

```

; This CAMAC Crate configuration file defines the passive acquisition.
; It is used for both the background and the gross.

```

```

Passive Background
Passive Gross

```

```

    TIME_T  0
    CRATE    2

STATION 14  2323         ;LeCroy 2323A Programmable Dual Gate Generator
    CHANNEL  0  SHORT_GATE_DELAY  6_us  SHIELDED_TOTAL
    CHANNEL  1  SHORT_GATE        35_us  SHORT_GATE_DELAY
    END
STATION 16  2323         ;LeCroy 2323A Programmable Dual Gate Generator
    CHANNEL  0  LONG_GATE_DELAY   6_us  SYSTEM_TOTAL
    CHANNEL  1  LONG_GATE        250_us  LONG_GATE_DELAY
    END
STATION 18  2551         ;LeCroy 2551 Scaler
    CLOCK     10KHz_CLOCK
    CHANNEL   0  SHIELDED_TOTAL
    CHANNEL   1  SYSTEM_TOTAL
    CHANNEL   2  LONG_GATE
    CHANNEL   3  SHORT_GATE
    CHANNEL   4  10KHz_CLOCK
    CHECK PORT 4
    CHANNEL   5  1KHz_CLOCK
    CHANNEL  10  FLUX_MONITOR
    CHANNEL  11  BARREL_FLUX_MONITOR
    END
END
STATION 19  2551         ;LeCroy 2551 Scaler
    CLOCK     10KHz_CLOCK
    CHANNEL   0  BARE_DOOR
    CHANNEL   1  BARE_RIGHT

```

```

        CHANNEL 2  BARE_BACK
        CHANNEL 3  BARE_LEFT
        CHANNEL 4  BARE_TOP
        CHANNEL 5  BARE_BOTTOM
        CHANNEL 6  SHIELDED_DOOR
        CHANNEL 7  SHIELDED_RIGHT
        CHANNEL 8  SHIELDED_BACK
        CHANNEL 9  SHIELDED_LEFT
        CHANNEL 10 SHIELDED_TOP
        CHANNEL 11 SHIELDED_BOTTOM
    END
END
STATION 20  450      ;DSP 450 Gated Quad Scaler
    GATE      SHORT_GATE
    CLOCK 1MHz_CLOCK
    CHANNEL 0  SHIELDED_TOTAL
    END
    CHANNEL 1  1MHz_CLOCK
    END
    GATE      LONG_GATE
    CLOCK 1MHz_CLOCK
    CHANNEL 2  SYSTEM_TOTAL
    END
    CHANNEL 3  1MHz_CLOCK
    END
END
STATION 21  217      ;Jorway 217 Gated Clock Generator
    CHANNEL 1  1MHz_CLOCK      1000000 ; /s  passive
    CHANNEL 3  10KHz_CLOCK     10000  ; /s  passive
    CHANNEL 4  1KHz_CLOCK      1000   ; /s  passive
    END
STATION 25  8901      ;LeCroy 8901 GPIB Interface
END

```

B21. REGRESS.DAC

```

; SWEPP Assay System
; $Header:  C:/DEVELOP/ASSAYSYS/VCS/REGRESS.DAV  1.0  11 Jan 1994 13:10:44  esm
$

; There are currently no regression test cases for the Drum Assay Chamber

;BACKGROUND      "\\develop\regress\background_file_name"
;TESTCASE        "\\develop\regress\data_file_name"

```

B22. REGRESS.CAC

```

; SWEPP Assay System
; $Header:  C:/DEVELOP/ASSAYSYS/VCS/REGRESS.CAV  1.0  11 Jan 1994 13:12:24  esm
$

; There are currently no regression test cases for the Crate Assay Chamber

;BACKGROUND      "\\develop\regress\background_file_name"
;TESTCASE        "\\develop\regress\data_file_name"

