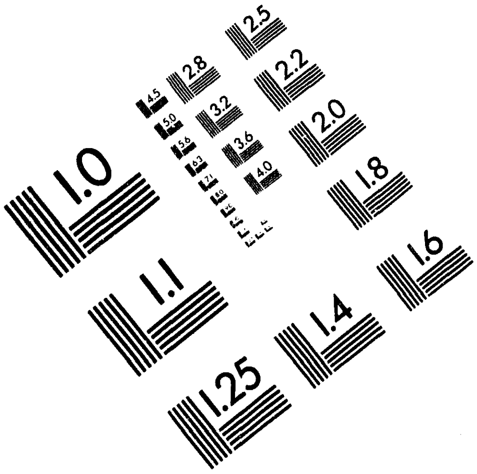
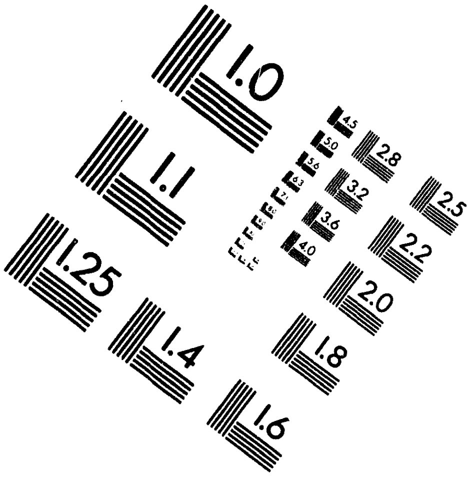


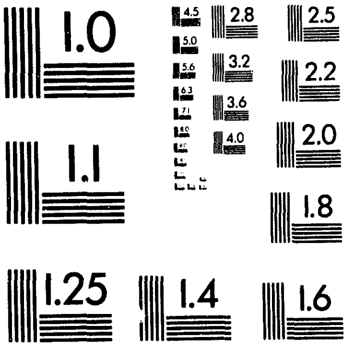
AIIM

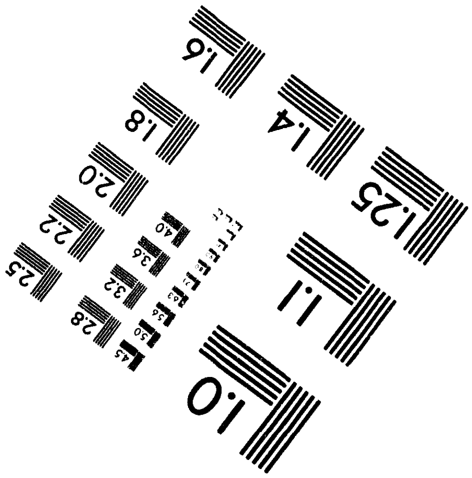
Association for Information and Image Management

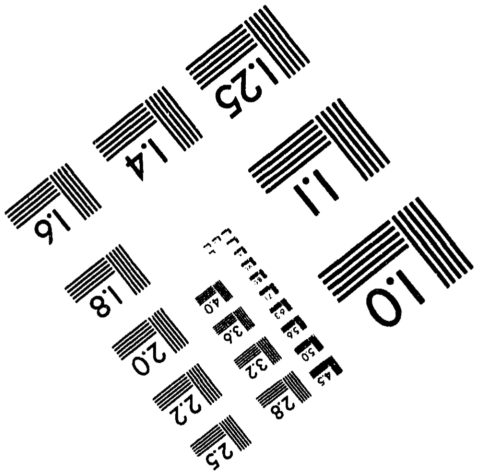
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

301/587-8202









MANUFACTURED TO AIIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 1

2

DISTRIBUTION SHEET

To Distribution	From Criticality & Radiological Analyses	Page 1 of 1 Date July 25, 1994
Project Title/Work Order User's Guide to The FFTF Plant Operational Data Management System (B1039)		EDT No. 161566 ECN No. N/A

