Received by 027

SEP 18 1991

**Plutonium Isotopic Analysis System for
Plutonium Samples Enriched in ^{238}Pu
in EP 60/61 and Fuel-Clad Containers
Volume 3
Part 1: Software Manual**

Wayne D. Ruhter

**DO NOT MICROFILM
COVER**

July 1, 1991

Lawrence
Livermore
National
Laboratory

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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.

DISCLAIMER

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

DISCLAIMER

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California 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 products, 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 the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Plutonium Isotopic Analysis System for Plutonium Samples Enriched in ^{238}Pu in EP 60/61 and Fuel-Clad Containers

Volume 3 Part 1: Software Manual

Wayne D. Ruhter

July 1, 1991

Prepared under the auspices of Westinghouse Hanford Company in partial
fulfillment of Memorandum Purchase Order MJL-XMV-012823.

LAWRENCE LIVERMORE NATIONAL LABORATORY
University of California • Livermore, California • 94551

MASTER

ds
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

PREFACE

This software manual is addressed to the Westinghouse Hanford's Radioisotope Power Systems Facility personnel (programmers and supervisors) who maintain the software on the Pu-238 isotopic analysis system. The document is divided into two parts. Part 1 describes the computer codes that control the system, analyze the spectral data, and determine the relative plutonium abundances. Part 2 contains the software listings of the analysis codes.

In addition to this manual, two other manuals complete the description of the Pu-238 isotopic analysis system. The Pu-238 Isotopic Analysis Users Manual (UCRL-CR-107467, Vol. 1) explains how to operate and calibrate the system. The Pu-238 Isotopic Analysis Hardware Manual (UCRL-CR-107467, Vol. 2) describes the system components and provides service and maintenance information. In that volume are detailed instructions on required settings for the nuclear instrument modules.

All commercial items mentioned in this manual are assumed to be functioning correctly for the purposes of system operation. Users are referred to individual equipment manufacturers' manuals for details of operation, troubleshooting, and maintenance of this commercial equipment.

RECEIVED

CONTENTS

PART 1

Chapter 1

1. SYSTEM OVERVIEW	1- 3
1.1 System Description	1- 3
1.2 Program Description	1- 6
1.3 Computer Configuration	1- 6
1.3.1 Terminal I/O Configuration	1-11
1.3.2 Canberra Series 90 I/O Configuration	1-11

Chapter 2

2. OPERATING SYSTEM	2- 3
2.1 Operating System Configuration	2- 3
2.2 Required Programs and Files	2- 3

Chapter 3

3. STARTUP FILE	3- 3
---------------------------	------

Chapter 4

4. PROGRAM INSTALLATION	4- 3
4.1 VMS Operating System	4- 3
4.2 FORTRAN and Run-Time Library	4- 3
4.3 Program .EXE Files and Supporting Files	4- 3
4.4 Canberra 8673V Device Driver	4- 3
4.5 Canberra FTXX73 FORTRAN Callable Subroutines	4- 3
4.6 ADAC Device Driver and FORTRAN Callable Subroutine	4- 4
4.7 Starting the WHC_ASY238 Program	4- 4
4.8 Stopping the WHC_ASY238 Program	4- 5
4.9 Program Editing, Compilation, and Link	4- 5
4.10 Creating a Backup TK50 Tape of System Disk	4-10
4.11 Restoring System Disk from Backup Tape	4-11

Chapter 5

5. OPERATING FILES	5- 3
Files Needed for WHC_ASY238 Program	5- 4
PU238.SHP File	5- 6
PU238.DET File	5- 7
PU238.EP6 File	5- 7
PU238.FCC File	5- 7
PU238.CTL File	5- 7
ABSCOF.DAT File	5- 8
Files created by WHC_ASY238 Program	5- 8
Spectral data files	5- 8
Analysis results files.	5- 8

Chapter 5 (continued)

ERRMSG.INS listing	5- 9
ABSCOF.DAT listing	5-10
PU238.CTL entries	5-11

Chapter 6

6. COMMON BLOCKS	6- 3
6.1 Overview	6- 3
6.2 Listings	6- 3
CABUND.CMN (Isotopic abundances and relative errors)	6- 4
ADAC.CMN (Variables for calls to ADAC utility subroutines)	6- 5
CASSAY.CMN (Assay variables)	6- 6
CBACKG.CMN (Background variables)	6- 7
CCAL.CMN (Calibration option variables)	6- 8
CDATE.CMN (Current date/time)	6- 9
CDIAG.CMN (Data diagnostic variables)	6-10
CFILES.CMN (Filenames)	6-11
CLOG.CMN (Assay log variables)	6-12
CLUNS.CMN (Logical units associated with input/output)	6-13
CMC.CMN (Measurement control variables)	6-14
CMCONS.CMN (Measurement control variables from parameter file)	6-15
CMEAS.CMN (Measurement control pass flags)	6-16
CMONTH.CMN (Julian date/time transformation variables)	6-17
CMSG.CMN (Current error message)	6-18
CONFIL.CMN (Variables concerning parameter file)	6-19
CONTRL.CMN (Control flags)	6-20
CONT90.CMN (Series 90 variables)	6-21
CPASS.CMN (Password variables)	6-22
CQSUMS.CMN (Stored arrays of data results, sigma)	6-23
CSPECT.CMN (Spectrum data)	6-24
CTIM.CMN (Start time and date of assay)	6-25
CVAX.CMN (VAX information)	6-26
GRPGAM.CMN (COMMON used by the data-analysis subroutines)	6-27
P238COM.CMN (COMMON variables for data-analysis routines)	6-28
GRPLGCM.PMS (Includes following three common block files)	6-29
GRPYLG.PMS (Parameter and control information for GRPANL)	6-30
GRPGSP.PMS (Parameters associated with a given spectrum)	6-31
GRPGMS.PMS (Arrays produced in OUT38 for SVPK38)	6-32
GRPKCM.PMS (Peak information used by GRPANL)	6-33
GRPEVC.PMS (Common blocks for EVAL part of GRPANL)	6-34
GRPRFL.PMS (Transfer peak profiles from FIT to graphics)	6-35

Chapter 7

7. SUBROUTINE DESCRIPTIONS	7- 3
ACQR90 (ACQR38)	7- 4
ADATE	7- 5
AEND	7- 6
ADAC_ERR	7- 7
ASOPT (ASOPTL)	7- 8

Chapter 7 (continued)

ASSAY (ASSAY38)	7- 9
ASYP238	7-10
AUTCYC	7-11
CALEFF	7-12
CALIB	7-13
CALGPM	7-14
CHK90	7-15
CMPLTN (CMPLTL)	7-16
CURDTA	7-17
DATA90	7-18
DATTIM	7-19
DIALOG (DIALGS)	7-20
ERR_HANDLER	7-21
ERRMSG	7-22
EXEC (EXEC238)	7-23
FIT (FIT38)	7-24
FRDSK	7-25
GET_KEY	7-26
GETOPS	7-27
GET90	7-28
HEADER (HEADR38)	7-29
HELP (HELP38)	7-30
INIT90	7-31
INITLP	7-32
INTFLG	7-33
JULDAT (JULDAL)	7-34
LGMENU	7-35
LHEAD	7-36
LMCLOG	7-37
LNMN38	7-38
LODGRP	7-39
LSLOG (LSLOG8)	7-40
MCBIAS	7-41
MCCHK (MCCHK8)	7-42
MCHEAD	7-43
MCPREC	7-44
MENU	7-45
NEWNAM (NEWNM8)	7-46
OUT (OUT38)	7-47
P238GRP	7-48
P238AB	7-49
PASS	7-50
PRECHK	7-51
PRINT1 (PRINTL)	7-52
PUT90	7-53
RDGAMS	7-54
RDHDR	7-55
READ38	7-56
READB	7-57
READC (READC38)	7-58
READNM	7-59
RESET (RESET38)	7-60

Chapter 7 (continued)

ROTMON	7-61
SIZESAMP	7-62
SLEEP	7-63
STABSET	7-64
START	7-65
STATUS	7-66
SVPKS (SVPK38)	7-67
WRITLG (WRTL8)	7-68
WRTDAT	7-69
WTMCLG	7-70
YESNO	7-71

Chapter 8

8. PROGRAM FLOW	8- 3
---------------------------	------

Chapter 9

9. SUPPORTING PROGRAMS	9- 3
How to Update and Run the PU238_CHNG Program	9- 4
PU238_CHNG Program Options	9- 6
PU238_CHNG Program Subroutines	9- 7
Files Needed for PU238_CHNG Program	9- 8
Command File to Link PU238_CHNG Program Subroutines	9- 9
Operations Guide for GRPANL Program	9-10

Chapter 10

10. ERROR DIAGNOSTICS AND OPERATING ANOMALIES	10- 3
10.1 Error Diagnostic Messages	10- 3
10.2 Operating Anomalies	10- 4

ILLUSTRATIONS

PART 1

Chapter 1

Fig. 1-1. Block diagram of the Pu-238 isotopic analysis system	1- 4
Fig. 1-2. Pu-238 isotopic analysis system	1- 5
Fig. 1-3. System configuration.	1- 7

TABLES

PART 1

Chapter 1

Table 1-1. MicroVAX II minicomputer interface board device addresses and vectors.	1- 8
Table 1-2. Bit assignments for ADAC 1616 CCI board.	1- 9
Table 1-3. Bit assignments for ADAC 1616 HCO board.	1-10

Chapter 1

SYSTEM OVERVIEW

This two-part manual describes and provides instructions for installing software for Lawrence Livermore National Laboratory's Pu-238 isotopic analysis system built for Westinghouse Hanford's Radioisotope Power Systems Facility. Part 1 contains descriptions of all the subroutines found in the main software program, WHC_ASY238. Also provided in this part are general instructions for modifying a subroutine and specific directions for relinking the WHC_ASY238 program, as well as information on the supporting program PU238_CHNG. Part 2 contains listings of the Pu-238 isotopic analysis system codes.

The system uses a large (20% rel. efficiency), coaxial, n-type germanium detector (COAX). Parameter files for the detector have filenames with IS8 extensions. Spectral data files also have WH8 and I01, I02, etc. filename extensions.

1.1 SYSTEM DESCRIPTION

Nondestructive calorimetric measurements and gamma-ray spectroscopy are used to inventory the total plutonium content in the Pu-238 oxide samples. The calorimetric measurement determines the heat or power output of the sample. The gamma-ray measurement needed for interpreting the calorimetric measurement determines the relative abundances of various plutonium isotopes and americium-241. The Pu-238 isotopic analysis system does not require prior calibration.

Figure 1-1 is a block diagram of the system, which consists of one rack of electronics with a single MicroVAX computer. The basic components of the system (Fig. 1-2) are the detector, detector positioning stand, a sample cave (detector shield), a sample rotation and scanning mechanism, and the electronics rack, which houses a multichannel analyzer and other electronic units in addition to the computer. The system has an operator terminal and also a console terminal used by the supervisor to operate system programs.

During a measurement, the sample is positioned on a vertical scanning and rotation mechanism (in the sample cave) to give a time-averaged reading of the entire sample. A rotation sensor beneath the cave sends a signal to the computer if the mechanism stops rotating for any reason. Another sensor sends a signal to computer if the sample scanning mechanism (elevator) stops during a measurement.

Fig. 1-1. Block diagram of the Pu-238 isotopic analysis system.

Fig. 1-2. Pu-238 isotopic analysis system.

SYSTEM OVERVIEW

1.2 PROGRAM DESCRIPTION

The WHC_ASY238 program, which carries out control of the Canberra Series 90 multichannel analyzer (MCA) and analysis of the spectral data, is driven by a menu of one-, two-, or three-letter options chosen by the operator. Available options and their functions are described in the Pu-238 Isotopic Analysis Users Manual (UCRL-CR-107467, Vol. 1). The selected option determines the sequence of subroutines used for that option.

WHC_ASY238 also monitors the liquid nitrogen level in the detector dewar through a module mounted in the NIM bin (ORTEC 729A) that is interfaced to the computer. Monitoring occurs only when the system is waiting for the user to enter a menu option on the operator terminal. If the liquid nitrogen level is low, the program generates a warning message on the operator terminal.

The program enables/disables the loading/unloading of a sample from the instrument and starts/stops the rotation and scanning of the sample. Before a measurement begins, the position of the sample-cave lid is checked. The sample-cave lid must be closed before a measurement can start. The operator should manually position the detector to give an optimum count rate. Once the sample is loaded, the program checks the detector count rate before the actual measurement begins. Using the measured count rate, the program calculates the measurement time required to achieve 1% precision in the specific power and gives the operator the choice of using the calculated (recommended) or default measurement time. When the actual measurement begins, the sample load/unload capability is disabled. The sample cannot be unloaded until the measurement is completed, terminated, or aborted by the operator (see Pu-238 Isotopic Analysis Users Manual, UCRL-CR-107467, Vol. 1). When the measurement has stopped, the program stops the sample rotation and scanning and enables the sample unload.

The program monitors the rotation and scanning of the sample during a measurement. If the rotation or scanning stops, data acquisition is suspended, and a message to that effect appears on the operator terminal. To resume the measurement, type a carriage return on the operator terminal.

1.3 COMPUTER CONFIGURATION

Figure 1-3 shows the backplane layout of the system computer, which is a Digital Equipment Corporation (DEC) MicroVAX II minicomputer contained in a BA23-A enclosure. Standard to this computer are the MicroVAX II CPU, memory, serial line multiplexer, and disk controller boards. Additions are the Canberra 8673V, ADAC 1616 CCI, and ADAC 1616 HCO boards. Table 1-1 gives the MicroVAX II board addresses and vectors. Details of these boards and installed jumpers are given in the Pu-238 Isotopic Analysis Hardware Manual (UCRL-CR-107467, Vol. 2, Chapter 2). Table 1-2 gives bit assignments of board addresses for the 1616 CCI (contact closure input) board used for sensing TTL signals from the sample size, rotation, scanning, and LN monitor sensors. Table 1-3 gives bit assignments for the 1616 HCO (high current output) board used to start/stop sample rotation/scanning and enable/disable sample unloading/loading.

MicroVAX II chassis

M7606-EF	
M7608-BP	
M7516 DELQA	EMPTY
M3106 DZQ11	M7546 TQK-50BA
M7404 Extender	M9047 Continuity
EMPTY	EMPTY
EMPTY	EMPTY

Expansion chassis

M9405 Extender	EMPTY
M7555 RQDX3	EMPTY
ADAC 1616 HCO	EMPTY
ADAC 1616 CCI	M9047 Continuity
8673V Canberra hi-speed interface	
EMPTY	EMPTY
EMPTY	EMPTY

Fig. 1-3. System configuration

The MicroVAX II is configured as follows:

KA630-AA CPU	Q-22 bus map/interface
5-Mb on-board memory	One 73-Mb Winchester unformatted disk drive
Floating-point processor (FPP)	One TK50 magnetic tape
Console serial line unit (SLU)	Four-line asynchronous multiplexer (DZQ11)
Disk controller (RQDX3)	
RX33 high-density floppy	
5 1/4-inch disk drive	

Table 1-1. MicroVAX II minicomputer interface board device addresses and vectors. (Manufacturer's default addresses and vectors are used.)

Device	Name	CSR address	Interrupt vector	Interface board
System console ^a	OPA0	---	---	MicroVAX II CPU
Series 90 MCA	ANA	760610	370	Canberra 8673V
Printer ^a	TTA0	760100	300	DZQ11
Stabilizer ^c	TTA1	760100	300	DZQ11
User terminal ^b	TTA2	760100	300	DZQ11
Free port	TTA3	760100	300	DZQ11
Sample platform monitor	ZWA	764010	270	ADAC 1616 CCI
Sample platform control	ZWB	764000	100	ADAC 1616 HCO
RD53	PUA	772150	774	RQDX3 M7555
TK50	XQA	774440	120	M7546

^aBaud rate = 4800

^bBaud rate = 1200

^cBaud rate = 9600

Table 1-2. Bit assignments for ADAC 1616 CCI board.

Bit No.	1616 CCI board function
0	
1	Lid closed
2	Sample size 0 = Fuel-clad canister
3	Sample elevator not moving
4	Sample not rotating
5	LN sensor 0 = low LN level
6	---
7	---
8	---
9	---
10	---
11	---
12	---
13	---
14	---
15	---
<u>Address</u>	764010
<u>Vector</u>	270

All logic is 0 (low) @ TRUE.

SYSTEM OVERVIEW

Table 1-3. Bit assignments for ADAC 1616 HCO board.

Bit No.	1616 HCO board function
0	---
1	---
2	---
3	---
4	---
5	---
6	---
7	---
8	---
9	Load/unload disable; start sample rotation and elevator
10	---
11	---
12	---
13	---
14	---
15	---
<u>Address</u>	764000
<u>Vector</u>	100

All logic is 0 (low) @ TRUE.

1.3.1 Terminal I/O Configuration

Important MicroVAX II interfaces include those with the Pu-238 isotopic analysis system console and the remote operator terminal/printer (see Table 1-1).

The system console communicates with the MicroVAX II via CPU port OPA0: speed 9600 baud, 8 bits, no parity, 1 stop bit.

The Digital Equipment Corporation VT320 terminal communicates with the MicroVAX II via port TTA0 (user terminal 1): speed 9600 baud, 8 bits, no parity, 1 stop bit.

The Canberra 8232 stabilizer communicates with the MicroVAX II via port TTA1: speed 1200 baud, 8 bits, no parity, 1 stop bit.

The Digital Equipment Corporation LA75 printer is connected to the MicroVAX II via port TTA2 at 4800 baud.

1.3.2 Canberra Series 90 I/O Configuration

The MicroVAX II communicates with the Canberra Series 90 multichannel analyzer through a high-speed buffered serial interface. This interface (Canberra 8673V board) conforms to EIA standards RS-422 and RS-232C. The Canberra 8673V board is installed in a factory default configuration.