```

B23. SAMPLE.001

```

;Program_ID  SWEPP Assay System, Version 2.0
;File_ID     C:\ASSAYSYS\V2_TEST\DAC\PROD\VM_5.001
;
;;File created Mon Apr 15 14:36:39 1996 by user QA Tester
;
;CHAMBER
;  ; Detector efficiencies
;  SHIELDED    0.029  0
;  SYSTEM      0.1265 0
;  ; Zero matrix calibration factors
;  ACTIVE      4.35  0
;  SHIELDED_COINCIDENCE 28.1 0
;  SYSTEM_COINCIDENCE  1.28 0
;  ; Mass fractions used for calibration
;  PU_238 0.00012  0
;  PU_239 0.93826  0
;  PU_240 0.0582  0
;  PU_241 0.0034  0
;  PU_242 0.00024  0
;END
;CONTAINER
;  TYPE  55 gallon drum
;  VOLUME 208
;  NAME  VM_5
;END
;MATRIX      001
;  TYPE  First-Stage Sludge / Combined Sludge
;  MASS_SELECTION  Active
;  SLUDGE_TYPE  Americium Sludge
;  MODERATOR_INDEX  0.64
;  ; Coefficients for active self absorption correction.
;  C0      1
;  C1      0
;  C2      0
;  C3      0
;  DIV     1
;  ; Externally supplied correction factors.
;  ACTIVE_CF      0  0
;  PASSIVE_SHIELDED_CF 0  0
;  PASSIVE_SYSTEM_CF  0  0
;  ; Net matrix weight in Kg.
;  WEIGHT 1
;END
;PU_MASS_FRACTIONS
;  238    0.00012  0
;  239    0.93977  0
;  240    0.05821  0
;  241    0.00166  0
;  242    0.00024  0
;END
;
;Active
;  TIME T 829603920      ; Mon Apr 15 14:32:00 1996
;  CRATE 2
;  STATION      1      2323
;  CHANNEL      0      EARLY_GATE_DELAY  700  ACTIVE_PULSE
;  CHANNEL      1      EARLY_GATE  2000  EARLY_GATE_DELAY
;  END
;  STATION      3      2323
;  CHANNEL      0      LATE_GATE_DELAY  5700  ACTIVE_PULSE
;  CHANNEL      1      LATE_GATE  10000  LATE_GATE_DELAY
;  END
;  STATION      5      2551
;  GATE EARLY_GATE
;  CHANNEL      0      SHIELDED_TOTAL  16548

```

```

; CHANNEL 1 FLUX_MONITOR 14380
; CHANNEL 2 BARREL_FLUX_MONITOR 13896
; END
; STATION 6 2551
; GATE LATE_GATE
; CHANNEL 0 SHIELDED_TOTAL 2409
; CHANNEL 1 FLUX_MONITOR 35
; CHANNEL 2 BARREL_FLUX_MONITOR 14
; END
; END
; STATION 11 2551
; CHANNEL 6 ACTIVE_PULSE 2000
; ALIAS 6 EARLY_GATE
; ALIAS 6 LATE_GATE
; END
;END
;
;Passive Gross
; TIME T 829603988 ; Mon Apr 15 14:33:08 1996
; CRATE 2
; STATION 7 2323
; CHANNEL 0 SHORT_GATE_DELAY 6 SHIELDED_TOTAL
; CHANNEL 1 SHORT_GATE 35 SHORT_GATE_DELAY
; END
; STATION 9 2323
; CHANNEL 0 LONG_GATE_DELAY 6 SYSTEM_TOTAL
; CHANNEL 1 LONG_GATE 250 LONG_GATE_DELAY
; END
; STATION 11 2551
; CLOCK 10KHz_CLOCK
; CHANNEL 0 SHIELDED_TOTAL 18972
; CHANNEL 1 SYSTEM_TOTAL 82197
; CHANNEL 2 LONG_GATE 70810
; CHANNEL 3 SHORT_GATE 18687
; CHANNEL 4 10KHz_CLOCK 2000276
; CHANNEL 5 1KHz_CLOCK 200027
; CHANNEL 10 FLUX_MONITOR 28
; CHANNEL 11 BARREL_FLUX_MONITOR 18
; END
; END
; STATION 12 2551
; CLOCK 10KHz_CLOCK
; CHANNEL 0 BARE_DOOR 12694
; CHANNEL 1 BARE_RIGHT 13022
; CHANNEL 2 BARE_BACK 12765
; CHANNEL 3 BARE_LEFT 12568
; CHANNEL 4 BARE_TOP 6726
; CHANNEL 5 BARE_BOTTOM 5452
; CHANNEL 6 SHIELDED_DOOR 3685
; CHANNEL 7 SHIELDED_RIGHT 3800
; CHANNEL 8 SHIELDED_BACK 3855
; CHANNEL 9 SHIELDED_LEFT 3925
; CHANNEL 10 SHIELDED_TOP 1734
; CHANNEL 11 SHIELDED_BOTTOM 1973
; END
; END
; STATION 13 450
; GATE SHORT_GATE
; CLOCK 1MHz_CLOCK
; CHANNEL 0 SHIELDED_TOTAL 255
; END
; CHANNEL 1 1MHz_CLOCK 663387
; END
; GATE LONG_GATE
; CLOCK 1MHz_CLOCK
; CHANNEL 2 SYSTEM_TOTAL 11385