Name	MSIN	Text With All Attach.	Text Only	Attach./ Appendix Only	EDT/ECN Only
D. M. Ahrens	N2-01	X			
J. W. Daughtry	H4-64	X			
K. D. Dobbin	N2-32	X			
S. F. Kessler	N2-32	X			
J. V. Nelson, Jr.	H4-64	X			
R. F. Richard	H4-64	X			
C&RA File	H4-64	X			
Technical Reference Center	N2-12	X			
Central Files (Orig. +2)	L8-04	X			
IRA (2)	H4-17	X			

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

35 Station 21

ENGINEERING DATA TRANSMITTAL

Page 1 of 1

1. EDT 161566

2. To: (Receiving Organization) Criticality & Radiological Analyses	3. From: (Originating Organization) Criticality & Radiological Analyses	4. Related EDT No.: N/A
5. Proj./Prog./Dept./Div.: FFTF	6. Cog. Engr.: J. V. Nelson, Jr.	7. Purchase Order No.: B1039

8. Originator Remarks:
The purpose of this EDT is to release the user's guide for the FFTF Plant Operational Data Management System

9. Equip./Component No.: N/A

10. System/Bldg./Facility: N/A

11. Receiver Remarks:

12. Major Assm. Dwg. No.: N/A

13. Permit/Permit Application No.: N/A

14. Required Response Date: 08-94

15. DATA TRANSMITTED

(A) Item No.	(B) Document/Drawing No.	(C) Sheet No.	(D) Rev. No.	(E) Title or Description of Data Transmitted	(F) Approval Designator	(G) Reason for Transmittal	(H) Originator Disposition	(I) Receiver Disposition
1	WHC-SD-FF-CSUD-019		0	User's Guide to The FFTF Plant Operational Data Management System	N/A	1/2	1	

16. KEY

Approval Designator (F)	Reason for Transmittal (G)	Disposition (H) & (I)
E, S, Q, D or N/A (see WHC-CM-3-5, Sec.12.7)	1. Approval 4. Review 2. Release 5. Post-Review 3. Information 6. Dist. (Receipt Acknow. Required)	1. Approved 4. Reviewed no/comment 2. Approved w/comment 5. Reviewed w/comment 3. Disapproved w/comment 6. Receipt acknowledged

17. SIGNATURE/DISTRIBUTION
(See Approval Designator for required signatures)

(G) Reason	(H) Disp.	(J) Name	(K) Signature	(L) Date	(M) MSIN	(J) Name	(K) Signature	(L) Date	(M) MSIN	(G) Reason	(H) Disp.
2	1	Cog.Eng. J.V. Nelson, Jr.	JV Nelson Jr	7/25/94	H4-64						
2	1	Cog. Mgr. J.W. Daughtry	JW Daughtry	7/25/94	H4-64						
		QA									
		Safety									
		Env.									

18. JV Nelson, Jr. JV Nelson, Jr. 7/25/94 Signature of EDT Originator Date	19. Authorized Representative Date for Receiving Organization	20. JW Daughtry JW Daughtry 7-25-94 Cognizant Manager Date	21. DOE APPROVAL (if required) Ctrl. No. □ Approved □ Approved w/comments □ Disapproved w/comments

Date Received:
7-26-94 Arl

INFORMATION RELEASE REQUEST

Reference:
WHC-CM-3-4

Complete for all Types of Release

Purpose

[] Speech or Presentation
  [] Full Paper (Check only one suffix)
  [] Summary
  [] Abstract
  [] Visual Aid
[] Speakers Bureau
[] Poster Session
[] Videotape

[] Reference
[] Technical Report
[] Thesis or Dissertation
[] Manual
[] Brochure/Flier
[] Software/Database
[] Controlled Document
[X] Other - SD

ID Number (include revision, volume, etc.)
WHC-SD-FF-CSUD-019, Rev 0

List attachments.
N/A

Date Release Required
N/A

Title User's Guide to The FFTF Plant Operational Data Management System

Unclassified Category
UC-N/A

Impact Level N/A

New or novel (patentable) subject matter? [X] No [] Yes
If "Yes", has disclosure been submitted by WHC or other company?
[] No [] Yes Disclosure No(s).