To communicate with the MicroVAX II, the Canberra Series 90 MCA must be set up under its CURRNT USER menu with HOST ASSIGN set to [yes]. See the Pu-238 Isotopic Analysis Users Manual (UCRL-CR-107467, Vol. 1, Chapter 2) and the Pu-238 Isotopic Analysis Hardware Manual (UCRL-CR-107467, Vol. 2, Chapter 3) for detailed directions on setting up the Canberra Series 90.

CHAPTER 2

CONTENTS

2. OPERATING SYSTEM	2- 3
2.1 Operating System Configuration	2- 3
2.2 Required Programs and Files	2- 3

Chapter 2

OPERATING SYSTEM

2.1 OPERATING SYSTEM CONFIGURATION

The MicroVMS operating system, version 5.4, is used on the Pu-238 isotopic analysis system. It is beyond the scope of this manual to present an in-depth discussion of the MicroVMS operating system. For additional information, the reader is referred to the appropriate Digital Equipment Corporation (DEC) hardware and software manuals.

To run the executable program that controls and monitors the Pu-238 isotopic analysis system, the FORTRAN run-time library must be present on the system. In addition, device drivers for the Canberra and ADAC 1616 CCI and HCO interface boards must be installed. Their installation is described in Chapter 4 (see paragraphs 4.4 and 4.6).

To compile and link programs, FORTRAN must be installed on the system. Version 5.1 of FORTRAN 77 is currently installed. Also, libraries for the Canberra and ADAC device drivers must be installed.

2.2 REQUIRED PROGRAMS AND FILES

The following programs and files are needed for the system to operate and to compile and link system programs:

FORTTRAN compiler	Version 5.1
FORTTRAN run-time library	
Canberra library	FTXX73.OLB in [FTXX73.OBJ]
Canberra Series 90 device driver	ANDRIVER.EXE in SYS\$SYSDEVICE:[ANDRV]
ADAC device drivers	ZWDRIVER.EXE in SYS\$SYSROOT:[SYS0.SYSEXE.ADLIBVMS]

CHAPTER 3

CONTENTS

3. STARTUP FILE	3- 3
---------------------------	------

U

Chapter 3

STARTUP FILE

Startup commands for the Pu-238 isotopic analysis program are in the system command file SYSTARTUP_V5.COM, found in the directory SYS\$SYSROOT:[SYSMGR], which executes automatically when the system is booted up. This command file may be changed as the system manager desires, but installation of the Canberra and ADAC device drivers must be preserved. Contents of the file:

This is the template for the site-specific startup command procedure.
!!

NOTE: This file replaces SYS\$MANAGER:SYSTARTUP.COM as the file to be used for site-specific startup. Only the name has been changed, the functionality of the old and new files is the same. See the documentation for further details.

!!

The logical SYS\$MICROVAX is true ("1") if you are operating on a MicroVAX system, but are not in a VAXcluster. Otherwise the logical symbol is false.

Create the symbol MICROVAX from the above logical, to be used throughout the rest of this command procedure.

```
MICROVAX = F$TRNLNM("SYS$MICROVAX")
IF (.NOT. MICROVAX) THEN GOTO SKIP_MESSAGE
WRITE SYS$OUTPUT ""
WRITE SYS$OUTPUT "The VAX/VMS system is now executing the site-specific startup"
WRITE SYS$OUTPUT ""
SKIP_MESSAGE:
```

You may add commands to this procedure or change the commands within this procedure as needed for your system. This command procedure is saved when you upgrade or update to a new version of VMS. Other startup command procedures are replaced with each release of VMS. Place symbol definitions from startup procedures that will be replaced when you upgrade to a new release of VMS in this command procedure. In this way, you can save symbol definitions from one VMS release to the next.

Beginning with VMS V5.2, OPCOM is started by default on all VMS systems. However, by default, operators will not be able to satisfy user mount requests. To allow operators to satisfy user mount requests, edit the SYS\$MANAGER:SYLOGIN command procedure by removing the command line that equates MOUNT to MOUNT/NOASSIST.

The following commands install the secondary page file, if it exists. (Note that the secondary swap file is installed by the command procedure SYS\$SYSTEM:STARTUP.COM if it exists.)

```
IF F$SEARCH("SYS$SYSROOT:<SYSEXE>PAGEFILE1.SYS") .EQS. "" THEN -
    GOTO NO_PAGEFILE1
SYSGEN := $SYSGEN
SYSGEN INSTALL SYS$SYSROOT:<SYSEXE>PAGEFILE1.SYS /PAGEFILE /NOCHECK
NO_PAGEFILE1:
```

The following commands turn off account logging and delete the account log. To keep an account log on your system, delete the following 5 command lines.

```
IF (.NOT. MICROVAX) THEN GOTO SKIP_ACCOUNTING
SET ACCOUNTING/DISABLE
IF F$SEARCH("SYS$MANAGER:ACCOUNTNG.DAT") .NES. "" THEN -
    DELETE SYS$MANAGER:ACCOUNTNG.DAT;*
SKIP_ACCOUNTING:
```

```

! IF (.NOT. MICROVAX) THEN GOTO SKIP_PURGING
! IF F$SEARCH("SYS$MANAGER:OPERATOR.LOG;-1") .NES. "" THEN -
!   PURGE SYS$MANAGER:OPERATOR.LOG
! IF F$SEARCH("SYS$MANAGER:EVL.LOG;-1") .NES. "" THEN -
!   PURGE SYS$MANAGER:EVL.LOG
! SKIP_PURGING:

```

! This command defines a system-wide login command procedure for all users.

```

! DEFINE /SYSTEM /EXEC SYS$SYLOGIN SYS$MANAGER:SYLOGIN

```

! This command defines a message to be displayed before each user logs in.

```

! DEFINE /SYSTEM SYS$ANNOUNCE "Welcome to VAX/VMS "F$GETSYI("VERSION")"
! DEFINE /SYSTEM SYS$ANNOUNCE "@sys$manager:announce.txt"

```

! This command defines a message or file to be displayed after each user logs in.

```

! DEFINE /SYSTEM SYS$WELCOME "@SYS$MANAGER:WELCOME.TXT"

```

! The following command starts the system queue manager. To start the queue manager, remove the comment delimiter (!) from the following two command lines and remove the dollar sign (\$) from the second command line below.

```

! START/QUEUE/MANAGER/BUFFER_COUNT=10/EXTEND_QUANTITY=25 -
!   SYS$COMMON:[SYSEXE]JBCSYSQUE.DAT

```

! Set up a print queue.

! Choose the device you will use for printing. Define the logical name \$PRINTER to name the device you will use for printing; either remove the comment delimiter (!) from one of the following commands or add a command in the following format.

```

! DEFINE /SYSTEM $PRINTER $TERMINALO ! if a hard copy terminal is the printer
! DEFINE /SYSTEM $PRINTER _LPAO: ! if using a line printer

```

! If you will use a line printer, set the printer characteristics.
! Delete the comment character from the following line.

```

! SET PRINTER /NOWRAP /NOTRUNCATE /CR $PRINTER

```

! If you will use a hard copy terminal as the print device, set the terminal characteristics. The following terminal characteristics vary from one hard copy terminal to another: speed, width, length, and device type.
! If your terminal is an LA100, remove the comment characters from the following command lines. If your terminal is not an LA100, add a command in the following format.

```

! SET TERMINAL $PRINTER /PERMANENT /NOBROADCAST /NOTYPEAHEAD /NOWRAP -
!   /SPEED=(4800) /WIDTH=(132) /PAGE=(66) /DEVICE_TYPE=(LA75)

```

To set your printing device as spooled to the queue SYS\$PRINT, remove the comment character in the following command line.

```
SET DEVICE /SPOOLED=(SYS$PRINT, SYS$SYSDEVICE:) $PRINTER.
```

To start the print queue, remove the comment character from one of the following command lines.

```
INITIALIZE /QUEUE /START /ON=$PRINTER SYS$PRINT ! terminal
INITIALIZE /QUEUE /START /ON=$PRINTER SYS$PRINT /DEFAULT=FLAG ! line printer
```

To start a batch queue, remove the comment character from the following command line.

```
INITIALIZE /QUEUE /START SYS$BATCH /BATCH /JOB:3 /BASE_PRIORITY=3.
```

To use asynchronous DECnet, set the terminal line characteristics. Change the following SET TERMINAL command to specify the terminal line you will use. Remove the comment delimiter (!) from the SET TERMINAL command line. (Reminder: To set the terminal line characteristics for the other side of the asynchronous connection, enter a SET TERMINAL command on the routing node to which you are connecting.)

```
SET TERMINAL $TERMINAL1 /PERMANENT /NOAUTOBAUD /SPEED=9600 /PROTOCOL=BDCMP
```

Following are the commands that start up the DECnet network, if you installed the DECnet software. Note that only one of these commands is needed to start the network, the other should be commented out. Configure the network and define the other node names before starting up DECnet; otherwise you will get error messages stating that the database is not initialized.

If you have batch queues set up on your system, insert a comment delimiter (!) after the dollar sign in the first command line below. Remove the comment delimiter from the second and third command lines below and remove the dollar sign from the third command line below. This allows the system to startup more quickly, and decreases the amount of time you must wait to log in.

```
IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" THEN @SYS$MANAGER:STARTNET
  IF F$SEARCH("SYS$SYSTEM:NETACP.EXE") .NES. "" - !This is faster, if you
    THEN SUBMIT SYS$MANAGER:STARTNET.COM !have batch queues set up.
```

Once you have removed the comment delimiter from one of the above commands, insert a comment delimiter after the dollar sign in the command line below. The logical is only required if the DECnet network has not been started.

```
DEFINE DECW$IGNORE_DECNET TRUE
```

The following command should be used only if you do not wish the DECwindows startup file to be run. The DECwindows startup file should be run on all types of machines, even if they are not using a DECwindows display. If you wish to run the DECwindows startup file at a later time during system startup remove the comment from the line below.

@sys#manager:andrvstr.com
@sys#manager:andrvstr.com
@SYS#SYSROOT:[sysexec.adlibvms]adlibvms.com
EXIT

CHAPTER 4

CONTENTS

4. PROGRAM INSTALLATION	4- 3
4.1 VMS Operating System	4- 3
4.2 FORTRAN and Run-Time Library	4- 3
4.3 Program .EXE Files and Supporting Files	4- 3
4.4 Canberra 8673V Device Driver	4- 3
4.5 Canberra FTXX73 FORTRAN Callable Subroutines	4- 3
4.6 ADAC Device Driver and FORTRAN Callable Subroutine	4- 4
4.7 Starting the WHC_ASY238 Program	4- 4
4.8 Stopping the WHC_ASY238 Program	4- 5
4.9 Program Editing, Compilation, and Link	4- 5
How to Update and Run the WHC_ASY238 Program	4- 8
4.10 Creating a Backup TK50 Tape of System Disk	4-10
4.11 Restoring System Disk from Backup Tape	4-11

Chapter 4

PROGRAM INSTALLATION

4.1 VMS OPERATING SYSTEM

For installation, operation, and upgrading of the MicroVMS operating system, consult DEC's VMS System Manager's Manual. It should not be necessary to install the VMS operating system, because it is saved on a backup TK50 tape of the system. However, the user may wish to upgrade the system as upgrades become available from DEC.

4.2 FORTRAN AND RUN-TIME LIBRARY

The run-time library is needed to run the WHC_ASY238 program. To compile and link programs, FORTRAN must be installed on the system. To install or upgrade FORTRAN and the run-time library on the system, consult DEC's FORTRAN Installation Guide/Release Notes found in the VAX FORTRAN manual.

4.3 PROGRAM .EXE FILES AND SUPPORTING FILES

The simplest and most reliable way to install the programs needed to operate the instrument is to copy the program .EXE files from the backup TK50 tape labeled WHC_ASY238 System Backup (see paragraph 4.11).

4.4 CANBERRA 8673V DEVICE DRIVER

The VMS device driver for the Canberra 8673V interface board has been installed on the system. The distribution kit on TK50 tape and the Canberra VMS ANDRIVER Installation Guide have been provided with the system. Instructions for installing the device driver are given in this guide.

The device name used for the Canberra 8673V board is ANA0. This name is assigned automatically during the device installation. The CSR address used for the board is 760610, and the vector address is 370.

4.5 CANBERRA FTXX73 FORTRAN CALLABLE SUBROUTINES

The Canberra FTXX73 FORTRAN callable subroutines have been installed on the system. Descriptions of these subroutines are given in the Canberra FTXX73 Software V2 CISE 422 User's Manual. Installation of this software from the distribution kit is also described in the User's manual. The library, FTXX73.OLB, must be present when the program that controls and monitors the instrument is being linked. After installation, this library file is found in the directory [FTXX73.OBJ].

4.6 ADAC DEVICE DRIVER AND FORTRAN CALLABLE SUBROUTINE

The I/O drivers and FORTRAN callable subroutines for interaction with the ADAC 1616 CCI and 1616 HCO boards are installed on the system under the directory SYS\$SYSROOT:[SYS0.SYSEXE.ADLIBVMS]. The distribution kit on TK50 tape and the ADLIBVMS VI.I User's Manual have been provided with the system. Instructions for installing the software are given in the User's manual.

I/O drivers must be built for the 1616 CCI and 1616 HCO interface boards. The drivers are assigned device names ZWA and ZWB, respectively. The ZWA driver for the 1616 CCI device must be configured for two device names, which correspond to the status and data registers on the device.

A command file, CONFIGURE.COM, that is provided in the distribution kit asks the user questions about the hardware setup. This command file creates a configuration file called ADLIBVMS.COM, which contains a list of SYSGEN commands that will load the necessary devices into the system. The ADLIBVMS.COM procedure has been included in the startup command procedure. The ADLIBVMS.COM file should contain the following instructions:

```
$ Run SYS$SYSROOT:[SYS0.SYSEXE]SYSGEN
RELOAD SYS$SYSROOT:[ADLIBVMS]ZWDRIVER
CONNECT ZWAO: /ADAPTER=0 -
/CSR=%o164010/VECTOR=%o270

CONNECT ZWA1: /ADAPTER=0 -
/CSR=%o164012/VECTOR=%o270

CONNECT ZWBO: /ADAPTER=0 -
/CSR=%o164000/VECTOR=%o100
```

Once the ADLIBVMS software has been installed on the system, all that is needed to use the callable subroutines with the instrument control and monitoring program is to include the file ADSUBS.OBJ at link time. The ADSUBS.OBJ file is located in the SYS\$SYSROOT:[SYS0.SYSEXE.ADLIBVMS] directory.

4.7 STARTING THE WHC_ASY238 PROGRAM

The WHC_ASY238 is started automatically on the operator console when the system is booted up. If the program stops or is halted, the program may be restarted by logging into a special account (to be determined by WHC: currently the account is WHC_ASY238 with no password required) on the operator terminal. This will restart the WHC_ASY238 program on the operator terminal and log the user out of the special account required to restart the program.

4.8 STOPPING THE WHC_ASY238 PROGRAM

A supervisor may stop the program from the operator terminal by using the X option. This option requires a password and stops the execution of the program. To restart the program, see paragraph 4.7.

4.9 PROGRAM EDITING, COMPILATION, AND LINK

The WHC_ASY238 program should not be reassembled unless its .EXE file is destroyed or it is necessary to change the program. The components of the WHC_ASY238 program are contained in several directories. The directory hierarchy is as follows:

```

[WHC.PU_238]      [-.ASSAY]
                  [-.SOURCE]
                  [-.OBJECT]
                  [-.LIBRARY]  [-.SOURCE]
                              [-.OBJECT]

```

These directories contain the following files for compiling and linking the WHC_ASY238 program:

[WHC.PU_238.ASSAY] - Main Directory

```

WHC_ASY238.EXE      = Working Executable

PU238_COMPILE.COM   = Command File to compile the sources in
                     [-.SOURCE] required to
                     generate WHC_ASY238 executable.

PU238_LINKER.COM     = Command File to link Object modules in
                     [-.OBJECT] and libraries in
                     [-.LIBRARY] to generate
                     WHC_ASY238 executable.

```

[WHC.PU_238.SOURCE] - Source Code Directory

```

PRINTALL.COM        = Command File to print all *.FOR and
                     *.CMN files.

PRNT.COM            = Command File to print a single file.

FOR.COM             = Command File to compile a single file
                     and produce a list file. The object
                     module is save in object module directory
                     [-.OBJECT].

*.FOR               = FORTRAN Source Code of WHC_ASY238

*.CMN               = Include files for FORTRAN code

```

PROGRAM INSTALLATION

[WHC.PU_238.OBJECT] - Object Module Directory

*.OBJ = Object Modules for WHC_ASY238

[WHC.PU_238.LIBRARY] - Library Directory

PU_COMPILE.COM = Command File to compile specific sources
for PULIB library in directory
[WHC.PU_238.LIBRARY.SOURCE] and copy
the object modules to directory
[WHC.PU_238.LIBRARY.OBJECT].

PU_LIB.COM = Command File to build PULIB.OLB library
file. Object Modules from directory
[WHC.PU_238.LIBRARY.OBJECT].

*.OLB = Linkable Library Object Modules required
to generate WHC_ASY238.

[WHC.PU_238.LIBRARY.SOURCE] - Library Source Directory

FOR.COM = Command File to compile a single file
and produce a list file. The object
module is saved in object module directory
[WHC.PU_238.LIBRARY.OBJECT].

PU_UPDATE.COM = Command File to update the current PULIB
library with define module.

*.FOR = FORTRAN Source Codes for the PULIB libraries.

[WHC.PU_238.LIBRARY.OBJECT] - Library Object Module Directory

*.OBJ = Object Modules for the PULIB libraries.

Instructions for modifying and running the WHC_ASY238 program are contained in the WHC_ASY238.RUN file, which is found in the [WHC.PU_238.SOURCE] directory.