```

```

;           END
;   CHANNEL      3      1MHz_CLOCK      18002210
;           END
;   END
;   STATION      21      217
;   CHANNEL      1      1MHz_CLOCK      1000000
;   CHANNEL      3      10KHz_CLOCK      10000
;   CHANNEL      4      1KHz_CLOCK      1000
;   END
;END
;
;Passive Background
;   TIME_T 828560997      ; Wed Apr 03 11:49:57 1996
;   CRATE 2
;   STATION      7      2323
;   CHANNEL      0      SHORT_GATE_DELAY      6      SHIELDED_TOTAL
;   CHANNEL      1      SHORT_GATE      35      SHORT_GATE_DELAY
;   END
;   STATION      9      2323
;   CHANNEL      0      LONG_GATE_DELAY      6      SYSTEM_TOTAL
;   CHANNEL      1      LONG_GATE      250      LONG_GATE_DELAY
;   END
;   STATION      11      2551
;   CLOCK 10KHz_CLOCK
;   CHANNEL      0      SHIELDED_TOTAL      3333
;   CHANNEL      1      SYSTEM_TOTAL 17934
;   CHANNEL      2      LONG_GATE      17096
;   CHANNEL      3      SHORT_GATE      3266
;   CHANNEL      4      10KHz_CLOCK      6000358
;   CHANNEL      5      1KHz_CLOCK      600036
;   CHANNEL      10      FLUX_MONITOR 5
;   CHANNEL      11      BARREL_FLUX_MONITOR 17
;   END
;   END
;   STATION      12      2551
;   CLOCK 10KHz_CLOCK
;   CHANNEL      0      BARE_DOOR      3375
;   CHANNEL      1      BARE_RIGHT      2726
;   CHANNEL      2      BARE_BACK      2545
;   CHANNEL      3      BARE_LEFT      2451
;   CHANNEL      4      BARE_TOP      1852
;   CHANNEL      5      BARE_BOTTOM      1652
;   CHANNEL      6      SHIELDED_DOOR      494
;   CHANNEL      7      SHIELDED_RIGHT      524
;   CHANNEL      8      SHIELDED_BACK      580
;   CHANNEL      9      SHIELDED_LEFT      667
;   CHANNEL      10      SHIELDED_TOP 449
;   CHANNEL      11      SHIELDED_BOTTOM      619
;   END
;   END
;   STATION      13      450
;   GATE SHORT_GATE
;   CLOCK 1MHz_CLOCK
;   CHANNEL      0      SHIELDED_TOTAL      49
;   END
;   CHANNEL      1      1MHz_CLOCK      115925
;   END
;   GATE LONG_GATE
;   CLOCK 1MHz_CLOCK
;   CHANNEL      2      SYSTEM_TOTAL 838
;   END
;   CHANNEL      3      1MHz_CLOCK      4344494
;   END
;   END
;   STATION      21      217
;   CHANNEL      1      1MHz_CLOCK      1000000
;   CHANNEL      3      10KHz_CLOCK      10000