Information received from others in confidence, such as proprietary data, trade secrets, and/or inventions?
[X] No [] Yes (Identify)

Copyrights? [X] No [] Yes
If "Yes", has written permission been granted?
[] No [] Yes (Attach Permission)

Trademarks?
[] No [X] Yes (Identify) CRAY, SUN, SUNOS, SPARC station, SUN Workstation, UNISYS, UN1X

Complete for Speech or Presentation

Title of Conference or Meeting

Group or Society Sponsoring

Date(s) of Conference or Meeting

City/State

Will proceedings be published? [] Yes [] No
Will material be handed out? [] Yes [] No

Title of Journal

CHECKLIST FOR SIGNATORIES

Review Required per WHC-CM-3-4	Yes	No	Reviewer - Signature Indicates Approval Name (printed)	Signature	Date
Classification/Unclassified Controlled Nuclear Information	[]	[X]			
Patent - General Counsel	[]	[X]			
Legal - General Counsel	[]	[X]			
Applied Technology/Export Controlled Information or International Program	[]	[X]	approval per telecon J. D. Watrous VB		8/12/94
WHC Program/Project	[]	[X]			
Communications	[]	[X]			
RL Program/Project	[X]	[]	R.A. Almquist	R A Almquist	7/27/94
Publication Services	[X]	[]	Brenda Ady	B. Ady	8-8-94
Other Program/Project	[]	[X]			

Information conforms to all applicable requirements. The above information is certified to be correct.

	Yes	No
References Available to Intended Audience	[X]	[]
Transmit to DOE-HQ/Office of Scientific and Technical Information	[X]	[]

Author/Requestor (Printed/Signature) Date
J. V. Nelson, Jr JV Nelson Jr. 7/25/94

Intended Audience
[] Internal [] Sponsor [X] External

Responsible Manager (Printed/Signature) Date
J. W. Daughtry JW Daughtry 7-25-94

INFORMATION RELEASE ADMINISTRATION APPROVAL STAMP

Stamp is required before release. Release is contingent upon resolution of mandatory comments.

APPROVED FOR RELEASE
OFFICIAL DOCUMENT
VB
Date 8/12/94

Date Cancelled

Date Disapproved

SUPPORTING DOCUMENT

1. Total Pages 68

2. Title

User's Guide to The FFTF Plant Operational Data Management System

3. Number

WHC-SD-FF-CSUD-019

4. Rev No.

0

5. Key Words

FFTF, Data Management, Nuclear Reactor, Operational Data

APPROVED FOR PUBLIC RELEASE

V. Birkland 8/12/94

6. Author

Name: J. V. Nelson, Jr.

J V Nelson Jr.

Signature

Organization/Charge Code 8D150/B1039

7. Abstract

The FFTF Plant Operational Data Management (PODM) System was designed to store manage and retrieve operational data recorded by computer-based data acquisition systems in the FFTF.

This document describes the PODM system and includes detailed instructions for using the computer software available to retrieve and process the stored FFTF data.

8. ~~PURPOSE AND USE OF DOCUMENT - This document was prepared for use within the U.S. Department of Energy and its contractors. It is to be used only to perform, direct, or integrate work under U.S. Department of Energy contracts. This document is not approved for public release until reviewed.~~

~~PATENT STATUS - This document copy, since it is transmitted in advance of patent clearance, is made available in confidence solely for use in performance of work under contracts with the U.S. Department of Energy. This document is not to be published nor its contents otherwise disseminated or used for purposes other than specified above before patent approval for such release or use has been secured, upon request, from the Patent Counsel, U.S. Department of Energy Field Office, Richland, WA.~~

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, nor any of their contractors, subcontractors or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use 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 or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