Command files are contained in the main directory [WHC.PU_238.ASSAY] for compiling all of the source files for WHC_ASY238 and for linking them. The command file to compile the sources in [WHC.PU_238.SOURCE] is PU238_COMPILE.COM, and the command file to link the object modules and libraries to generate the WHC_ASY238 executable is PU238_LINKER.COM.

Before linking the WHC_ASY238 object modules, the library PULIB.OLB must exist. The command files for compiling the sources and for building these libraries are in the library directory [WHC.PU_238.LIBRARY]. The command files for compiling and building the library PULIB are PU_COMPILE and PU_LIB.

WHC_ASY238.RUN
22-MAY-91 WDR

HOW TO UPDATE AND RUN THE WHC_ASY238 PROGRAM

System

MicroVMS operating system with FORTRAN installed.

Editing

Use any available editor to edit the source file. All source files have the extension FOR and are contained in the directory [WHC.PU_238.SOURCE].

If necessary, change a COMMON block by editing the appropriate field with the extension CMN. If a COMMON block needs to be added to a FOR file, add an INCLUDE line to the appropriate source files (FOR) for the COMMON block file as shown:

```
INCLUDE 'CONTRL.CMN'      (NOTE the use of single quotes!)
```

Compiling

To compile a single FOR file, use the command file FOR.COM, which will compile the file specified and create a list file. For example, the ADATE.FOR file is compiled by using the command

```
@FOR ADATE
```

The compilation creates the OBJ and LIS files. The object file is stored in the directory [WHC.PU_238.OBJECT]. The listing file is stored in the source directory.

To compile all the .FOR files for the WHC_ASY238 program, type

```
@PU238_COMPILE
```

when in the main directory [WHC.PU_238.ASSAY].

Linking

To link the program WHC_ASY238, use the command

```
@PU238_LINKER
```

when in the main directory [WHC.PU_238.ASSAY].

NOTE: To link, the following library file is needed on the system in the directory [WHC.PU_238.LIBRARY]: PULIB.OLB, required for the data analysis routines (which can be created with the command files PU_COMPILE and PU_LIB). Other files needed are: [WHC.PU_238.ASSAY]GRPLIB.OLB and SYS\$SYSDEVICE:[FTXX73.OBJ]ftxx73.olb, the Canberra utility subroutine library module; and ADSUBS.OBJ, the ADAC utility subroutines module;

The new version of the program will be WHC_ASY238.EXE.

Documentation

Record all changes made in the change record, one line per routine. Print the LIS file and include it in the notebook.

Backup

Create a new TK50 backup tape per DEC instructions for creating backups.

Running

To run, these files must reside on the system:

- CONSTA.WH8 - List of parameter file names
- BACKGR.WH8 - Contains background information and passwords
- PARMTR.WH8 - Contains default parameters for instrument
- PU238.EP6 - Contains sample-absorber information for EP60/61 container
- PU238.FCC - Contains sample-absorber information for FCC
- PU238.DET - Contains detector-efficiency constants
- PU238.CTL - Contains data-analysis control information
- PU238.SHP - Contains peak-shape constants
- ABSCOF.DAT - Mass attenuation coefficients
- ANDRV.EXE - Handler for Canberra interface
- ZWDRIVER.EXE - Handlers for ADAC boards for communication with MicroVAX
- WHC_ASY238.EXE - Main program

4.10 CREATING A BACKUP TK50 TAPE OF SYSTEM DISK

To create a backup TK50 tape of the Pu-238 isotopic analysis system, you must be logged into the WHC account on the system terminal and a TK50 tape must be installed in the TK50 tape drive. (Make sure that the tape is write enabled.) Now type the following command:

```
@WHC238BACKUP
```

When the process of writing the backup tape is completed (it takes about 35 minutes), the following message appears on the system terminal:

```
Job PU238BACKUP (queue SYS$SYSBATCH, entry 319) completed
```

Remove the tape from the TK50 drive, write-protect it, label it with the appropriate saveset name, and place it in a safe place along with the stand-alone backup system TK50 tape provided with the Pu-238 isotopic analysis system.

4.11 RESTORING SYSTEM DISK FROM BACKUP TK50 TAPE

If the files on the Pu-238 isotopic analysis system disk need to be restored, do the following:

1. Make sure that the reason for the loss of the files is understood and has been remedied.

2. Install the stand-alone backup system TK50 tape in the TK50 device. (Make sure that it is write-protected.)

3. Place the MicroVAX in console I/O mode by placing the HALT ENABLE/DISABLE switch on the back of the MicroVAX in the HALT ENABLE position (UP).

4. Press the BREAK key on the system terminal; a >>> prompt should appear.

5. Following this prompt, type

B MUA0

Soon, the following will appear on the system terminal:

2..1..0..

This indicates a countdown through test procedures of the MicroVAX; if the 0.. is displayed, the tests were successful. (If the test fails, consult the MicroVAX Hardware Information manual.) The computer will then begin to boot up the stand-alone backup procedure. Boot-up takes about 15 minutes.

When the bootup procedure is completed, the following message appears on the terminal:

VAX/VMS Version 5.0 6-JUL-1987 17:00

and the date and time are requested. Type the correct date and time in the format shown. The computer will respond with warning messages such as

SYSGEN%W - NOSUCHFILE, file not found
LPDRIVER.EXE

These messages may be ignored. They are followed by the message

%BACKUP-I-IDENT, Stand-alone BACKUP V5.0, the date is 2-MAR-1988

indicating that the stand-alone backup system is ready.

6. Remove the stand-alone backup system TK50 tape from the TK50 drive and install the backup TK50 tape.

7. Type the following on the system terminal:

BACKUP MUA0:<name of saveset on tape> DUA0:/INIT

PROGRAM INSTALLATION

The saveset name should be recorded on the tape in the form of a date, e.g., 2MAR90.BCK, the date when the backup saveset was created. Once the saveset is copied to the disk drive (this may require 30 minutes to 2 hours, depending on the size of the saveset), the following messages appear on the system terminal:

```
%BACKUP-PROCDONE, Operation completed. Processing completed
```

```
2-MAR-1990 15:45:36.17
```

If you do not want to perform another stand-alone BACKUP operation, use the console to halt the system.

If you do want to perform another stand-alone BACKUP operation, ensure that the stand-alone application volume is online and ready. Enter "YES" to continue.

8. Remove the backup TK50 tape from the drive. Press the BREAK key on the system terminal; a >>> prompt should appear. Following this prompt, type

```
B DUA0
```

In a short period of time, the following will appear on the system terminal:

```
KA630-A.V1.3
```

```
Performing normal system tests.
```

```
7..6..5..4..3..
```

```
Tests completed.
```

```
Loading system software
```

```
2..1..0..
```

```
MicroVMS Version 5.1 Major version id =1 Minor version id = 0
```

This indicates that the test procedures of the MicroVAX were successful, and the system is booting up from the disk. The Pu-238 isotopic analysis control program will be restarted automatically.

8. At this point, place the HALT ENABLE/DISABLE switch on the back of the MicroVAX in the HALT DISABLE (DOWN) position.

CHAPTER 5

CONTENTS

5. OPERATING FILES	5- 3
Files Needed for WHC_ASY238 Program	5- 4
PU238.SHP File	5- 6
PU238.DET File	5- 7
PU238.EP6 File	5- 7
PU238.FCC File	5- 7
PU238.CTL File	5- 7
ABSCOF.DAT File	5- 8
Files created by WHC_ASY238 Program.	5- 8
Spectral data files.	5- 8
Analysis results files	5- 8
ERRMSG.INS listing	5- 9
ABSCOF.DAT listing	5-10
PU238.CTL entries	5-11

Chapter 5

OPERATING FILES

A file named WHC_ASY238.FIL in the main directory [WHC.PU_238.ASSAY] contains a listing and descriptions of the files required for execution of the WHC_ASY238 program. A listing of this file's contents is shown on the following pages.

The PARMTR and BACKGR files must exist at initial startup. They are initially created by running the PU238_CHNG program.

The ERRMSG.INS file is an ASCII file. It must be present at initial startup. A listing of this file is given at the end of this section.

The data-analysis control file, PU238.CTL, contains sequential sets of parametric data about the specific peak groups to be analyzed in a spectrum. This includes instructions about the starting and ending energies of each peak grouping, the widths of the background regions on either side of the peak groupings, the peaks to be used for energy scaling, the energies and relationships of peaks in each grouping, and values and options related to peak-shape parameters. This file is created and edited with the EDIGRP program. Consult "Operations Guide for GRPANL: A Suite of Computer Programs for Analyzing Complex Ge and Alpha-Particle Detector Spectra" for instructions on how to use the EDIGRP program. This reference is provided in Chapter 9. It should not be necessary for the operator to modify the PU238.CTL data-analysis control file.

The detector constants file, PU238.DET, contains the information necessary to calculate gamma emission rates from the determined peak areas. The file is in ASCII format. It contains information pertaining to the detector size, shape and efficiency. It should not be necessary to change this file unless the detector is replaced with another that is significantly different in size and shape.

The peak shape constants file, PU238.SHP, is in ASCII format. It contains five basic peak-shape constants.

The spectrum header files, PU238.EP6 and PU238.FCC, contain information on the measurement conditions. Most important is the absorber information that describes the EP60/61 and FCC sample containers and other absorbers used between the plutonium sample and the detector. These files also contain the spectrum gain and zero-channel energy offset. This file is created and edited with the EDISMP program. Entry 18 in this file contains absorber materials and their thicknesses in g/cm². These values for the EP60/61 container are: NI, 2.7; FE 6.9; CD 1.36; PB, .98; HO, 9.0. For the FCC container they are: NI, 0.15; FE 1.0; CD 1.36; PB, .98; HO, 9.0; IR, 1.37.

The assay result log file is not required at initial startup of the WHC_ASY238 program. If it does not exist, it will be created at the time results are to be written to it. Before an assay can be performed, the measurement control results file MCLOGF must exist. To create it, you must perform a bias or precision measurement that will create the file when results are written to it.

WHC_ASY238.FIL
22-MAY-91 WDR

FILES NEEDED FOR WHC_ASY238 PROGRAM

Name in data statement	File name	Description
MCLOG	MCLOGF.WH8	Log for measurement control information
ASLOG	ASLOGF.WH8	Log for assay information
CONFIL	PARMTR.WH8	Parameter file containing specific constants used in the assay
BKFIL	BACKGR.WH8	Password information
NAMSFL	CONSTA.WH8	List of parameter file names
ERRFIL	ERRMSG.INS	Error message file
	PU238.SHP	Peak-shape constants file for detector
	PU238.CTL	Data-analysis control file
	PU238.DET	Detector-efficiency constants file
	PU238.EP6	Sample-absorber information file for EP60/61 sample container
	PU238.FCC	Sample-absorber information file for fuel-clad container
	ABSCOF.DAT	Mass attenuation coefficients for absorption corrections

MCLOGF: Binary file

ASDATE(9),ASTIME(8),SAMPID(20),OPERID(8),MTYP(2),
MCRUNS,VALUE,SIGMA,VAR,ARCHV1,ARCHV2,DGFLG,BIASOK,PRECOK,
BIASOK,MCERR,MCVAX

where: ASDATE is a 9-character ASCII string for measurement control date
ASTIME is an 8-character ASCII string for measurement control time
SAMPID is a 20-character ASCII string for measurement control sample
ID
OPERID is an 8-character ASCII string for measurement control
operator ID
MTYPE is a 2-character ASCII string for measurement control type
MCRUNS is the number of bias runs
VALUE is the result for bias; reduced chi square for precision
SIGMA is the fractional error for bias run; measured variance for
precision
VAR is the standard value for bias run; calculated variance for
precision
ARCHV1 is first archival value sent to ICC VAX
ARCHV2 is second archival value sent to ICC VAX
DGFLG is a logical array for data analysis diagnostics flags
BIASOK is a logical flag to indicate if bias run was ok
PRECOK is a logical flag to indicate if precision run was ok
BACKOK is a logical flag to indicate background run was ok
MCERR is a logical variable for measurement control error
MCVAX is a logical variable for measurement control VAX flag

ASLOGF: Binary file

ASDATE(9),ASTIME(8),SAMPID(20),OPERID(8),MATYP(3),ASTYP(2),
LGFILE(15),ICYCLE,MUMCYC,ASANS,ERRBAR,DGFLG,BIASOK,PRECOK,
BACKOK,BAD,
CONDIT,LVFLAG,WTPCT(7),PCT(7)

where: ASDATE is a 9-character ASCII string for date of assay
ASTIME is an 8-character ASCII string for time of assay
SAMPID is an 8-character ASCII string for sample ID
OPERID is an 8-character ASCII string for operator ID
MATYP is a 3-character ASCII string for material type
ASTYP is a 2-character ASCII string for type of assay
LGFILE is a 15-character ASCII string for filename of assay raw
data
ICYCLE is the current run cycle number
ASANS is the assay answer - written as E13.6
ERRBAR is the assay answer fractional error - written as a real
number
DGFLG is a logical array for data analysis diagnostic flags
BIASOK is a logical flag to indicate if bias run was ok
PRECOK is a logical flag to indicate if precision run was ok
BACKOK is a logical flag to indicate background run was ok
BAD is a bad assay flag
CONDIT is a conditional assay flag
LVFLAG is a logical variable for data sent to VAX flag

OPERATING FILES

ASLOGF file continued

WTPCT is an array of 7 real values containing the isotopic weights
PCT is an array of the isotopic weight error values

PARMTR: Binary file

Parameters are set up in PU238_CHNG.FOR for each instrument

BACKGR: Binary file

BKDAT, BKTIM, BKOP, FILL, BK, SGBK, TIMEBK, PASSWD, BKSFIL

where: BKDAT is a 9-character ASCII string for background date
BKTIM is an 8-character ASCII string for background time
BKOP is a 12-character ASCII string: operator ID for background assay
FILL is one dummy character used for spacing purposes
BK is an array of 25 background areas
SGBK is an array of 25 sigmas for background areas
TIMEBK is the counting time for background
PASSWD is a 3-word array for passwords required for supervisory menu
options
BKSFIL is a 15-character ASCII string for name of background raw
data file

CONSTA: ASCII file

CFLNAM, CMMNT
15A1 , 30A1

where: CFLNAM is the constants file name
CMMNT is a remark line for the constants parameters

ERRMSG: ASCII file

NUM, IARG, ICHAR
I4, A1, 72A1

where: NUM is the error message call number
IARG is the ASCII argument passed
ICHAR is the number of characters in the error message

PU238.SHP: ASCII file

SHAPC
E13.7

where: SHAPC is an array of 10 peak shape constants determined with the
PKSNAL program.

PU238.DET: ASCII file

DCNST
24E15.7

where: (1-6) are coefficients of the polynomial function used to fit the high-energy portion of the detector's intrinsic efficiency.

(7-11) are the coefficients of the polynomial function used to fit the low-energy portion of the detector's efficiency curve.

(12) is crossover energy (keV) for the above two polynomial equations describing the efficiency curve for the detector.

(13) is the detector depth (cm).

(14) is the detector radius (cm).

(15) is the distance (cm) from the detector surface to the window.

(16) is the germanium dead layer on the front surface of the detector (g/cm^2).

(17-18) are the pair peak coefficients. (Not used for the Pu-238 isotopic analysis instrument.)

PU238.EP6 : Binary file

This file contains information pertaining to the measurement setup. It contains the sample absorber information for the EP60/61 container and the system gain. It is created and edited with the EDISMP program.

PU238.FCC : Binary file

This file contains information pertaining to the measurement setup. It contains the sample absorber information for the FCC container and the system gain. It is created and edited with the EDISMP program.

PU238.CTL: Binary file

This file contains information regarding the analysis of the spectral data. This file is created and edited with the EDIGRP program.

OPERATING FILES

ABSCOF.DAT:ASCII file

```
NSMBL(20),EDGE(20),COFMU(8,20)
1X,A2,F7.3,4F12.9/10X/4F12.9
```

This file contains mass absorption vs. energy information for each element entry. The mass absorption data for a particular element as a function of energy is fit to a seventh order polynomial. The element's symbol, its K-edge energy (keV) and the eight polynomial coefficients are stored in this file

Spectral data files: Binary file (REAL*4)

If data files are to be stored (see "D" option), then files are named with the sample ID and have the WH8 extension. Data files are unformatted with 512-byte blocks, 128 channels per block. Channel data are stored as 4-byte REALS. Thirty-two blocks are needed to store a 4096-channel spectrum. An additional block (33rd block) is used to store measurement information in 4-byte REALS in the following order:

DUM,BK, SGBK, TIMEBK, BKDAT, BKTIM, BKOP, BKSFIL, SAMPID, OPERID, SYSID, SDAT, STIM

where SAMPID is the sample ID
OPERID is the operator ID
SYSID is the system ID
SDAT is the measurement start time in days since 1/1/1900
STIM is the measurement start time of day in seconds

the other listed variables are not used in this application

Analysis results files: ASCII

File names are the sample ID with an RST extension. Analysis results are stored as ASCII in the same format as the results output listing (see Volume 1, UCRL-CR-107467, Chapter 5).