```

```

; CHANNEL      4      1KHz_CLOCK  1000
; END
;END
;
;Chamber Configuration Parameters
; ACTIVE_FIRST      YES
; INCLUDE_PU242      NO
; INCLUDE_U233      NO
; INCLUDE_U235      YES
; USE_GAMMA_PU_DATA  NO
; PASSIVE_DELAY      15
; NOMINAL_SG_BKGND   0.09
; NOMINAL_LG_BKGND   1.1
; MAXIMUM_BKGND_AGE  10
; MAX_SIDE_VAR_BASE  0.2
; NOMINAL_PU240_FRACTION  0.058  0.006
; NOMINAL_PU241_FRACTION  0.002  0.001
;END
;
;Net Active Count Rates
; SHIELDED_TOTAL_RATE 4016.55  32.2533
; CHAMBER_FLUX_RATE   3593.25  29.9806
; BARREL_FLUX_RATE    3473.3   29.4709
;END
;
;Net Passive Count Rates
; SHIELDED_TOTAL_RATE 89.2922  0.695289
; SYSTEM_TOTAL_RATE   381.04   1.45058
; SHIELDED_COINC_RATE 0.883397  0.0810207
; SYSTEM_COINC_RATE   20.7679  0.606445
;END
;
;Active/Passive Mass Values
; PASSIVE_TOTAL_PU_MASS 114.569  3.34553
; ACTIVE_TOTAL_PU_MASS  7.50001  0.0868911
; LANL_ERR_ESTIMATE     0.957444
; SELF_ABSORPTION_CF    1
; NEUTRON_EXCESS_AM_MASS 0.0990157  0.00105936
;END
;
;Reported Mass Values
; ; Pu Mass Derived from Active Assay data.
; REPORTED_TOTAL_PU_MASS 7.50001  0.0868911
; ; Default mass fractions used for Pu isotopic masses.
; REPORTED_PU238_MASS    0.000900001  1.04269e-005
; REPORTED_PU239_MASS    7.04828  0.0816576
; REPORTED_PU240_MASS    0.436575  0.00505793
; REPORTED_PU241_MASS    0.01245  0.000144239
; ; Am241 Mass Calculated using correlation coefficients.
; REPORTED_AM241_MASS    -0.041968  0.00199259
;END
;
;Derived Quantities
; ; Quantities derived from reported mass values.
; THERMAL_POWER          0.0125586  0.000306902
; THERMAL_POWER_DENSITY  0.0017087  4.17564e-005
; TOTAL_ACTIVITY          1.70849  0.0225662
; ALPHA_ACTIVITY          0.413722  0.00947639
; TRU_ACTIVITY            0.41369  0.00947614
; TRU_ACTIVITY_CONCENTRATION 413690  9476.14
; PU_CI_EQUIVALENT        0.412275  0.00946494
; FISSILE_GRAM_EQUIVALENT 7.08543  0.0816584
;END
;
;Calculated Matrix Correction Factors
; ; See MATRIX section for any externally supplied values.
; CALC_ABSORBER_INDEX     1.03453

```

```

;      CALC_MODERATOR_INDEX      0
;      CALC_ACTIVE_CF            1
;      CALC_PASSIVE_SHIELDED_CF  1.08764
;      CALC_PASSIVE_SYSTEM_CF    1.03104
;      CALC_SELF_ABSORPTION_CF   1
;END
;
;Recalculated Mass Values
;      ; Pu masses calculated using internally determined factors.
;      PASSIVE_SG_PU_MASS 26.9944 2.47579
;      PASSIVE_LG_PU_MASS 27.4033 0.800205
;      ACTIVE_PU_MASS     4.85009 0.0562336
;END

```

```

Program_ID  SWEPP Assay System, Version 2.0
File_ID     C:\ASSAYSYS\V2_TEST\DAC\PROD\VM_5.001

```

```

;File appended Mon Apr 15 14:39:01 1996 by user QA Tester

```

CHAMBER

```

; Detector efficiencies
;      SHIELDED      0.029 0
;      SYSTEM        0.1265 0
; Zero matrix calibration factors
;      ACTIVE        4.35 0
;      SHIELDED_COINCIDENCE      28.1 0
;      SYSTEM_COINCIDENCE       1.28 0
; Mass fractions used for calibration
;      PU_238 0.00012 0
;      PU_239 0.93826 0
;      PU_240 0.0582 0
;      PU_241 0.0034 0
;      PU_242 0.00024 0

```

```

END

```

CONTAINER

```

;      TYPE 55 gallon drum
;      VOLUME 208
;      NAME VM_5

```

```

END

```

MATRIX 001

```

;      TYPE First-Stage Sludge / Combined Sludge
;      MASS_SELECTION Active
;      SLUDGE_TYPE Americium Sludge
;      MODERATOR_INDEX 0.64
; Coefficients for active self absorption correction.
;      C0 1
;      C1 0
;      C2 0
;      C3 0
;      DIV 1
; Externally supplied correction factors.
;      ACTIVE_CF 0 0
;      PASSIVE_SHIELDED_CF 0 0
;      PASSIVE_SYSTEM_CF 0 0
; Net matrix weight in Kg.
;      WEIGHT 1

```

```

END

```

PU_MASS FRACTIONS

```

;      238 0.00012 0
;      239 0.93977 0
;      240 0.05821 0
;      241 0.00166 0
;      242 0.00024 0

```

```

END

```

```

Active

```

```

;      TIME_T 829603920 ; Mon Apr 15 14:32:00 1996

```

```

CRATE 2
STATION      1      2323
CHANNEL      0      EARLY_GATE_DELAY  700  ACTIVE_PULSE
CHANNEL      1      EARLY_GATE  2000  EARLY_GATE_DELAY
END
STATION      3      2323
CHANNEL      0      LATE_GATE_DELAY  5700  ACTIVE_PULSE
CHANNEL      1      LATE_GATE  10000  LATE_GATE_DELAY
END
STATION      5      2551
GATE EARLY_GATE
CHANNEL      0      SHIELDED_TOTAL  16548
CHANNEL      1      FLUX_MONITOR 14380
CHANNEL      2      BARREL_FLUX_MONITOR 13896
END
END
STATION      6      2551
GATE LATE_GATE
CHANNEL      0      SHIELDED_TOTAL  2409
CHANNEL      1      FLUX_MONITOR 35
CHANNEL      2      BARREL_FLUX_MONITOR 14
END
END
STATION      11     2551
CHANNEL      6      ACTIVE_PULSE 2000
ALIAS      6      EARLY_GATE
ALIAS      6      LATE_GATE
END
END

Passive Gross
TIME T 829603988 . ; Mon Apr 15 14:33:08 1996
CRATE 2
STATION      7      2323
CHANNEL      0      SHORT_GATE_DELAY  6  SHIELDED_TOTAL
CHANNEL      1      SHORT_GATE  35  SHORT_GATE_DELAY
END
STATION      9      2323
CHANNEL      0      LONG_GATE_DELAY  6  SYSTEM_TOTAL
CHANNEL      1      LONG_GATE  250  LONG_GATE_DELAY
END
STATION      11     2551
CLOCK 10KHz_CLOCK
CHANNEL      0      SHIELDED_TOTAL  18972
CHANNEL      1      SYSTEM_TOTAL 82197
CHANNEL      2      LONG_GATE  70810
CHANNEL      3      SHORT_GATE  18687
CHANNEL      4      10KHz_CLOCK 2000276
CHANNEL      5      1KHz_CLOCK 200027
CHANNEL      10     FLUX_MONITOR 28
CHANNEL      11     BARREL_FLUX_MONITOR 18
END
END
STATION      12     2551
CLOCK 10KHz_CLOCK
CHANNEL      0      BARE_DOOR  12694
CHANNEL      1      BARE_RIGHT 13022
CHANNEL      2      BARE_BACK  12765
CHANNEL      3      BARE_LEFT  12568
CHANNEL      4      BARE_TOP  6726
CHANNEL      5      BARE_BOTTOM 5452
CHANNEL      6      SHIELDED_DOOR 3685
CHANNEL      7      SHIELDED_RIGHT 3800
CHANNEL      8      SHIELDED_BACK 3855
CHANNEL      9      SHIELDED_LEFT 3925
CHANNEL      10     SHIELDED_TOP 1734
CHANNEL      11     SHIELDED_BOTTOM 1973