ERRMSG.INS FILE

```
100A      THIS IS A JUNK MESSAGE
  1A      *** ERROR IN OPENING FILE ***
  2A      *** ERROR IN READING FILE ***
  3A      *** ERROR IN WRITING FILE ***
  5A      *** RECORD POINTER INVALID ***
102       *** SUPERVISOR NEEDS TO CORRECT PROBLEM ***
  75      *** PRECISION CHECK WARNING LIMITS EXCEEDED ***
  76      *** PRECISION CHECK ACTION LIMITS EXCEEDED ***
  77      *** BIAS CHECK WARNING LIMITS EXCEEDED ***
  78      *** BIAS CHECK ACTION LIMITS EXCEEDED ***
  20      *** INPUT MUST BE A LETTER ***
-999
```

ABSCOF.DAT

H 0.000,-1.92846,.0307827,.693511,.547648
 .228254,.0551328,.00716967,.00038764
 N 0.000,-2.44156,.935266,2.57961,2.586
 1.4569,.467986,.0785673,.00522465
 O 0.000,-3.43302,-3.27288,-4.47342,-3.47299
 -1.43398,-.301178,-.0274736,-.000658759
 HO 0.000,-2.653999924,-0.449299991,-0.022110000,-0.046289992
 -0.022000000,0.012651702,0.007289297,0.000803100
 BE 0.000,-2.874000072,-0.493900001,-0.053600002,0.007914998
 0.006463999,0.000292600,-0.000949000,-0.000194000
 AL 0.000,-2.803000211,-0.547499895,-0.008919999,0.092049992
 0.002248999,-0.019479998,-0.003388000,0.000084460
 CA 4.037,-2.759000000,-0.495599985,-0.018610003,0.042193001
 0.002148000,-0.009788997,0.002505599,0.000877800
 FE 7.112,-2.81894,-.366719,.462304,.645550
 .320409,.0298238,-.00788492,-.00119276
 CU 8.981,-2.829999923,-0.517099976,-0.074659910,0.071426897
 0.079119992,-0.017512997,-0.011849002,-0.001256000
 GE 11.100,-2.863819599,-0.514241933,-0.055719304,0.064142191
 0.085226488,-0.019533702,-0.014515802,-0.001669029
 ZR 17.998,-2.853000164,-0.515499830,-0.028619992,0.015287002
 0.105959042,-0.007190001,-0.015634003,-0.002217999
 CD 26.711,-2.861999869,-0.530269861,0.050594007,-0.035420000
 0.097249996,0.003019199,-0.013492998,-0.002210999
 TA 67.416,-2.777000069,-0.731649994,0.373130918,-0.090159988
 0.006392999,0.013119301,-0.000515000,-0.000493000
 PT 78.395,-2.742000222,-0.822149991,0.391638994,-0.069509994
 0.007472999,0.012730604,-0.000324000,-0.000452000
 PB 88.004,-2.698999643,-0.837720036,0.439069986,-0.075499987
 -0.012566003,0.011334203,0.001791999,-0.000073000
 U 115.606,-2.594000101,-0.978459835,0.464566946,-0.020709997
 -0.001882000,0.002823799,-0.003287000,-0.000847000
 PU121.180,-2.55783,-1.46698,-1.55018,-3.11612
 -2.23006,-.800250,-.141631,-.00973186
 CR 5.988,-2.82926,-.392186,.304819,.355531
 .0849343,-.0559626,-.0222783,-.00210806
 NI 8.332,-2.79347,-.326739,.692511,1.04766
 .641407,.148717,.0125978,.000144599
 MO 20.000,-2.85944,-.461279,.447096,.851473
 0.826400,.303489,.0487307,.00289381
 IR 76.111,-2.72630,-.811315,.292270,-.043257
 0.123387,0.0384493,-.0167456,-.00533886
 0.000,0.000000000,0.000000000,0.000000000,0.000000000
 0.000000000,0.000000000,0.000000000,0.000000000

PU238.CTL

The following is a summary of the information stored in the PU238.CTL file. This information is entered into this file using the EDIGRP program in the supervisor's directory. It should not be necessary to modify this information or reenter it unless the file is lost and cannot be replaced.

<u>Group</u>	<u>Low energy bound</u>	<u>High energy end</u>	Width (keV) of bkgrd region ADJACENT to low <u>E bound</u>	Width (keV) of bkgrd region ADJACENT to <u>high E end</u>	<u>Peak Energies</u>
1	199.7	201.9	Remote E=195. width = 1.	Remote E=205. width = 0.5	201.1
2	206.7	209.7	0.5	0.5	208.
3	236.3	242.3	1.0	1.0	238.626
4	372.8	377.3	0.5	0.5	241.1
					375.042
					376.6 (fixed in energy to 375.04 peak)
5	410.0	418.0	1.0	1.0	411.15 (fixed in energy to 413.7 peak)
					413.712
					415.75
6	580.4	586.1	1.5	1.5	583.139
7	724.6	730.5	0.5	0.5	727.27
8	738.9	746.6	1.0	1.0	742.82
9	760.7	772.9	0.5	1.0	766.41
10	783.0	788.8	2.0	2.0	786.3
11	858.0	863.5	2.0	2.0	860.49
12	310.25	314.25	1.0	1.0	311.9
13	149.80	155.10	1.0	1.0	152.68

All singlet peaks are simply integrated. Those peak regions with multiple peak entries are fit.

CHAPTER 6

CONTENTS

6.	COMMON BLOCKS	6- 3
6.1	Overview	6- 3
6.2	Listings	6- 3
	CABUND.CMN (Isotopic abundances and relative errors)	6- 4
	ADAC.CMN (Variables for calls to ADAC utility subroutines)	6- 5
	CASSAY.CMN (Assay variables)	6- 6
	CBACKG.CMN (Background variables)	6- 7
	CCAL.CMN (Calibration option variables)	6- 8
	CDATE.CMN (Current date/time)	6- 9
	CDIAG.CMN (Data diagnostic variables)	6-10
	CFILES.CMN (Filenames)	6-11
	CLOG.CMN (Assay log variables)	6-12
	CLUNS.CMN (Logical units associated with input/output)	6-13
	CMC.CMN (Measurement control variables)	6-14
	CMCONS.CMN (Measurement control variables from parameter file)	6-15
	CMEAS.CMN (Measurement control pass flags)	6-16
	CMONTH.CMN (Julian date/time transformation variables)	6-17
	CMSG.CMN (Current error message)	6-18
	CONFIL.CMN (Variables concerning parameter file)	6-19
	CONTRL.CMN (Control flags)	6-20
	CONT90.CMN (Series 90 variables)	6-21
	CPASS.CMN (Password variables)	6-22
	CQSUMS.CMN (Stored arrays of data results, sigma)	6-23
	CSPECT.CMN (Spectrum data)	6-24
	CTIM.CMN (Start time and date of assay)	6-25
	CVAX.CMN (VAX information)	6-26
	GRPGAM.CMN (COMMON used by the data-analysis subroutines)	6-27
	P238COM.CMN (COMMON variables for data-analysis routines)	6-28
	GRPLGCM.PMS (Includes following three common block files)	6-29
	GRPYLG.PMS (Parameter and control information for GRPANL)	6-30
	GRPGSP.PMS (Parameters associated with a given spectrum)	6-31
	GRPGMS.PMS (Arrays produced in OUT38 for SVPK38)	6-32
	GRPKCM.PMS (Peak information used by GRPANL)	6-33
	GRPEVC.PMS (Common blocks for EVAL part of GRPANL)	6-34
	GRPREFL.PMS (Transfer peak profiles from FIT to graphics)	6-35

Chapter 6

COMMON BLOCKS

6.1 OVERVIEW

Each common block in the SRS_ASY238 program is contained in an individual file named with the common block name and the extension CMN. A common block is included in a source file by using an INCLUDE statement; for example, INCLUDE CONTRL causes the inclusion of the CONTRL.CMN file, which contains the CONTRL common block. The INCLUDE statement specifies that the contents of a file are to be incorporated in the FORTRAN compilation directly following the INCLUDE statement.

The INCLUDE statement has the form

```
INCLUDE '[file-spec]/[NO]LIST'
```

where:

"file-spec" is a character string that specifies the file (file name) to be included in the compilation.

the /LIST qualifier indicates that the statements in the specified file are to be listed in the compilation source listing. The default is "/NOLIST".

6.2 LISTINGS

Listings of the individual common blocks, including definitions of the variables contained within, are given on the following pages.

COMMON BLOCKS

COMMON /CABUND/ WTPCT(7),PCT(7)

C
C Isotopic abundances and their relative errors
C
C WTPCT = WEIGHT PERCENT ABUNDANCES
C WTPCT(1) = PU238,, WTPCT(5) = AM241, WTPCT(6) = PU242
C WTPCT(7) = U235/PU
C PCT = PERCENTAGE ERROR ON EACH ABUNDANCE

```
COMMON /ADAC/      C_DATA,C_STATUS,H_DATA,H_STATUS

C
C  Common variables used for communication with the ADAC 1616CCI
C  and 1616HCO boards
C

      INTEGER*4      C_DATA,C_STATUS
      INTEGER*4      H_DATA,H_STATUS
      INTEGER*2      IOSB(4)
      LOGICAL*1      ERROR_CHECK, ESTATUS
      EXTERNAL        ERROR_CHECK, DINW,DINIW,BSETW,BCLRW,BTSTW
      INTEGER*4      DINW,DINIW,BSETW,BCLRW,BTSTW
```

COMMON BLOCKS

CHARACTER SAMPID*20, OPERID*8

LOGICAL*1 ASTYPE(2), MATTYP(3)

LOGICAL*1 BAD, CONDIT, LVFLAG

COMMON /CASSAY/ TIMEPR, TIMELV, TIMECL, NUMCYC, ICYCLE,

X SAMPID, OPERID, ASANS, ERRBAR, C240FG, ASTYPE, MATTYP,

X BAD, CONDIT

C

C Assay variables

C TIMEPR = Preset assay time

C TIMELV = Live time after assay finished

C TIMECL = Clock time after assay finished

C NUMCYC = Number of cycles for autocycle run

C ICYCLE = Current cycle number

C SAMPID = 20-character ASCII sample ID

C OPERID = 8-character ASCII operator ID

C ASANS = Assay answer for assay log

C ERRBAR = Error bar for assay results

C C240FG = Pu-240 calculated or declared

C ASTYPE = Assay measurement type

C MATTYP = 3-character material type

C BAD = Bad assay flag

C CONDIT = Conditional assay flag

C

```
REAL*4 BK(25),SGBK(25)
CHARACTER*9 BKDAT
CHARACTER*8 BKTIM
LOGICAL*1 BD,BKLIMS(10)
CHARACTER*15 BKOP,BKSFIL
COMMON /CBACKG/ BK,SGBK,TIMEBK,BKDAT,BKTIM,
X      BKOP,BKSFIL,BD,BKLIMS
```

```
C
C      Background variables
C BK      = 25 background areas
C SGBK     = 25 sigmas for background areas
C TIMEBK  = Counting time for background
C BKDAT   = 9-character ASCII string for background date
C BKTIM   = 8-character ASCII string for background time
C BKOP    = 12-character ASCII string: operator id for background assay
C BKSFIL  = 15-character ASCII string for name of background raw data file
C BD      = Logical flag for background option from disk option
C BKLIMS  = Background limit checks - set for up to 10
C
```

COMMON BLOCKS

```
LOGICAL * 1 CAL, CALDSK
INTEGER CALCYC
REAL * 4 CALCNS
COMMON /CCAL/ CAL,CALDSK,CALCNS,CALCYC
```

C

C Calibration option variables

C CAL = Logical flag for calibration option select

C CALDSK = Logical flag for calibration from disk option select

C CALCNS = Calibration constant

C CALCYC = Calibration cycle times

```
CHARACTER*9 NOWDAT,ASDATE
CHARACTER*8 NOWTIM,ASTIME
COMMON /CDATE/ NOWDAT,NOWTIM,ASDATE,ASTIME
```

C

C Current date/time

C NOWDAT = 9-character ASCII string: current date (dd-mm-yy)

C NOWTIM = 8-character ASCII string: current time (hh:mm:ss)

C Assay date/time

C ASDATE = 9-character ASCII string: assay date (dd-mm-yy)

C ASTIME = 8-character ASCII string: assay time (hh:mm:ss)

C

COMMON BLOCKS

```
REAL*4 DIAVAL(10),DIALIM(10)
INTEGER*2 DERR(11),DERRNM,DIAGMX
LOGICAL*1 DGFLG(10)
LOGICAL*1 DIAGTB(15)
COMMON /CDIAG/ DIAGMX,DIAVAL,DIALIM,DERR,DERRNM,DGFLG,DIAGTB
```

```
C
C      Data diagnostic variables
C  DIAGMX  =  Maximum number of diagnostic checks input in parameter file
C  DIAVAL  =  Value of the diagnostic checks
C  DIALIM  =  Limits of the diagnostic checks
C  DERR    =  Error number for the data diagnostic error - references
C            an error message printed out in the PRINT routine
C  DERRNM  =  Error number in the diagnostic table DIAGCK
C  DGFLG   =  Logical flag for data diagnostic - set if error in data
C  DIAGTB  =  File 'DIAGCK' which contains data diagnostic checks for
C            the instrument
```

```
CHARACTER*15 FILNAM, ASLOG, BKFIL, MCLOG, ERRFIL  
CHARACTER*15 CONFIL, NAMSFL
```

C

```
COMMON /CFILES/ FILNAM, ASLOG, BKFIL, MCLOG, ERRFIL,  
X      CONFIL, NAMSFL
```

C

C Filenames

```
C FILNAM = 15-character ASCII string: data file  
C ASLOG  = 15-character ASCII string: assay log file  
C BKFIL  = 15-character ASCII string: background areas file  
C MCLOG  = 15-character ASCII string: measurement control log  
C ERRFIL = 15-character ASCII string: error message file  
C CONFIL = 15-character ASCII string: constants file name  
C NAMSFL = 15-character ASCII string: list of constants files  
C
```

COMMON BLOCKS

CHARACTER LGFILE*15

COMMON /CLOG/ MAXASY, LOGREC, LGFILE

C

C

Assay log variables

C LGFILE = 15-character ASCII array; filename for assay raw data

C MAXASY = Total number of records in the assay file

C LOGREC = Record size of assay record

INTEGER DKLUN, DKLUN1

C

COMMON /CLUNS/ LOUT,DKLUN,DKLUN1

C

C

Logical units associated with input/output

C

C LOUT = Logical output device for printout

C DKLUN = Logical input/output device for disk files

C DKLUN1 = Second logical i/o device for disk files

C

COMMON BLOCKS

```
LOGICAL*1 MCERR,MCVAX,MTYP(2)
COMMON /CMC/ VALUE,SIGMA,VAR,ARCHV1,ARCHV2,
1          MCMAX,MCREC,MCRUNS,MCERR,MCVAX,MTYP
```

```
C
C      Measurement control variables
C  VALUE = Result in grams
C  SIGMA = Fractional sigma of result
C  VAR   = Standard value for bias; calculated variance for precision
C  ARCHV1 = Archival value 1 (for log and VAX comm)
C  ARCHV2 = Archival value 2 (for log and VAX comm)
C  MCMAX  = Maximum record number in MC log
C  MCREC  = Measurement control log record size
C  MCRUNS = Number of precision runs
C  MCERR  = Logical flag set for error in measurement control run
C  MCVAX  = Logical flag set for entry sent to VAX
C  MTYP   = 2-character ASCII array; MCRUN type
C
```

```
      INTEGER BKINDX
      COMMON /CMCONS/ BITIME,BIREF,HISTSD,PRTIME,NRUNS,
X          BKTIME,BKLIM(2),BKINDX(2),BASTIM
C      Measurement control variables from parameter file
C
C BITIME      = Time between bias checks
C BIREF       = Standard reference value for bias check
C HISTSD      = Historical standard deviation of the measured value
C PRTIME      = Time between precision checks
C NRUNS       = Number of autocycle runs for precision check
C BKTIME      = Time between background checks
C BKLIM(2)    = Maximum of two background limits
C BKINDX(2)   = Index of the two background limits
C BASTIM      = Background preset assay time
C
```

COMMON BLOCKS

```
      LOGICAL*1 BIASOK,PRECOK,BACKOK,MCTIM
C
      COMMON /CMEAS/ BIASOK,PRECOK,BACKOK,MCTIM
C
      Measurement control pass flags
C
C  BIASOK = Flag for acceptable bias check
C  PRECOK = Flag for acceptable precision check
C  BACKOK = Flag for acceptable background check
C  MCTIM  = Flag for acceptable MC time intervals
C
```

CHARACTER AMONTH*3

COMMON /CMONTH/ AMONTH(12), IDAY, IMONTH, IYEAR, IHOUR,
IMIN, ISEC

X

C

Julian date/time transformation variables

C AMONTH = 3-character ASCII month names (JAN, FEB, MAR, etc.)

C IDAY = Day of month (1-31)

C IMONTH = Integer month (1-12)

C IYEAR = Year-1900 (83, 84, 85)

C IHOUR = Integer hour (0-24)

C IMIN = Minute

C ISEC = Second

C

COMMON BLOCKS

```
LOGICAL*1 MESSAG(72)
COMMON /CMMSG/ MESSAG, ICHAR
```

```
C
C      Current error message
C MESSAG = 72-character ASCII error message
C ICHAR  = Integer message character count
C
```

```
LOGICAL*1 NEWFIL  
CHARACTER LSTFIL*12,COMMNT*30,FILOP*8  
CHARACTER*9 FILDAT  
CHARACTER*8 FILTIM  
COMMON /CONFIL/ LSTFIL,COMMNT,NEWFIL,FILDAT,FILTIM,FILOP
```

C

C Variables concerning parameter file

C LSTFIL = 15-character ASCII string: name of last constants file read

C COMMNT = 30-character ASCII string: comment concerning current constants file

C NEWFIL = Logical variable flagging a new constants file

C FILDAT = 9-character ASCII string: date current constants file written

C FILTIM = 8-character ASCII string: time current constants file written

C FILOP = 8-character ASCII string: ID of operator who wrote constants file

C

COMMON BLOCKS

```
      LOGICAL*1 ABORT, BACKG, WRITNG, FROMDK, AUTOCY, VAXCOM,
X      OLDATA, LIVE, LONGPR, CUMLTV, BEGIN, STPCYC,
X      ASFLG, AC, AUD, AD, MP, MB, TKEY, CHANG, CHANG1
      REAL * 8 RTYP
      COMMON /CONTRL/ RTYP, ITMOUT, IRETRY, ABORT, BACKG, WRITNG, VAXCOM,
X      FROMDK, AUTOCY, OLDATA, LIVE, LONGPR, CUMLTV, BEGIN, STPCYC,
X      ASFLG, AC, AUD, AD, MP, MB, TKEY, CHANG, CHANG1
C
C      Control flags
C ABORT  = Abort flag returned from subroutines
C BACKG  = Background collect flag
C WRITNG = Write data to disk flag
C FROMDK = Read disk data
C AUTOCY = Autocycle
C OLDATA = Analyze old data
C LIVE   = Live data analysis
C LONGPR = Output long printout
C CUMLTV = Accumulate assays
C BEGIN  = Start-up, initialize
C STPCYC = Stop cycle
C VAXCOM = Communicate with VAX
C ITMOUT = Time out for VAX communication
C IRETRY = Number of times to try for VAX acknowledgement
C ASFLG  = Assay option flag
C AC     = Current data in MCA option flag
C AUD    = Autocycle from disk option flag
C AD     = Logical flag to indicate assay from disk
C MB     = Measurement control bias
C MP     = Measurement control precision
C RTYP   = Type of measurement - used in printout
C TKEY   = Logical flag to indicate abort by T key in data acquisition
C CHANG  = Logical flag to indicate CHANGE program is running
C CHANG1 = Logical flag to indicate CHANGE should get password
C
```

```
COMMON /CONT90/ IADC,MCA,MEMSIZ,TLIVE,TCLOCK,IDLE,ISTAT
```

```
C
```

```
C      Series 90 variables
```

```
C  IADC  = ADC number
```

```
C  MCA   = MCA number (default: 1)
```

```
C  MEMSIZ = Size of memory setup for this ADC
```

```
C  TLIVE  = Live time (channel 0)
```

```
C  TCLOCK = Clock time (channel 1)
```

```
C  IDLE   = Idle(1) or busy(0) returned from INQADC
```

```
C  ISTAT  = Status of ADC (manual, remote)
```

```
C
```

COMMON BLOCKS

```
LOGICAL * 1 OK
REAL * 8 PASSWD
COMMON /CPASS/ PASSWD(3)
```

C

C Password variables

C OK = Logical flag for the password check

C PASSWD = 3-word array for passwords needed for menu options

C

```
COMMON /CQSUMS/ XVAL(20),XSIG(20)
```

```
C
C      Stored arrays of data results, sigma
C  XVAL = Array of results
C  XSIG = Array of sigma
C
```

COMMON BLOCKS

LOGICAL*1 RAWDTA,NOBACK

COMMON /CSPECT/ SPCTRM(128),IFIRST,RAWDTA,NOBACK

C

C

Spectrum data

C SPCTRM = 128 channels of spectrum

C IFIRST = Channel number of first channel in SPCTRM

C RAWDTA = Logical flag for raw data to be sent to VAX

C NOBACK = Logical flag for no background data at end of data file

COMMON /CTIM/ STIM,SDAT

C

C STIM = Start time of assay (seconds since midnight)

C SDAT = Start date of assay (# days since 1900)

COMMON BLOCKS

```
LOGICAL*1 VAXUP,BUSY,NOGO,ANINC
CHARACTER*9 VAXDAT
CHARACTER*8 VAXTIM
```

C

```
COMMON /CVAX/ VAXUP,VAXDAT,VAXTIM,BUSY,NOGO,ANINC
```

C

C

```
VAX information
```

C

```
VAXUP = Logical flag for VAX status
```

C

```
VAXDAT = 9-character ASCII string: date returned from VAX
```

C

```
VAXTIM = 8-character ASCII string: time returned from VAX
```

C

```
BUSY = Message from instrument to VAX on local busy status
```

C

```
NOGO = Message from VAX to instrument - Don't go
```

C

```
ANINC = Message from VAX to instrument - Ancillary measurements
```

C

```
incomplete
```

C

```
C   C   C   C           C   C           C           C   C
C
C   "GRPGMS.FOR" = Arrays produced in OUT
C       Last edit 30-Jan-84
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   PARAMETER MXPB=50
    BYTE BKFLGS

    COMMON/GAM/CHNL(50),ENG(50),GAMS(50),PCTER(50),
      1  COUNTS(50),PKS(50),PKE(50),NGAMS,BKFLGS(50)

    DIMENSION FRCE(50)
    EQUIVALENCE (PCTER,FRCE)
    DIMENSION PRPK(2)
    EQUIVALENCE (OLSTD1,PRPK),(KALPFG,NDFIL),(INPDEV,MGMS),
      1  (KTGRP,NSPEC)
```

COMMON BLOCKS

COMMON /P238COM/PU240,PU242,
IZPEAK,IZWNW,ZARNG,IGPEAK,IGWNDW,GARNG

C	PU240	User input of 240Pu abundance
C	PU242	User input of 242Pu abundance
C	IZPEAK	Peak location for zero stabilization
C	IZWNDW	Peak window for zero stabilization
C	ZARNG	Analog range for zero stabilization
C	IGPEAK	Peak location for gain stabilization
C	IGWNDW	Peak window for gain stabilization
C	GARNG	Analog range for gain stabilization

C File: GRPLGCM.PMS

C

INCLUDE 'GRPYLG.PMS'

INCLUDE 'GRPGSP.PMS'

INCLUDE 'GRPKCM.PMS'

COMMON BLOCKS

```

C
C   GRPYLG.PMS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   The common blocks used by GRPANL are defined separately in
C   order to guarantee consistency between programs & subroutines.
C   These blocks contain parameters & control info needed by GRPANL and
C   make batch runs possible if properly filled out by EDIGRP.
C   Major rearrangement in 1982 to facilitate use of multiple groups
C   and sharing of space with EVAL matrices. Note most of first block
C   is overlaid by AMTRX in the evaluation phase.
C   Redone for VAX-11 3-Feb-84
C   Edit 6-Feb-87(RHO); 11-Mar-87 Expand #of data pts & free params
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

PARAMETER MXFNM=30
PARAMETER MXDP=1024,MXNP=40
PARAMETER LENLG=214/2

```

```

C   "LENLG" of 107 preserves from NWRDS thru EDGE in EDIGRP allowing
C   storage of the shape constant file name.

```

```

INTEGER*2 NSMBL,NABS,MTLZ !Used for (element symbols)
REAL*8 COFMU

```

```

C   - - - - -
COMMON/YLG/RM(MXDP),WM(MXDP),VM(MXNP),DM(MXNP),
+   Y(MXDP),DELY(MXDP),PKPOS(MXNP),PKHT(MXNP),DEL(MXNP),
+   ERR(7),SUM(MXNP),SUMR(MXNP),
+   NWRDS(2),OFFSET,ALPHA,SHAPC(15),BG(2),AVEBG,GAIN,ZERO,
+   APRGN,APRZRO,REFEN,REFCH,ASLP,DUM2,DUM3,
+   MTLZ(5),CMPOS(5),NABS(6),ABSRB(6),
+   DCNST(24),GEOM,SURFC,SMPWT,DEPTH,NF1,NF2,NSMBL(20),
+   EDGE(20),COFMU(8,20)

```

```

DIMENSION YNET(MXDP)
EQUIVALENCE(Y,YNET)

```

```

INTEGER*2 IDBUF(256)
REAL*4 RLHDR(128)
BYTE HDRBYT(512)
EQUIVALENCE (Y,IDBUF),(HDRBYT,IDBUF),(RLHDR,IDBUF)

```

```

EQUIVALENCE (NWRDS(1),ITER)
DIMENSION CLIN(5)
EQUIVALENCE (SHAPC(11),CLIN(1))
EQUIVALENCE (WM(35),RHO)

```

```

BYTE AFILNM(MXFNM),SFILNM(MXFNM),DFILNM(MXFNM)
EQUIVALENCE(AFILNM,NSMBL),(SFILNM,EDGE),(DFILNM,EDGE(10))

```

```

C   The absorption coefficients are read once at run time,
C   but the file name and many other constants are read in EDIGRP.
C   - - - - -

```

C
C File: GRPGSP.CMN
C
C The file of absorption coefficients is read once at run time,
C but only the file name is stored. Other constants are read in EDIGRP.
C - - - - -
C
C "GRPGSP.FTN" Last Edit for RT-11 2-Jun-83
C - - - - -
C "GSP" contains parameters associated with a spectrum.
LOGICAL*1 ICFIL,IFNAME,LIBNAM,BKGDFL
CHARACTER*30 STGFIL
CHARACTER DSNAME*6,SAMPID*12,EXPTID*12,TBUF*8,DTE*10
REAL*4 LIVTIM
REAL*8 DSTTIM,DTZERO
C PARAMETER LENGSP=121
C LENGSP of 121 preserves GSP thru OLSTD2 (in EDIGRP), as long
C as MXFNM=30

COMMON/GSP/LIVTIM,DSTTIM,DTZERO,DELTAT,DEDTIM,TIMEL,WT100,
1 ANSFCT,LDUMY,KALPFG,IOFLG(6),DSNAME,SAMPID,EXPTID,
2 ICFIL(40),IFNAME(40),BKGDFL(30),LIBNAM(30),
3 STGFIL,IZYR,ICTYR,OLSTD1,OLSTD2,
4 TBUF,DTE,KTGRP,INPDEV,NDEV,INDEV

COMMON BLOCKS

```
C
C  GRPGMS.PMS
C
C  Arrays produced in OUT for SVPKS/GEVAL
C      Last edit for VAX 3-Feb-84
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
PARAMETER MXBKP=100
PARAMETER MXPKS=200
BYTE BKFLGS
```

```
COMMON/GAM/CHNL(MXPKS), ENGY(MXPKS), GAMS(MXPKS), PCTER(MXPKS),
+  CNTS(MXPKS), PKS(MXPKS), PKE(MXPKS), NGAMS, BKFLGS(MXPKS),
+  NBKPKS, BKPKEN(MXBKP), BKPKCT(MXBKP)
```

```
DIMENSION FR CER(MXPKS)
EQUIVALENCE (PCTER, FR CER)
```

```

C
C   GRPKCM.PMS
C
C   Block GPKS carries peak info needed in GRPANL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C   This block is split off to facilitate use of multiple groupings.
C   Edit by J.B.Niday for VAX 5-Dec-1985 (Remove STRJ)
C       Last edit 26-Feb-1986 (adding to BK specs)
C   Edit 11-Mar-87 to increase allowable free parameters
C   Edit 17-June-87 for TSHP9
C   Edit 13-Oct-87 for FIXFLGs
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

PARAMETER MXFP=40
PARAMETER LENPKS=7*MXNP+51           !Define length of block

BYTE MFLGS,NGAIN,NUSHPS,INTFLG,NFLG,FIXFLG2,FIXFLG4
BYTE MCHBUF(32),TXTBUF(70)
REAL*8 DSTPTM

COMMON/GPKS/NXFLG(MXNP),NYFLG(MXNP),EN(MXNP),ENJ(MXNP),
1  HIGHT(MXNP),GAMA(MXNP),STDUMY(MXNP),STD(2),
2  MAXITR,NUMS,NUME,IST,IEND,NDPTS,NP,
3  ATAIL,BTAIL,FWHM,EXP1,EXP2,EXP3,EXP4,SLP(2),EST,EEND,BKLO,BKHI,
4  TSHP5,TSHP8,MFLGS(4),NFLG(6),FIXFLG2,FIXFLG4,RBKEN(2),RBKWID(2),
5  XTRAPL,XTRAPH,DUMY(18)

EQUIVALENCE(MFLGS(1),NGAIN),(MFLGS(2),NUSHPS),(MFLGS(3),INTFLG)
CC Note: MFLGS(4) is used as a graphics flag.
CCC Note that "GPKS" is not used after last call to OUT.
CC Stop time, source machine, and text buffers are saved only until INITLP.
EQUIVALENCE(NXFLG,MCHBUF),(NXFLG(9),TXTBUF)
EQUIVALENCE(EN,DSTPTM),(EN(4),ISTPYR)

```


COMMON BLOCKS

```

C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  GRPEVC.PMS
C
C  Common blocks for EVAL section of GRPANL & GEVAL
C    Last edit for VAX 13-Apr-84
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C  C
C  Current library allocation is for 20 nuclides and 100
C  gamma rays.

```

```

PARAMETER MXNUC=20,MXGAM=100
PARAMETER MXCMP=20
INTEGER*2 ISMBL

```

```

COMMON/LIB/NBRL(2),NMBR1(MXNUC),MASS1(MXNUC),NPRNT(MXNUC),
1  NDGTR(MXNUC),IA1(MXNUC),IA2(MXNUC),INDX1(MXGAM),
2  ENG(MXGAM),BR(MXGAM),T1HLF(MXNUC),ISMBL(100)
EQUIVALENCE (NN1,NBRL(1)),(NN2,NBRL(2))

```

```

COMMON/MTRX/ NCMP,NROW,NCOL,NDXCH(MXPKS),LIST1(MXCMP),
+  LIST2(MXCMP),INDXP(MXCMP),INDX(MXPKS),YGAMS(MXPKS),ANS(MXCMP),
+  FCTR1(MXCMP),INUCL(4),ERROR(MXCMP),
+  BM(MXPKS,MXCMP),AMTRX(MXPKS,MXCMP)

```

C NOTE-- AMTRX not equivalenced for VAX version.(20-Aug-83)

```

DIMENSION PRPK(2)
EQUIVALENCE (OLSTD1,PRPK)
EQUIVALENCE (INPDEV,MGMS),(KTGRP,NSPEC)

```

```
C
C  GRPFL.PMS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C  Common block for transferring peak profiles from FIT to graphics.
C      Last edit for VAX 19-Nov-84 (PRFILS size)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
BYTE PRFNAM,NETFNM,REMFNM
```

```
COMMON/PRFL/PRFILS (MXDP,MXNP),PKPRFL (MXNP),KKB (MXNP),SVYNET (MXDP),
+  ERYNET (MXDP),XEN (MXDP),
+  PRFNAM (MXFNM),NETFNM (MXFNM),REMFNM (MXFNM),KPHE
```

CHAPTER 7

CONTENTS

7. SUBROUTINE DESCRIPTIONS	7- 3
ACQR90 (ACQR38)	7- 4
ADATE	7- 5
AEND	7- 6
ADAC_ERR	7- 7
ASOPT (ASOPTL)	7- 8
ASSAY (ASSAY38)	7- 9
ASYP238	7-10
AUTCYC	7-11
CALEFF	7-12
CALIB	7-13
CALGPM	7-14
CHK90	7-15
CMPLTN (CMPLTL)	7-16
CURDTA	7-17
DATA90	7-18
DATTIM	7-19
DIALOG (DIALGS)	7-20
ERR_HANDLER	7-21
ERRMSG	7-22
EXEC (EXEC238)	7-23
FIT (FIT38)	7-24
FRDSK	7-25
GET_KEY	7-26
GETOPS	7-27
GET90	7-28
HEADER (HEADR38)	7-29
HELP (HELP38)	7-30
INIT90	7-31
INITLP	7-32
INTFLG	7-33
JULDAT (JULDAL)	7-34
LGMENU	7-35
LHEAD	7-36
LMCLOG	7-37
LNMN38	7-38
LODGRP	7-39
LSLOG (LSLOG8)	7-40
MCBIAS	7-41
MCCHK (MCCHK8)	7-42
MCHEAD	7-43
MCPREC	7-44
MENU	7-45
NEWNAM (NEWNM8)	7-46
OUT (OUT38)	7-47
P238GRP	7-48
P238AB	7-49
PASS	7-50
PRECHK	7-51

CONTENTS (CONTINUED)

PRINT1 (PRINTL)	7-52
PUT90	7-53
RDGAMS	7-54
RDHDR	7-55
READ38	7-56
READB	7-57
READC (READC38)	7-58
READNM	7-59
RESET (RESET38)	7-60
ROTMON	7-61
SIZESAMP	7-62
SLEEP	7-63
STABSET	7-64
START	7-65
STATUS	7-66
SVPKS (SVPK38)	7-67
WRITLG (WRTLG8)	7-68
WRTDAT	7-69
WTMCLG	7-70
YESNO	7-71

Chapter 7

SUBROUTINE DESCRIPTIONS

The following pages give, in alphabetical order, descriptions of the subroutines called in the WHC_ASY238 program. Each description includes a listing of the subroutines called by each subroutine in the order they are called. The description also includes a listing of those subroutines that call the subroutine being described.

In some cases the subroutine source filename is different from the subroutine name. In those cases, the source filename is shown in parentheses.

SUBROUTINE DESCRIPTIONS

ACQR90 (ACQR38) This subroutine acquires data from the ADC on the Canberra Series 90 analyzer. It clears the data in the analyzer, sets the acquisition time, starts the ADC, and monitors it for timeout. It also disables the sample LOAD/UNLOAD capability and starts the sample rotation and scanning motions.

Called by: ASSAY
Flags used: LIVE, TKEY, VAXUP, MP, MB, STPCYC, ABORT

ACQR90 Collects the spectral data on the Canberra Series 90 analyzer

BSETW	ADAC call to disable sample LOAD/UNLOAD
BCLRW	ADAC call to ensure sample rotation
CTIM	Canberra subroutine to clear the time
CHK90	Checks CTIM call for error
CDAT	Canberra subroutine to clear the data
CHK90	Checks CDAT call for error
PSET	Sets the acquisition clock time in the analyzer
ROTMON	Checks whether sample is rotating and scanning
CMPLTN	Calculates completion time for acquisition
COLL	Canberra subroutine to start ADC
CHK90	Checks COLL call for error
INQADC	Canberra subroutine to check ADC for timeout
CHK90	Checks INQADC call for error
ROTMON	Checks whether sample is still rotating and scanning
DATTIM	Gets current date and time from system
ADCOFF	Canberra subroutine to stop ADC
CHK90	Checks ADCOFF call for error
BSETW	Calls to ADAC boards to stop rotation/scanning and
BCLRW	enable LOAD/UNLOAD, if assay was aborted
FROMCA	Retrieves live time and clock time from analyzer
CHK90	Checks FROMCA call for error

SUBROUTINE DESCRIPTIONS

ADATE This subroutine sets the date and time that the assay
was started.