```

```

      END
    END
  STATION      13      450
    GATE      SHORT_GATE
    CLOCK     1MHz_CLOCK
  CHANNEL      0      SHIELDED_TOTAL      255
    END
  CHANNEL      1      1MHz_CLOCK      663387
    END
    GATE      LONG_GATE
    CLOCK     1MHz_CLOCK
  CHANNEL      2      SYSTEM_TOTAL      11385
    END
  CHANNEL      3      1MHz_CLOCK      18002210
    END
  END
  STATION      21      217
  CHANNEL      1      1MHz_CLOCK      1000000
  CHANNEL      3      10KHz_CLOCK      10000
  CHANNEL      4      1KHz_CLOCK      1000
  END
END

Passive Background
TIME_T 828560997 ; Wed Apr 03 11:49:57 1996
CRATE 2
STATION      7      2323
CHANNEL      0      SHORT_GATE_DELAY      6      SHIELDED_TOTAL
CHANNEL      1      SHORT_GATE      35      SHORT_GATE_DELAY
  END
STATION      9      2323
CHANNEL      0      LONG_GATE_DELAY      6      SYSTEM_TOTAL
CHANNEL      1      LONG_GATE      250      LONG_GATE_DELAY
  END
STATION      11      2551
  CLOCK     10KHz_CLOCK
CHANNEL      0      SHIELDED_TOTAL      3333
CHANNEL      1      SYSTEM_TOTAL      17934
CHANNEL      2      LONG_GATE      17096
CHANNEL      3      SHORT_GATE      3266
CHANNEL      4      10KHz_CLOCK      6000358
CHANNEL      5      1KHz_CLOCK      600036
CHANNEL      10      FLUX_MONITOR      5
CHANNEL      11      BARREL_FLUX_MONITOR      17
  END
  END
  STATION      12      2551
    CLOCK     10KHz_CLOCK
  CHANNEL      0      BARE_DOOR      3375
  CHANNEL      1      BARE_RIGHT      2726
  CHANNEL      2      BARE_BACK      2545
  CHANNEL      3      BARE_LEFT      2451
  CHANNEL      4      BARE_TOP      1852
  CHANNEL      5      BARE_BOTTOM      1652
  CHANNEL      6      SHIELDED_DOOR      494
  CHANNEL      7      SHIELDED_RIGHT      524
  CHANNEL      8      SHIELDED_BACK      580
  CHANNEL      9      SHIELDED_LEFT      667
  CHANNEL      10      SHIELDED_TOP      449
  CHANNEL      11      SHIELDED_BOTTOM      619
    END
  END
  STATION      13      450
    GATE      SHORT_GATE
    CLOCK     1MHz_CLOCK
  CHANNEL      0      SHIELDED_TOTAL      49
    END

```

```

CHANNEL      1      1MHz_CLOCK  115925
      END
      GATE  LONG_GATE
      CLOCK 1MHz_CLOCK
CHANNEL      2      SYSTEM_TOTAL 838
      END
CHANNEL      3      1MHz_CLOCK  4344494
      END
END
STATION      21      217
CHANNEL      1      1MHz_CLOCK  1000000
CHANNEL      3      10KHz_CLOCK 10000
CHANNEL      4      1KHz_CLOCK  1000
END