Called by: ASOPT, AUTCYC, CALIB, DIALOG, MCBIAS, MCPREC

ADATE Sets date and time assay was started

DATTIM Gets date and time from computer system

SUBROUTINE DESCRIPTIONS

AEND This subroutine stops sample rotation and enables
sample LOAD/UNLOAD at the end of a normal assay.

Called by: ASOPT, AUTCYC, CALIB, MCBIAS, MCPREC
Flags used: TKEY, VAXUP

AEND Stops sample rotation and enables sample LOAD/UNLOAD

BSETW ADAC calls to stop sample rotation and enable sample
BCLRW LOAD/UNLOAD

ADAC_ERR This function routine reports I/O errors encountered
 with ADAC calls

Called by: AEND, ACQR90, LNMN38, PRECHK, ROTMON, START, STATUS

Function name is ERROR_CHECK

No subroutines called.

VMS system calls used.

SUBROUTINE DESCRIPTIONS

ASOPT (ASOPTL) This subroutine is called when the assay option (A) is selected. It makes calls to perform the assay, print the results.

Called by: EXEC
Flags used: ASFLG, ABORT

--

ASOPT	Routine makes calls to perform assays and print results.
MCCHK	Checks measurement control status
ADATE	Gets assay date and time
DIALOG	Receives input from operator
ASSAY	Performs assay
AEND	Stops sample rotation and enables sample UNLOAD/LOAD
PRINT1	Prints results on operator terminal
WRITLG	Writes assay results to assay log file

ASSAY (ASSAY38) This subroutine performs the necessary calls to collect or retrieve spectral data and then analyzes it for normal assays, measurement control, and disk options.

Called by: ASOPT, AUTCYC, CALIB, CURDTA, FRDSK, MCBIAS, MCPREC
Flags used: AC, FROMDK, ABORT

ASSAY	Performs necessary calls for normal assays, measurement control, and disk options
DATA90	Retrieves live times and clock times from data
DATTIM(1)	Gets date and time from system
ACQR90	Acquires data
DATTIM(1)	Gets date and time from system
WRTDAT	Gets data from ADC and writes into a disk file
ASYP238	Analyzes spectral data for plutonium isotopic abundances

SUBROUTINE DESCRIPTIONS

ASYP238 This is a dummy main program to facilitate the
 structure of the WHC_ASY238 program.

ASYP238 Dummy main program

EXEC Main executive subroutine for control of the
 instrument and analysis of the data

AUTCYC This subroutine is called when the AU or AUD options are selected. The subroutine makes calls to perform repeated assays for the number of times indicated in the DIALOG subroutine. If the AU option is selected, assays are performed on the sample loaded in the instrument. If the AUD option is selected, the assays are of data stored on disk.

Called by: EXEC
Flags used: AUTOCY, ASFLG, FROMDK, ABORT

AUTCYC	Performs a series of assays. If FROMDK is set to TRUE, analyses are of data stored on disk
MCCHK	Checks measurement control status
DIALOG	Receives input from operator
ADATE	Gets assay date and time
ASSAY	Makes calls to perform assay
AEND	Stops sample rotation and enables sample UNLOAD/LOAD
WRITLG	Writes results to assay log file
PRINT1	Prints results of assay
NEWNAM	Creates new data filename by incrementing filename extension; i.e., IO1 becomes IO2
HEADER	Prints summary of analysis

SUBROUTINE DESCRIPTIONS

CALEFF

This subroutine calculates detector efficiency for a given energy based on polynomial coefficients read from the PU238.DET file.

Called by:

READ38

No subroutines are called.

CALIB This subroutine is used to calculate the gain and zero using reference peaks specified by the PU238.CTL file. This subroutine is called once for each peak grouping.

Called by: P238GRP

CALIB	Performs spectrum calibration, i.e. gain and zero
RDCHNS	Reads in spectral data for reference peaks. Subroutine contained in PULIB.
MAXVAL	Finds channel with maximum data in region. Subroutine contained in PULIB.
MINVAL	Finds channel with minimum data in region. Subroutine contained in PULIB.
AVE	Determines average number of counts in specified region. Subroutine contained in PULIB.
GFIT	Fits a Gaussian to the top of the given peak data to determine the peak centroid and height. Subroutine contained in PULIB.

SUBROUTINE DESCRIPTIONS

CALGPM This subroutine calculates gammas per minute from the sample by correcting the observed gamma-ray peak intensity for sample self-absorption, absorption between sample and detector, geometry, and detector efficiency.

Called by: OUT

CALGPM Calculates gammas per minute from the sample.

CALMU Calculates mass absorption coefficients. Subroutine contained in PULIB.

GPM Calculates geometry and detector efficiency correction factors. Subroutine contained in PULIB.

CHK90 This subroutine checks error return on subroutine
calls to the Canberra Series 90 analyzer. An error
has occurred if the error variable returned in the
call is less than zero.

Called by: ACQR90, DATA90, GET90, INIT90, PRECHK, PUT90

No subroutines are called.

SUBROUTINE DESCRIPTIONS

CMPLTN (CMPLTL) This subroutine calculates and prints out the completion date and time for the assay, given the current time and the preset clock time for the assay. It also gets the start time for the assay.

Called by: ACQR90

CMPLTN Calculates completion time for the current acquisition

DATTIM(1) Gets date and time from system

JULDAT Converts ASCII time and date into Julian date

CURDTA This subroutine is called when the Assay Current data
in the analyzer option (AC) is selected.

Called by: EXEC
Flags used: AC, ABORT

CURDTA Calls subroutines necessary to analyze the data
currently stored in the analyzer

DIALOG Receives input from the operator

ASSAY Calls routines to perform the analysis

PRINT1 Prints results of the assay

SUBROUTINE DESCRIPTIONS

DATA90 This subroutine returns spectral data from the Canberra Series 90 analyzer or disk file for a specified region of interest.

Called by: ASSAY
Flags used: FROMDK

DATA90 Returns data from analyzer or disk file for a specified region

FROMCA Canberra subroutine to read data from the analyzer in a specified region of the spectrum

CHK90 Checks FROMCA call for errors on return

DATTIM This subroutine retrieves, verifies, or changes the system date and time. Four different options are available in calling this subroutine:

 Option 0: Gets date and time from operator input and gives them to system.

 Option 1: Gets date and time from system computer.

 Option 2: Gets date and time from system and lets operator verify or change them.

Called by: ACQR90, ADATE, ASSAY, CALIBS, CMPLTN, EXEC, GETOP, MCHEAD

DATTIM Retrieves, changes, or verifies the system date and time

JULDAT Converts ASCII date and time to Julian date

MicroVMS system calls are used to set the system date and time.

SUBROUTINE DESCRIPTIONS

DIALOG (DIALGS) This subroutine receives input from the operator for various assay parameters or conditions.

Called by: ASOPT, AUTCYC, CALIB, CURDTA, FRDSK, MCBIAS, MCPREC
Flags used: ABORT, AC, WRITNG, FROMDK, AUTOCY, AUD

DIALOG	Receives operator input
GETOPS	Gets operator ID and sample name
ADATE	Gets time and date
NEWNAM(0)	Creates filename for spectral data from sample name
PRECHK	Preliminary check of loaded sample for count rate/detector positioning
ADATE	Gets time and date of assay

SUBROUTINE DESCRIPTIONS

ERR_HANDLER This subroutine detects math errors (e.g., division by zero, square root of a negative number, logarithm of zero or a negative number).

Called by: P238AB

Uses MicroVMS system calls for error detection.

SUBROUTINE DESCRIPTIONS

ERRMSG This subroutine reads appropriate error message from the error message file (ERRMSG.INS) and prints it on the operator terminal. Error messages are numbered in the error message file, and the ERRMSG call requires the message number.

Called by: ASSAY, EXEC, GET90, LMCLOG, LSLOGG, MCBIAS, MCCHKS, MCPREC, MENU, PUT90, READB, READC, READNM, WRTLGG, WTMCLG

No subroutines are called.

EXEC (EXEC238) This subroutine is the main executive subroutine for the Pu-238 isotopic analysis instrument which controls the overall program flow.

Called by: ASYP238

EXEC (EXEC238) Executive subroutine that controls the overall program flow

START Initializes the system and control flags

RESET(0) Gets default constants from constants file

INTFLG Initializes logical flags

DATTIM(1) Gets system date and time

GET_KEY Allows single character keyboard entry for menu options

LNMN38 Checks liquid nitrogen level and monitors status

STATUS(0) Reports status of hardware

MENU Searches 'MENU.TXT' table to locate option entered by operator

ERRMSG Prints error message on menu selection

PASS Checks password for supervisor functions

SUBROUTINE DESCRIPTIONS

FIT (FIT38) This subroutine performs peak fitting to a single peak or a grouping of peaks peak by the method of least squares. It determines the peak position(s), height(s), and full-width-half-maximum(s) of a gamma-ray peak(s).

Called by: P238GRP
Flags used: ABORT, LONGPR

FIT (FIT38) Subroutine to fit gamma-ray peak(s).

BWF Computes x-ray profile. Subroutine contained in PULIB

AVE Determines average number of counts in specified region. Subroutine contained in PULIB.

MATINV Subroutine to invert matrix to solve for unknowns in the fitting process

FRDSK This subroutine is selected by the Assay Disk (AD) option. The subroutine performs assays on spectral data stored on disk.

Called by: EXEC
Flags used: FROMDK, AD, ABORT

FRDSK	Subroutine to analyze spectral data stored on disk
DIALOG	Receives input from operator
ASSAY	Calls routines to perform analysis
PRINT1	Prints results of assay

SUBROUTINE DESCRIPTIONS

GET_KEY This subroutine gets keyboard input from a virtual keyboard buffer. It allows single-character entry for menu options.

Called by: EXEC

MicroVMS system calls are used.

GETOPS This subroutine gets the operator ID and sample ID
 from the operator.

Called by: DIALOG

Flags used: ABORT, AC, FROMDK, MP, MB, MTL

GETOPS Gets operator and sample ID

SUBROUTINE DESCRIPTIONS

GET90 This subroutine gets a spectrum from the Canberra Series 90 analyzer and writes it into a disk file.

Called by: EXEC, WRTDAT
Flags used: SMALL, WRITNG

GET90 Gets spectral data from the analyzer and writes to disk

ERRMSG Prints error message

FROMCA Canberra subroutine to get data from the analyzer into the computer

CHK90 Checks FROMCA call for errors on return

HEADER (HEADR38) This subroutine prints an assay header after an assay or summary header after a series of assays. The header printout identifies the type of assay, cycle number, data filename, operator ID, constants filename, and assay date and time.

Called by: AUTCYC, CALIB, MCPREC, PRINT1
Flags used: AUTOCY, LIVE, FROMDK

No subroutines are called

SUBROUTINE DESCRIPTIONS

HELP (HELP38) This subroutine is called when the Help (H) or Help Supervisor (HS) options are selected. The H option prints a list of menu options for the operator on the operator terminal. The HS option also prints the supervisor menu.

Called by: EXEC

Flags used: SUPV

No subroutines are called.

INIT90 This subroutine initializes the Canberra Series 90 analyzer. The Canberra analyzer must be set to "ON LINE" for this subroutine to complete initialization of the analyzer.

Called by: START

INIT90 Initializes Canberra Series 90 analyzer

INIT Canberra subroutine to initialize the analyzer

CHK90 Check INIT call for errors on return

SUBROUTINE DESCRIPTIONS

INITLP This subroutine begins output from the data-analysis if a long printout is requested. This part of the output gives analysis parameters and file names.

Called by: READ38

INITLP Begins output from data analysis for long printout

No subroutines are called.

INTFLG This subroutine initializes logical flags used in the control of the instrument and in the analysis of the data. This subroutine is called when the system is first initialized and after each option has been completed.

Called by: EXEC, RESET, START
Flags used: ABORT, BACKG, AUTOCY, FROMDK, OLDDATA, CUMLTV, STPCYC,
 ASFLG, AC, AUD, AD, MB, MP, TKEY, CHANG, CHANG1, BAD,
 BD, CONDIR, CAL, CALDSK, DGFLG, MCERR

No subroutines are called.

SUBROUTINE DESCRIPTIONS

JULDAT (JULDAL)

This subroutine calculates the Julian date from ASCII strings for the date and time. The LLNL version of the subroutine calculates the Julian date from 1/1/1900. The calling arguments are

IDAT - ASCII string for date, i.e., 30-APR-85

ITIM - ASCII string for time, i.e., 11:45:10

The output arguments are

XDAT - Julian day

XTIM - Time of day in seconds

Called by: CMPLTN, DATTIM, LSLOG, LMCLOG, MCCHK

No subroutines are called.

LGMENU

This subroutine lists menu options for the list assay log and lists measurement control log options. The menu options are:

Option 0: Return to main menu.

Option 1: List entries not yet sent to the VAX. (NOT USED)

Option 2: List entries between two dates.

Option 3: List all entries.

Option 4: List n entries.

Called by:

LSLOG, LMCLOG

No subroutines are called.

SUBROUTINE DESCRIPTIONS

LHEAD This subroutine prints the header information for the
list assay log option.

Called by: LSLOG

LHEAD Prints header for listing of assay log

DATTIM(1) Gets date and time from system

LMCLOG This subroutine is called when the List Measurement Control (LM) option is selected.

Called by: EXEC
Flags used: MCVAX

LMCLOG Lists measurement control entries.

 LGMENU Selects listing option

 JULDAT Converts ASCII date and time to Julian date

 MCHEAD Prints measurement control log header

SUBROUTINE DESCRIPTIONS

LNMN38 This subroutine checks the signals from the liquid-nitrogen-level monitor (ORTEC 729A). The signal is interfaced to the computer through an ADAC 1616 CCI board. The subroutine checks bits #5 device ZWA1. If it goes low, this indicates a low liquid-nitrogen level in the detector.

Called by: EXEC
Flags used: ABORT

LNMN38	Checks liquid nitrogen level monitors
BTSTW	ADAC subroutine to test bit setting of device
ERROR_CHECK	Tests for I/O in ADAC subroutine call

LODGRP This subroutine loads a specified peak grouping and removes the background counts from the region. It also sets up any specified energy & intensity relationships between peaks. For single gamma-ray peaks specified for simple integration only, the peak sum and centroid are determined.

Called by: P238GRP

LODGRP	Loads a specified peak region, subtracts background, and integrates single peaks, if specified.
RDCHNS	Reads specified portion of spectrum from spectral data file. Subroutine contained in PULIB
AVE	Gets average number of counts in specified region of the spectral data. Subroutine contained in PULIB
BKGRD	Subtracts background from specified region giving net counts for the region. Subroutine contained in PULIB
MAXVAL	Determines channel with maximum counts in specified region. Subroutine contained in PULIB
GFIT	Fits a Gaussian to the top of the given peak data to determine the peak centroid and height. Subroutine contained in PULIB

SUBROUTINE DESCRIPTIONS

LSLOG (LSLOG8) This subroutine is called when the List Assay log (LA) option is selected.

Called by: EXEC
Flags used: AUTO, LVFLAG

LSLOG Lists the assay log entries.

LGMENU Selects listing option

JULDAT Converts ASCII date and time into Julian date

LHEAD Prints assay log header

ERRMSG Report error in handling log file

MCBIAS This subroutine is called when the measurement bias (MB) option is selected. The subroutine performs a measurement control assay to check instrument bias. For the Pu-238 isotopic analysis instrument, this requires that a specific standard sample be measured by the instrument.

Called by: EXEC
Flags used: MB, BIASOK, MCERR, MCVAX

MCBIAS	Performs measurement control check for bias using a standard sample. The specific power result is checked against the known value for that standard
DIALOG	Receives input from operator
ADATE	Sets the assay date and time
ASSAY	Performs assay
ERRMSG	Prints error message if bias results exceed warning or action limits
PRINT1	Prints results of bias measurement on operator terminal
YESNO	Queries user for an additional measurement if the first measurement did not meet specifications
AEND	Stops sample rotation and enables sample UNLOAD/LOAD
WTMCLG	Writes measurement results to measurement control log file

SUBROUTINE DESCRIPTIONS

MCCHK (MCCHKS) This subroutine checks that measurement control has been performed within the required time limits and also checks the status of the measurement control flag. These checks are made before an assay or an autocycle run. For the Pu-238 isotopic analysis instrument, the measurement control consists of precision and bias checks. The time limit for each is read in from the constants file and can be changed with the PU238_CHNG program.

Called by: ASOPT, AUTCYC
Flags used: BIASOK, PRECOK, BACKOK, ABORT, MCEFLG

MCCHKS	Checks measurement control status
ERRMSG	Prints error message if problem occurred in opening the measurement control log file
JULDAT	Converts ASCII date and time to Julian date
YESNO	Queries user for yes/no response

MCHEAD This subroutine prints header information for the
 measurement control log listing.

Called by: LMCLOG

MCHEAD Prints measurement control log listing header

DATTIM(1) Gets date and time from system

SUBROUTINE DESCRIPTIONS

MCPREC This subroutine is selected by the measurement precision (MP) option. The subroutine performs a precision measurement control check on the instrument. It performs a series of replicate assays (5 or 15). The number of assays is determined from the constants file, which may be changed by the supervisor by using the PU238_CHNG program.