```

END

Gamma Mass Ratio Data

```

; Contents of file F:\VM_5.gam
; Mass Fractions and Ratios for Container VM 5
; Generated 3-APR-96 by BATCH_GIDXC_SUMMARY V. 01.10
;
SGRS_ATTENUATION_PARAMETERS
END

```

SGRS_ASSAY_VALUES

Pu_Det1_point_source	5.9120E+00	5.4430E-02
Pu_Det2_point_source	1.5004E+01	8.1486E-02
Pu_Det3_point_source	1.5096E+01	8.6961E-02

END

SGRS_FILES

Det1_save	"GN\$SAVE:A1640314540.DGE"
Attn_Det1_table	"AT\$TABLE:21.TBL"
Eff_Det1_table	"EFFICIENCY\$TABLES:A100381110.EFF"
Det1_bkg1	"BK\$SAVE:A15C1307140.DGE"
Det1_bkg2	"BK\$SAVE:A1640309150.DGE"
Det1_bkg3	"BK\$SAVE:A15C1216220.DGE"
Det2_save	"GN\$SAVE:A2640314540.DGE"
Attn_Det2_table	"AT\$TABLE:21.TBL"
Eff_Det2_table	"EFFICIENCY\$TABLES:A200382110.EFF"
Det2_bkg1	"BK\$SAVE:A2640309150.DGE"
Det2_bkg2	"BK\$SAVE:A25C1216220.DGE"
Det2_bkg3	"BK\$SAVE:A25C1307140.DGE"
Det3_save	"GN\$SAVE:A3640314540.DGE"
Attn_Det3_table	"AT\$TABLE:21.TBL"
Eff_Det3_table	"EFFICIENCY\$TABLES:A300383110.EFF"
Det3_bkg1	"BK\$SAVE:A3640309150.DGE"
Det3_bkg2	"BK\$SAVE:A35C1216220.DGE"
Det3_bkg3	"BK\$SAVE:A35C1307140.DGE"
Det4_save	"GN\$SAVE:A4640314540.DGE"
Attn_Det4_table	"AT\$TABLE:21.TBL"
Eff_Det4_table	"EFFICIENCY\$TABLES:A400384110.EFF"
Det4_bkg1	"BK\$SAVE:A4640309150.DGE"
Det4_bkg2	"BK\$SAVE:A45C1216220.DGE"
Det4_bkg3	"BK\$SAVE:A45C1307140.DGE"

END

DET1_PU_MASS_RATIOS

241	1.1430E-03	1.8836E-04
240	1.0131E-01	2.2570E-02

END

DET1_U_PU_MASS_RATIOS

235	2.5537E-02	7.5019E-03
-----	------------	------------

END

DET2 AM PU MASS RATIOS
 241 3.0522E-03 1.5782E-04
 END

DET2 PU MASS RATIOS
 241 8.9595E-04 6.6489E-05
 240 2.2315E-02 8.7477E-03
 END

DET2 U PU MASS RATIOS
 235 9.9826E-03 3.3272E-03
 END

DET3 AM PU MASS RATIOS
 241 2.7215E-03 1.3972E-04
 END

DET3 PU MASS RATIOS
 241 1.0273E-03 6.9395E-05
 240 4.4857E-02 6.1624E-03
 END

DET3 U PU MASS RATIOS
 235 1.1662E-02 2.3871E-03
 END

PU MASS FRACTIONS
 238 1.2000E-04 0.0000E+00
 239 9.4929E-01 6.9398E-03
 240 4.9426E-02 7.3191E-03
 241 9.2087E-04 6.5372E-05
 242 2.4000E-04 0.0000E+00
 END

AVG PU MASS RATIOS
 238 0.0000E+00 0.0000E+00
 240 5.2066E-02 7.7007E-03
 241 9.7006E-04 6.8497E-05
 242 0.0000E+00 0.0000E+00
 END

AVG AM PU MASS RATIOS
 241 2.8668E-03 1.6176E-04
 END

AVG U PU MASS RATIOS
 235 1.1997E-02 1.8778E-03
 END

AVG AM U MASS RATIOS
 241 0.0000E+00 0.0000E+00
 END

END

Chamber Configuration Parameters

ACTIVE FIRST	YES		
INCLUDE PU242		NO	
INCLUDE U233	NO		
INCLUDE U235	YES		
USE GAMMA PU DATA	NO		
PASSIVE DELAY		15	
NOMINAL SG BKGND	0.09		
NOMINAL LG BKGND	1.1		
MAXIMUM BKGND AGE	10		
MAX SIDE VAR BASE	0.2		
NOMINAL PU240 FRACTION		0.058	0.006

```

      NOMINAL_PU241_FRACTION      0.002  0.001
END

Net Active Count Rates
      SHIELDED_TOTAL_RATE 4016.55  32.2533
      CHAMBER_FLUX_RATE   3593.25  29.9806
      BARREL_FLUX_RATE    3473.3   29.4709
END

Net Passive Count Rates
      SHIELDED_TOTAL_RATE 89.2922  0.695289
      SYSTEM_TOTAL_RATE   381.04   1.45058
      SHIELDED_COINC_RATE  0.883397  0.0810207
      SYSTEM_COINC_RATE    20.7679  0.606445
END

Active/Passive Mass Values
      PASSIVE_TOTAL_PU_MASS      114.569  3.34553
      ACTIVE_TOTAL_PU_MASS       7.4457  0.0866737
      LANL_ERR_ESTIMATE          0.950532
      SELF_ABSORPTION_CF        1
      URANIUM_CF                 1.00729  0.00114171
      NEUTRON_EXCESS_AM_MASS     0.0991912 0.00105888
END

Reported Mass Values
      ; Pu Mass Derived from Active Assay data.
      REPORTED_TOTAL_PU_MASS      7.4457  0.0866737
      ; Default mass fractions used for Pu isotopic masses.
      REPORTED_PU238_MASS         0.000893483 1.04008e-005
      REPORTED_PU239_MASS         6.99724  0.0814533
      REPORTED_PU240_MASS         0.433414  0.00504528
      REPORTED_PU241_MASS         0.0123599 0.000143878
      ; U235 Mass Derived from Isotopic Ratio Data.
      REPORTED_U235_MASS          0.0839459 0.0130805
      ; Am241 Mass Determined from Pu-239 Mass Ratio.
      REPORTED_AM241_MASS         0.0200597 0.00115571
END

Derived Quantities
      ; Quantities derived from reported mass values.
      THERMAL_POWER              0.0196277 0.000241934
      THERMAL_POWER_DENSITY      0.00267049 3.2917e-005
      TOTAL_ACTIVITY              1.9103  0.0217992
      ALPHA_ACTIVITY              0.624908 0.00760707
      TRU_ACTIVITY                0.624876 0.00760676
      TRU_ACTIVITY_CONCENTRATION 624876 7606.76
      PU_CI_EQUIVALENT           0.623471 0.00759287
      FISSIONABLE_GRAM_EQUIVALENT 7.11923 0.0824976
END

Calculated Matrix Correction Factors
      ; See MATRIX section for any externally supplied values.
      CALC_ABSORBER_INDEX        1.03453
      CALC_MODERATOR_INDEX       0
      CALC_ACTIVE_CF             1
      CALC_PASSIVE_SHIELDED_CF   1.08764
      CALC_PASSIVE_SYSTEM_CF     1.03104
      CALC_SELF_ABSORPTION_CF    1
END

Recalculated Mass Values
      ; Pu masses calculated using internally determined factors.
      PASSIVE_SG_PU_MASS         26.9944  2.47579
      PASSIVE_LG_PU_MASS         27.4033  0.800205
      ACTIVE_PU_MASS             4.81496  0.0560925
END

```