Called by: EXEC
Flags used: MP, ABORT, PRECOK, MCERR, MCVAX

MCPREC	Measurement control precision check. Performs 5 or 15 replicate analyses
DIALOG	Receives input from operator
ADATE	Sets the assay date and time
ASSAY	Performs assay
NEWNAM	Resets filename extension
PRINT1	Prints results of assay on operator terminal
ERRMSG	Prints error message if precision results exceed warning or action limits
HEADER	Prints header for summary of precision assay results
AEND	Stops sample rotation and enables sample UNLOAD/LOAD
WTMCLG	Writes precision results to measurement control log file

MENU This subroutine searches the table in the file
MENU.TXT to locate a match with the option entered by
the operator. If no match is found, a flag NOPT is
set to TRUE, and the subroutine returns.

Called by: EXEC
Flags used: ABORT

MENU Searches for option entry in menu table

ERRMSG Prints error message if MENU.TXT file cannot be read

SUBROUTINE DESCRIPTIONS

NEWNAM (NEWNM8)

This subroutine creates a new filename for an assay or autocycle run. If measurement is a single assay, then filename is created with WH8 as the extension for the spectrum data file. If an autocycle is run and called with OPTION=0, the filename extension is IO1. If OPTION=1, the filename extension is updated to IO2, IO3, etc.

Called by:

AUTCYC, DIALOG, MCPREC, MCBIAS

No subroutines are called.

OUT (OUT38) Subroutine to accumulate the results of the peak
fitting, calculate photon emission rates of the
source, tabulate data and results, and store them.

Called by: P238GRP
Flags used: LONGPR

OUT (OUT38) Tabulate results of data analysis

CALGPM Correct gamma-ray intensities for absorption and
detector efficiency

SUBROUTINE DESCRIPTIONS

P238GRP This subroutine is the executive subroutine for the data-analysis part of the Pu-238 isotopic analysis instrument program. It directs the sequence of calls to other subroutines required for analyzing the data.

Called by: ASSAY
Flags used: MP, MB, ABORT

P238GRP	Executive subroutine for data analysis
READ38	Read analysis control information and open spectral data file
RDGRP	Read in analysis specifications for a peak grouping (FORTRAN source found in READ38 source file)
CALIB	Determine spectrum calibration from specified reference peaks
LODGRP	Load spectral data for specified peak grouping and begin analysis
FIT	Fit specified gamma-ray peaks to spectral data for peak intensity information
OUT	Summarize results of peak fitting, calculate photon emission rates, tabulate data and results, and store them
SVPKS	Write gamma-ray emission results to a disk file
P238AB	Analyze gamma-ray emission rates to determine relative plutonium isotopic abundances

P238AB

This routine uses the gammas/min emission rates determined by the data analysis and corrected for Cd, neutron shielding, and EP 60/61 container absorption to determine the relative abundances of the plutonium isotopes in the sample. It estimates the thickness (g/cm^2) of the plutonium sample, and small changes in the shape of the detector efficiency curve from that of a nominal 20% coaxial germanium detector. The calculations are made using the measured intensities of the following 12 peaks: 152, 208, 238.6, 375., 376.6, 413.7, 582.9, 727.4, 742.8, 766.4, 786.3, and 860.5 keV. The equations involve 8 unknowns: the dpm's of plutonium-238, -239 and -241, americium-241 if present, Th-228, the plutonium sample thickness, the change in the slope of the efficiency curve, and the change in the curvature of the efficiency curve. The equations are nonlinear in form, requiring an iterative approach to their solution. The following boundary constraint is applied:

$$0.002 < \text{Pu } (\text{g}/\text{cm}^2) < 40.$$

Called by:

P238GRP

Flags used:

LONGPR, MB, MP

P238AB

Calculate the relative isotopic abundances of plutonium from the observed gamma-ray emission rates

RDGAMS

Read in gamma-ray emission rates from PU238.RST file

MATINV

Invert matrix to solve for unknowns in the nonlinear least-squares process

SUBROUTINE DESCRIPTIONS

PASS

This subroutine checks the password given by the operator in order to allow the operator to enter the supervisory mode. The passwords are contained in the BACKGR.WH8 file and can be changed with the PU238_CHNG program. If the password is matched, the logical variable OK is set to TRUE.

Called by: EXEC

No subroutines are called.

PRECHK This subroutine checks to make sure that the sample is loaded properly, rotating, and moving up and down. It also determines the sample size and performs a 60-second data acquisition to determine the counting time required for 1% precision on the specific power.

Called by: DIALOG

PRECHK	Performs preliminary check on sample counting conditions
BTSTW	ADAC subroutine call to check status of bits that indicate whether sample lid is open/closed and start sample rotation
ERROR_CHECK	Checks for I/O error in ADAC call
SLEEP	Wait before testing
ROTMON	Checks whether sample is rotating; moving up and down
CTIM	Canberra subroutine to clear the time
CHK90	Checks CTIM call for error
CDAT	Canberra subroutine to clear the data
CHK90	Checks CDAT call for error
PSET	Sets the acquisition clock time in the analyzer
COLL	Canberra subroutine to start the ADCS
CHK90	Checks COLL call for error
INQADC	Canberra subroutine to check for ADC timeout
CHK90	Checks INQADC call for error
SIZESAMP	Determines whether sample is an EP60/61 or FCC container by sensing its size.
ROTMON	Checks whether sample is rotating; moving up and down
FROMCA	Retrieves live time, clock time, and 152-keV peak data from analyzer
CHK90	Checks FROMCA call for error
MAXVAL	Finds channel with maximum data in 152-keV region. Subroutine contained in PULIB.
NCNTS	Removes background counts from specified region and integrates net data for total net counts

SUBROUTINE DESCRIPTIONS

PRINT1 (PRINTL) This subroutine prints the final results of the analysis on the operator terminal.

Called by: ASOPT, AUTCYC, CALIB, CURDTA, FRDSK, MCBIAS, MCPREC

PRINT1 Prints final results of the assay on operator terminal

HEADER Prints header containing assay information

PUT90 This subroutine reads a spectrum from disk and sends it to the Canberra Series 90 analyzer. The data read from disk are specified by the file name stored in FILNAM. The data are read and sent to the analyzer in 128-channel chunks.

Called by: EXEC
Flags used: ABORT, LIVE, NOBACK

PUT90 Reads spectral data from disk and sends them to the Series 90 analyzer memory

ERRMSG Prints error message if problems occur in reading the spectral data file

OPEN80 Canberra subroutine that opens the ADC memory in the analyzer for communication

CHK90 Checks OPEN80 call for errors on return

WRIT80 Canberra subroutine writes data into the analyzer ADC memory

CHK90 Checks WRIT80 call for errors on return

CLOS80 Canberra subroutine closes communication channel to analyzer ADC memory

CHK90 Checks CLOS80 call for errors on return

SUBROUTINE DESCRIPTIONS

RDGAMS This subroutine reads in the gamma-ray intensity information from all of the peak groupings in preparation for the analysis to determine the relative plutonium isotopic abundances. This information is read in from the PU238.RST file.

Called by: P238AB

RDGAMS Reads in gamma-ray emission rate information

No subroutines are called.

RDHDR This subroutine reads in the PU238.SMP file for the
 spectrum gain and zero, and sample absorber
 information.

Called by: READ38

RDHDR Reads in sample and system setup information

No subroutines are called.

SUBROUTINE DESCRIPTIONS

READ38 This subroutine reads the files required by the data analysis. It reads the analysis control file, PU238.CTL, the spectral data file, the detector-constants file, PU238.DET, and the peak-shape constants file, PU238.SHP.

Called by: P238GRP

READ38	Reads necessary files for data analysis
RDHDR	Read measurement information from PU238.SMP
CALEFF	Calculate detector efficiency
INITLP	Begin output for long printout

READB This subroutine reads instrument detector background information from disk. This information is contained in the BACKGR.WH8 file. For the Pu-238 isotopic assay instrument, this file is not used for background information. It is used only for storing the passwords.

Called by: RESET
Flags used: FROMDK, BEGIN, ABORT

READB Reads BACKGR.WH8 file

ERRMSG Prints error message if problems occur in reading the file

SUBROUTINE DESCRIPTIONS

READC (READC38) This subroutine reads the constants file PARMTR.WH8
for the Pu-238 isotopic assay instrument from disk.

Called by: RESET
Flags used: BEGIN

READC (READC38) Reads instrument constants file that contains assay
and measurement control information

ERRMSG Prints error message if problems occur in reading the
files

SUBROUTINE DESCRIPTIONS

READNM This subroutine reads the CONSTA.WH8 file that contains five constants filenames and allows the operator to select or change them.

Called by: RESET
Flags used: NEWFIL, BEGIN, ABORT

READNM Reads constants filenames from CONSTA.WH8 disk file and allows operator to change or select

ERRMSG Prints error message if error occurred in reading CONSTA.WH8 file or the constants file itself

YESNO Queries operator for a yes/no response

SUBROUTINE DESCRIPTIONS

RESET (RESET38) This subroutine resets the default assay parameters or permits the operator to change default flag settings or enter new assay parameters. This subroutine is called if the Default (D) option is selected, which allows the user to change default flags and parameters.

Called by: EXEC
Flags used: ABORT, BEGIN

RESET(1) If called with calling argument equal to 1, the operator is permitted to change various current or default flags or constants. Flags or constants are:

1. Long or short printout flag
2. Permanent or temporary storage of spectrum file on disk flag
3. Preset assay time (clock seconds)
4. Pu-240 abundance in weight %. If value is 0, the Pu-240 abundance will be calculated by the data analysis code.
5. Default parameter file name

READNM Reads parameter file names if option 5 is selected

READC Reads parameter file if option 5 is selected

.... OR

RESET(0) If the subroutine is called with the calling argument equal to 0, the following subroutines are called to reset the system:.

READNM Reads in parameter file names from the CONSTA.WH8 file

READC Reads parameter file for default assay values

READB Reads background file. For Pu-238 isotopic analysis instrument, this file is read only to get passwords

INTFLG Initializes flags

ROTMON This subroutine monitors the sample rotation sensor signal for a not-rotating condition. The rotation-sensor signal is interfaced to the computer by the ADAC 1616 CCI board. Bit #4 in the ADAC 1616 CCI data register is set if the sample stops rotating. If the bit is set, the logical flag RFLG is set to TRUE and returned to the subroutine ACQR90, which suspends the measurement.

Called by: ACQR90, PRECHK

ROTMON	Monitors the rotation-sensor signal
BTSTW	ADAC subroutine to test status of bit
ERROR_CHECK	Checks ADAC call for I/O error

SUBROUTINE DESCRIPTIONS

SIZESAMP This subroutine determines the sample type by sensing
 ists size. The sample-height sensor signal is
 interfaced to the computer by the ADAC 1616 CCI board.
 Bit #2 in the ADAC 1616 CCI data register is set if
 the sample is an EP60/61 container. If the bit is
 set, the logiccal variable MATYP is set to EP6
 otherwise it is set to FCC.

Called by: PRECHK

SIZESAMP Monitors the sample-height sensor signal

BTSTW ADAC subroutine to test status of bit

ERROR_CHECK Checks ADAC call for I/O error

SUBROUTINE DESCRIPTIONS

SLEEP This subroutine provides a pause in the program for
the number of seconds specified in the call.

Called by: ACQR90, PRECHK

No subroutines are called.

VMS system calls used.

SUBROUTINE DESCRIPTIONS

STABSET This subroutine initializes the Canberra 8232 stabilizer by setting its zero and gain peak positions and windows and analog ranges. Communication with the stabilizer is via a serial line from MicroVAX II port TTA1:.

Called by: EXEC (EXEC238)

STABSET Initializes the Canberra stabilizer

 Uses subroutines in 8232lib

 VMS system calls are used.

START This subroutine initializes the program, the hardware, and the system. It allocates and assigns I/O channels to the ADAC devices and clears the sample UNLOAD/LOAD setting.

Called by: EXEC
Flags used: BEGIN, ABORT

START	Initializes the instrument
INTFLG	Initializes all necessary logical flags
BSETW BCLRW	ADAC subroutines to set and clear bits that enable the sample LOAD/UNLOAD capability
ERROR_CHECK	Checks ADAC subroutine call for I/O error
INIT90	Initializes the Canberra Series 90 analyzer
STABSET	Sets up the Canberra 8232 stabilizer

VMS system calls are used.

SUBROUTINE DESCRIPTIONS

STATUS This subroutine is called when the Status (S) option is selected. When called in this manner, the hardware busy status is checked. When called directly by the EXEC subroutine, the hardware is checked for failures.

Called by: EXEC

STATUS Checks the status of the analyzer and sample control/monitor hardware

DINW ADAC subroutine to read sample control/monitor hardware status

ERROR_CHECK Checks ADAC subroutine call for I/O error

INQADC Canberra subroutine to determine if ADCs are busy or idle

SVPKS (SVPK38) This subroutine saves the results of the gamma-ray
peak analysis in a disk file, PU238.RST, for the final
analysis later by P238AB.

Called by: P238GRP

SVPKS (SVPK38) Save results of peak analysis in a disk file

No subroutines called.

SUBROUTINE DESCRIPTIONS

WRITLG (WRTLG8)

This subroutine writes the most recent assay information into the assay log file.

Called by:

ASOPT, AUTCYC, CALIB

Flags used:

WRITNG

WRITLG

Writes assay results into assay log file

ERRMSG

Prints an error message if problems occur in opening the assay log file

SUBROUTINE DESCRIPTIONS

WRTDAT This subroutine calls the subroutine that retrieves
the data from the analyzer and writes the data into a
disk file.

Called by: ASSAY

WRTDAT Calls routine to get spectral data from analyzer and
writes to disk

GET90 Gets spectral data from analyzer and writes to disk

SUBROUTINE DESCRIPTIONS

WTMCLG This subroutine writes the most recent measurement control results into the measurement control log file.

Called by: MCBIAS, MCPREC

WTMCLG Writes measurement control results into the measurement control log file

ERRMSG Prints error message if problems occur in opening the measurement control log file

SUBROUTINE DESCRIPTIONS

YESNO This subroutine prompts for and accepts a yes-or-no answer to a question passed in the call to the subroutine. Sixty characters are allowed in the argument passed.

Called by: DIALOG, GETOPS, LMCLOG, LSLOG, MCBIAS, MCCHK, MCPREC, READNM

No subroutines are called.

CHAPTER 8

CONTENTS

8. PROGRAM FLOW	8- 3
Definitions of subroutines called	8- 7

Chapter 8

PROGRAM FLOW

The sequence in which subroutines are called by the executive subroutine in the WHC_ASY238 program is shown below. This information is also contained within the file WHC_ASY238.DOC on the system disk. Also included in this file are brief descriptions of each of the subroutines. If the subroutine source filename differs from the subroutine name, the filename is shown in parentheses.

WHC_ASY238.DOC
22-MAY-91 WDR

WHC_ASY238 PROGRAM SUBROUTINE CALLS
WITH Pu-238 ISOTOPIC ANALYSIS SPECIFIC ROUTINES

This document may be found in the file WHC_ASY238.DOC. Any changes in the WHC_ASY238 program affecting the information shown here should be indicated in this document.

This diagram shows only the order in which subroutines are called by the WHC_ASY238 program. It does not show the actual program flow.

In those cases where the subroutine name is different from the source (.FOR) filename, the filename is shown in parentheses.

```
EXEC .....(EXEC238)
  START
    INTFLG
    INIT90
    CHK90
  RESET .....(RESET38)
    READNM
      ERRMSG
    READC .....(READC38)
      ERRMSG
    READB
      ERRMSG
    INTFLG
```

PROGRAM FLOW

```

STABSET
DATTIM
  JULDAT ..... (JULDAL)
STATUS
INTFLG
DATTIM
LNMN38
MENU
  ERRMSG
ERRMSG
ASOPT ..... (ASOPTL)
  MCCHK ..... (MCCHKs)
  ADATE
    DATTIM
  DIALOG ..... (DIALGS)
    GETOPS
    ADATE
      DATTIM
    DATTIM
    NEWNAM ..... (NEWNM8)
    PRECHK
      ROTMON
      SIZESAMP
      MAXVAL
      NCNTS
  ASSAY ..... (ASSAY38)
    ERRMSG
    DATA90
    CHK90
    DATTIM
    ACQR90 ..... (ACQR38)
    CHK90
    CMPLTN ..... (CMPLTL)
      DATTIM
      JULDAT ..... (JULDAL)
    ROTMON
    KPEEK
    DATTIM
    SLEEP
  WRTDAT
  P238GRP
  READ38
  CALIB
  LODGRP
  FIT ..... (FIT38)
  OUT ..... (OUT38)
  SVPKS ..... (SVPK38)
  P238AB
AEND
PRINT1 ..... (PRINTL)
  HEADER ..... (HEADR38)
  WRITLG ..... (WRTLGS)
  ERRMSG

```

```

MCBIAS
  DIALOG      ..... (DIALGS)
  ADATE
  ASSAY      ..... (ASSAY38)
  ERRMSG
  PRINT1     ..... (PRINTL)
  AEND
  WTMCLG
MCPREC
  DIALOG      ..... (DIALGS)
  ADATE
  ASSAY      ..... (ASSAY38)
  PRINT1     ..... (PRINTL)
  ERRMSG
  HEADER     ..... (HEADR38)
  AEND
  WTMCLG
PASS
CURDTA
  DIALOG      ..... (DIALGS)
  ASSAY      ..... (ASSAY38)
  PRINT1     ..... (PRINTL)
FRDSK
  DIALOG      ..... (DIALGS)
  ASSAY      ..... (ASSAY38)
  PRINT1     ..... (PRINTL)
AUTCYC
  MCCHK      ..... (MCCHK8)
  DIALOG      ..... (DIALGS)
  ADATE
  ASSAY      ..... (ASSAY38)
  AEND
  WRITLG     ..... (WRTL8)
  PRINT1     ..... (PRINTL)
  NEWNAM     ..... (NEWM8)
  HEADER     ..... (HEADR38)
RESET
  ..... (RESET38)
  READNM
  READC      ..... (READC38)
  READB
  INTFLG
LSLOG
  ..... (LSLOG8)
  ERRMSG
  LGMENU
    JULDAT    ..... (JULDAL)
  LHEAD
    DATTIM
LMCLOG
  ERRMSG
  LGMENU
    JULDAT    ..... (JULDAL)
  MCHEAD
    DATTIM
STATUS

```

PROGRAM FLOW

PUT90
GET90
HELP (HELP38)
LSLOG (LSLOG8)

DEFINITIONS OF SUBROUTINES CALLED

Subroutine	Definition
ACQR90 (ACQR38)	Acquires data on the Canberra Series 90
ADATE	Set the date and time for the assay
ADAC_ERR	Checks ADAC subroutine call for I/O error
AEND	Stops sample rotation/scanning and enables sample UNLOAD/LOAD.
ASOPT (ASOCTL)	Responds to the assay option when selected by the operator from the menu. Contains calls which perform the assay, prints results, and writes data to files
ASSAY (ASSAY38)	Performs an assay run or analysis of data files on disk for assay and measurement control. It then calls the analysis routines and the data diagnostic routine.
ASYP238	Dummy main program to facilitate structure of program
AUTCYC	Responds to the operator request for assay in autocycle mode where assays are collected repeatedly for the number of times input in the dialog mode
CALEFF	Calculates detector efficiency for given energy
CALIB	Performs data analysis to determine spectrum gain and zero
CALGPM	Corrects gamma-ray intensity for absorption and detector efficiency to determine gamma-ray emission rate from sample
CHK90	Checks error return on Canberra Series 90 call
CMPLTN (CMPLTL)	Estimates completion time for the assay, given the current time & the preset clock time of the measurement
CURDTA	Responds to the operator's request for assay from current data in MCA
DATA90	Returns Region of Interest data from the Canberra Series 90
DATTIM	Retrieves, verifies, or changes the system date and time
DIALOG (DIALGS)	Assay and calibration dialog with operator

DEFINITIONS OF SUBROUTINES CALLED (Continued)

Subroutine	Definition
LSLOG (LSLOG8)	Lists assay log
MCBIAS	Measurement control check for instrument bias
MCCHK (MCCHKS)	Checks that measurement control has been performed within the required time limits and also checks the status of the measurement control flag. These checks are made before an assay or an autocycle run
MCHEAD	Outputs header information for the measurement control log list option
MCPREC	Measurement control check for precision
MENU	Searches the table "MENU.TXT" to locate the option corresponding to the one entered by the operator
NEWNAM (NEWNM8)	Creates a new filename for an autocycle run
OUT (OUT38)	Accumulates results from FIT subroutine
P238GRP	Executive subroutine for data analysis
P238AB	Analyzes the gamma-ray emission results for relative plutonium abundances
PASS	Checks validity of user's password for entering supervisory mode
PRINT1 (PRINTL)	Prints assay results
PUT90	Gets a spectrum from disk and writes it to the Canberra Series 90
RDGAMS	Reads gamma-ray emission results from a disk file
RDHDR	Reads sample/detector information from PU238.SMP file
READ38	Reads analysis control information and open spectrum data file
READB	Reads background from disk
READC (READC38)	Reads constants from disk
READNM	Allows operator to select or change a constant's filename in a list of constants files

DEFINITIONS OF SUBROUTINES CALLED (Continued)

Subroutine	Definition
ERR_HANDLER	Sets system to look for occurrence of math errors in subroutine from which it is called, and sets ABORT flag to TRUE if an error occurs
ERRMSG	Reads appropriate error message from error message file, and outputs the message to the user terminal
EXEC (EXEC238)	Executive subroutine for controlling the Pu-238 isotopic assay instrument
FIT (FIT38)	Fits peaks to spectral data
FRDSK	Responds to operator selection of assay from disk
GET_KEY	Gets keyboard input from a buffer
GETOPS	Gets operator and sample ID from keyboard entry
GET90	Gets a spectrum from the Canberra Series 90, and writes to disk
HEADER (HEADR38)	Prints assay header
HELP (HELP38)	Outputs to the terminal the list menu options of the solid actinide assay instrument for the operator. If the supv flag is TRUE, supervisor options are listed
INIT90	Initializes Canberra Series 90
INITLP	Initializes printout for a long printout
INTFLG	Initializes logical flags to be used in the assay procedure
JULDAT (JULDAL)	Uses ASCII strings for date and time to compute Julian day
LGMENU	Lists menu options for the list assay log and list measurement control log
LHEAD	Outputs header information for the assay log list option
LMCLOG	Lists measurement control log
LNMN38	Checks the liquid-nitrogen monitor for low liquid-nitrogen-level condition in detector dewar

DEFINITIONS OF SUBROUTINES CALLED (Continued)

Subroutine	Definition
RESET (RESET38)	Resets default settings for assay options
ROTMON	Subroutine monitors the sample rotation sensor for a not-rotating condition.
SIZESAMP	Subroutine determines type of sample by sensing sample height (FCC or EP60/61 container)
SLEEP	Puts program to sleep for a given number of seconds
STABSET	Initializes Canberra 8232 stabilizer
START	Initializes program, allocates and assigns I/O channels to the ADAC device drivers
STATUS	Checks the status of the hardware for failures prior to the beginning of an assay, or responds to the option "S" entered from the menu list, which checks to see if hardware is busy
SVPKS (SVPK38)	Saves gamma-ray-emission results from current spectrum region in a disk file
WRITLG (WRTL8)	Writes the most recent assay information into the assay log file
WRTDAT	Calls the subroutine that retrieves the data from the hardware and writes it to disk, with the background data appended to the end of the raw data file
WTMCLG	Writes the most recent measurement control information into the measurement control log file
YESNO	Prompts for and accepts a yes-or-no answer to question passed in the call to the subroutine

CHAPTER 9

CONTENTS

9.	SUPPORTING PROGRAMS	9- 3
	How to Update and Run the PU238_CHNG Program	9- 4
	PU238_CHNG Program Options	9- 6
	PU238_CHNG Program Subroutines	9- 8
	Files Needed for PU238_CHNG Program	9- 9
	Command File to Link PU238_CHNG Program Subroutines	9-10
	Operations Guide for GRPANL Program	9-11

Chapter 9

SUPPORTING PROGRAMS

The supporting PU238_CHNG program allows the supervisor to change constant values for the Pu-238 isotopic analysis system that are contained in the PARMTR.WH8 file. The program is menu-driven and is run from the system console terminal. The system asks you to enter a two-letter code corresponding to the OPTION desired and then prompts you to enter the new values, if any. To leave any of the constants unchanged, enter RETURN when prompted.

PU238_CHNG program installation, flow, and subroutine descriptions are found on the system disk under the filename PU238_CHNG.DOC in the [WHC.PU_238.ASSAY] directory. A listing of this file and those of the PU238 CHNG program and the subroutines it calls are provided on the following pages.

HOW TO UPDATE AND RUN THE PU238_CHNG PROGRAM

System

MicroVMS operating system with FORTRAN installed.

Editing

Use any available editor to edit the source file. All source files have the extension FOR and are contained in the directory [WHC.PU_238.SOURCE].

If necessary, change a COMMON block by editing the appropriate common block file with the extension CMN. If a COMMON block needs to be added to an FOR file, add INCLUDE lines to the appropriate source files (FOR) as shown:

```
INCLUDE 'CONTRL.CMN'      (NOTE the use of single quotes!)
```

Compiling

To compile a single FOR file, use the command file FOR.COM, which will compile the file specified and create a list file. For example, the ADATE.FOR file is compiled by typing the command

```
@FOR ADATE
```

The compilation creates the OBJ and LIS files. The object file is stored in the directory [WHC.PU_238.OBJECT]. The listing file is stored in the source file directory.

To compile all the .FOR files for the PU238_CHNG program, type

```
@PU238_COMCHNG
```

when in the main directory [WHC.PU_238.ASSAY].

Linking

To link the program PU238_CHNG, use the command

```
@PU238_CHNG
```

when in the main directory [WHC.PU_238.ASSAY].

The new version of the program will be PU238_CHNG.EXE.

Documentation

Record all changes made in the change record, one line per routine. Print the LIS file and include it in the notebook.

Backup

Copy all changes to the backup TK50 tape per DEC instructions for creating backups.

Running

To run, these files must reside on the system:

- CONSTA.WH8 - List of parameter file names
- BACKGR.WH8 - Background information and passwords
- ERRMSG.INS - Error message file
- PU238_CHNG.EXE - Main program

PU238_CHNG Program Subroutine Calls

The sequence of subroutine calls in the PU238_CHNG program is shown below. Where the filename of the subroutine source differs from the subroutine name, the source filename is shown in parentheses.

```
CHANGE (PU238_CHNG)
GETOP
    COMM
    DATTIM
READB
    ERRMSG
PASS
READNM
    ERRMSG
READC (READC38)
    ERRMSG
DATTIM
    JULDAT
CHNGRL
CHNGAL
CHNGIN
WRITB
WRITC (WRIT38)
```


SUPPORTING PROGRAMS

PU238_CHNG Program Definition

This program allows you to change constant values. The program is menu-driven; you are asked to enter the two-letter CODE that corresponds to the OPTION desired. You will then be prompted to enter the new values. To leave any of the constants unchanged, enter RETURN when prompted. OPTION definitions follow.

PU238_CHNG PROGRAM OPTIONS

Code (Option)	Definition
HE (HElP)	Displays the menu.
MC (Measurement Control)	Allows you to modify bias or precision measurement controls using the subroutines CHNGRL and CHNGIN. When changing bias measurement controls, input the values for biastime, bias ref, biaswarn, bias act, backtime, and back lim in F format form [Format (F)]. When changing precision measurement controls, input the values for prcstime, prcswarn, and prcs act in F format form and input the value for prcsrns in I format form [Format (I)] 1 to 5 digits long.
PT (Preset Time)	Calls the subroutine CHNGRL, which allows you to change the time set
PW (PassWords)	Calls the subroutine CHNGAL, which allows you to change any of the passwords.
RD (Read constants from Disk)	Calls the subroutine READNM, allowing you to select or change a constant's filename in a list of constants files. Calls the subroutine READC, which reads constants from disk.
SZ	Calls the subroutines CHNGRL and CHNGIN to change the stabilizer settings for the zero and gain stabilizer settings. The values for peak positions and windows are entered as integers. The values for analog ranges are entered as floating point format.
DV (Default Values)	Calls the subroutine CHNGRL, which allows you to change the default value for the plutonium-240 weight percent abundance.
LI (LIst all constants)	Prints all constants on the output device.

PU238_CHNG PROGRAM OPTIONS

Code (Option)	Definition
EX (EXit)	Ends the Menu Options portion of the program. You are given the option of making the changes permanent by writing them to disk.
OU (change OUtput device)	Changes the output device from terminal to printer or vice versa.

SUPPORTING PROGRAMS

PU238_CHNG PROGRAM SUBROUTINES

Subroutine	Definition
GETOPS	Gets operator ID from keyboard entry. User may escape by entering RETURN in place of an ID number.
READB	Reads background from disk.
PASS	Checks validity of user's password to enter supervisory mode.
READNM	Allows operator to select or change a constant's filename in list of constants files.
READC (READC38)	Reads constants from disk.
DATTIM	Retrieves, verifies, or changes system date and time.
CHNGRL	Allows operator to change a REAL value.
CHNGIN	Allows operator to change an INTEGER value.
CHNGAL	Allows operator to change an ALPHANUMERIC value.
ERRMSG	Reads appropriate error message from the error message file, and outputs to the terminal.
JULDAT (JULDAL)	Uses ASCII strings for date and time to compute Julian day.
CHK90	Checks error return on Series 90 call.
WRITB	Writes background file to disk.
WRITC (WRITS38)	Writes constants to disk.

FILES NEEDED FOR PU238_CHNG PROGRAM

Name in data
statement

Filename

ERRFIL	ERRMSG.INS	Error message file name
BKFIL	BACKGR.IS8	Background run information
NAMSFL	CONSTA.IS8	Operator input of parameter file name

ERRMSG: ASCII file

NUM, IARG, ICHAR
I4, A1, 72A1

where: NUM is the error message call number
IARG is the ASCII argument passed
ICHAR is the number of characters in the error message

BACKGR: Binary file

BK(25), SGBK(25), TIMEBK, BKDAT(9), BKTIM(8), BKOP(12), BKSFIL(15)

where: BK is an array of 25 backgrounds
SGBK is an array of 25 sigmas for background areas
TIMEBK is the counting time for background
BKDAT is a 9-character ASCII string for background date
BKTIM is an 8-character ASCII string for background time
BKOP is a 12-character ASCII string: operator ID for background

assay

BKSFIL is a 15-character ASCII string for name of background raw-
data file

CONSTA:

CFLNAM, CCMMNT
15A1, 30A1

where: CFLNAM is the constants file name
CCMMNT is a remark line for the constants parameters

SUPPORTING PROGRAMS

COMMAND FILE TO LINK PU238_CHNG PROGRAM SUBROUTINES

```
!PU238_CHNG.COM
!  
! Command file to link CHNG program subroutines
!  
$ SD [.OBJECT]  
$ LINK/executable=[-]PU238_CHNG-  
    pu238_chng,-  
    chngri,-  
    dattim,-  
    errmsg,-  
    getop,-  
    juldal,-  
    nocomm,-  
    pass,-  
    readb,-  
    readc38,-  
    readnm,-  
    writb,-  
    writ38,-  
    yesno,-  
$SD [-.ASSAY]
```

Operations Guide for GRPANL Program

The following report, "Operations Guide for GRPANL: A Suite of Computer Programs for Analyzing Complex Ge and Alpha-Particle Detector Spectra, Volume 2", provides a detailed description of the operation of the GRPANL program and in particular the use of the EDIGRP program to edit the PU238.CTL file. The data-analysis control file, PU238.CTL, contains sequential sets of parametric information about the specific groups to be analyzed in a spectrum. It should not be necessary to modify the PU238.CTL file, however, this report is provided for informational purposes.

CHAPTER 10

CONTENTS

10. ERROR DIAGNOSTICS AND OPERATING ANOMALIES	10-3
10.1 Error Diagnostic Messages	10-3
10.2 Operating Anomalies	10-4

Chapter 10

ERROR DIAGNOSTICS AND OPERATING ANOMALIES

10.1 ERROR DIAGNOSTIC MESSAGES

The following is an error diagnostic that may occur when an assay is attempted:

```
**** ERROR #    1
*** ERROR IN OPENING FILE ***
MCLOGF.WH8
```

This message will occur when the assay option "A" is requested and the measurement control log file does not exist. This file does not exist when the instrument is started up initially. It is not created until a bias or precision check measurement is performed. The user may still proceed with the assay, but the results will be flagged, indicating that measurement control has not been performed.

Other types of errors that may occur are VMS FORTRAN Run-Time Library errors and System errors. Such errors may cause the instrument control program to halt and return the system to the keyboard monitor. Consult the VMS System Messages or VMS Run-Time Library Routines Reference Manuals for complete details on these types of errors. Most errors will occur because of math errors in the data-analysis routines. An example of such an error is

```
%SYSTEM-F-FLTDIV_F, arithmetic fault, floating divide by zero at
PC=00023CDB, PSL=03C00020
```

```
%TRACE-F-TRACEBACK, symbolic stack dump follows
```

module name	routine name	line	rel PC	abs PC
P238AB	P238AB	441	00000200	00023CDB
P238GRP	P238GRP	309	000002CB	0001FA17
ASSAY	ASSAY	279	000000DF	0001A1BB
FRDSK	FRDSK	127	0000007C	0001C1A0
EXEC	EXEC	3279	0000023E	001B98A
ASYP238	ASYP238	12	00000009	001A1D5

In this example, an arithmetic error caused by a division by zero has occurred. The source line numbers in the subroutines are given, allowing the user to trace the path and location of the error.

In the spectral data analysis part of the SRS_ASY238 program, a subroutine has been included to prevent the program from halting if certain math errors occur during the data analysis. These errors are division by zero, square root of a negative number, math overflow, and logarithm of a zero or negative argument. If such an error occurs, this subroutine will detect it, print a message indicating the type of error and the name of the subroutine where the error occurred, and set the ABORT flag to TRUE. This will cause the program to abort the analysis and return to the main option query message.

In those cases where an error occurs and the program either halts or hangs up, the program must be restarted. Nevertheless, try to determine the cause of the error so that (1) the error can be avoided in the future or (2) a fix can be made to the program.

10.2 OPERATING ANOMALIES

This section describes some of the problems that may arise in operating the system. It is impossible to foresee all the problems that may develop, so those described here are not inclusive. It is suggested that you keep a record of the problems encountered, either here or in another notebook, together with the procedures used to solve the problem.

One such problem is initial system startup before calibration or after a disk failure. In this case, some of the files needed by the instrument-control program may not exist. Unless the PARMTR and BACKGR files or the peak shape constants file PU238.SHP are on the disk, the following diagnostic messages will appear on the operator terminal when the program is started:

```
**** ERROR # 1
*** ERROR IN OPENING FILE ***
PARMTR.WH8

**** ERROR # 102
*** SUPERVISOR NEEDS TO CORRECT PROBLEM ***
```

These messages indicate that one of the files needed for operation of the instrument is missing. In this example, the parameter file PARMTR.WH8 is missing. To correct this problem, you should run the PU238_CHNG program and create the parameter file.

If the BACKGR file is missing, the program will not run at all. This file contains the supervisor passwords, and the program cannot operate unless the passwords are available. To create the BACKGR file, use the PU238_CHNG program with the default password (see description of the PU238_CHNG program in Chapter 9 and in the Pu-238 Isotopic Analysis Users Manual, UCRL-xxxxxx, Vol. 1).

Technical Information Department · Lawrence Livermore National Laboratory
University of California · Livermore, California 94551