9. Impact Level N/A

10. RELEASE STAMP

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

USER'S GUIDE TO THE FFTF PLANT OPERATIONAL DATA MANAGEMENT SYSTEM

J. V. Nelson

July 20, 1994

TABLE OF CONTENTS

Page

APPENDICES

Page

LIST OF FIGURES

LIST OF TABLES

USER'S GUIDE TO THE FFTF PLANT OPERATIONAL DATA MANAGEMENT SYSTEM

1.0 SUMMARY

The FFTF Plant Operational Data Management (PODM) System provides capabilities for storing, managing and retrieving data recorded by FFTF plant computers [the Plant Data System (PDS), in particular]. The PODM system is currently implemented on SUN[TM] Workstations[(R)]. This guide contains a description of the PODM System, and instructions for using programs available for retrieving and processing FFTF data stored in the data base.

Section 2.0 provides a brief overview and the background of the system. The organization and content of the data base are described in more detail in Sections 3.0 and 4.0. Available computer programs are described in sections 5.0 and 6.0 while subroutines that can be called by a user's FORTRAN program are described in section 7.0.

[TM] SUN is a trademark of Sun Microsystems, Inc.
[(R)] SUN workstation is a registered trademark of Sun Microsystems, Inc.

2.0 INTRODUCTION

A number of computers in the FFTF plant monitor and record data indicating the performance, environment and status of FFTF systems, components and tests. Data tapes recorded by some of these computers have been processed and stored for retrieval by PODM System software. In general, the PODM system was designed to provide long-term retention of FFTF operational records in computer retrievable form for use by operations, operations support, engineering, safety, and experimenter organizations. Specific functional requirements that the system was designed to satisfy can be found in References 1 and 2. Preliminary scoping and planning work was documented in References 3 and 4.

2.1 Purpose

The purpose of this guide is to briefly describe the PODM system and provide detailed instructions for using software available for retrieving, processing and displaying information in the data base.

2.2 Scope

Data recorded by three plant computers have been loaded into the data base in the past. Two computers are part of the Digital Data Handling and Display System (DDH&DS). These are the Plant Data System (PDS) and the Experimenters Data System (EDS). The PDS records facility data such as reactor power, core temperatures and temperatures, flowrates and pressures in the heat transport system. The EDS records experimenters' data from instrumented test assemblies, plus key facility data as a backup to the PDS. The third data source is the In-Vessel Handling Machine (IVHM) computer that records assembly insertion and withdrawal forces during refueling operations. Recently though, only data recorded by the PDS have been processed into the data base.

TM SPARCstation is a trademark of SPARC International, Inc.
SUNOS is a trademark of SUN Microsystems, Inc.

2.3 Environment

The PODM System is currently plemented on SUN SPARCstation™ 2 computers ces1 and ces3. These workstations operate under SUNOS™ 4.1.1. Almost all coding is in FORTRAN 77. One subroutine in one program was coded in C. The FORTRAN code was compiled using f77.1.3.1, and the C code was compiled using the C compiler distributed with SUNOS 4.1.1.

Executable code and UNIX(R) shell scripts are maintained on SUN disk space. Data base files are maintained on the Common File System (CFS), but must be transferred to SUN temporary disk space for processing. This transfer is handled automatically by the PODM System.

2.4 Background

The PODM System was first implemented on a UNISYS™ 1100 computer (Ref. 5). All data recorded by the PDS from July 27, 1979 to September 24, 1988 were stored using the system implemented on the UNISYS computer. The EDS began recording data routinely on July 10, 1980. Data recorded by the EDS from then to September 2, 1988 were processed using the UNISYS.

In September 1988, the PODM was migrated to the first Hanford CRAY(R) computer. PDS data recorded from September 16, 1988 to August 12, 1993 were processed using the CRAY-based PODM system. EDS data recorded from September 2, 1988 to December 23, 1989 were processed using the CRAY-based system. EDS data recorded after that time were not processed.

Data recorded by the IVHM computer from October 18, 1988 to December 25, 1989 were processed into the FFTF data base.

In August 1993, the PODM System was migrated from the second Hanford CRAY computer to the SUN workstations, where the system currently resides.

(R) UNIX is a registered trademark of UNIX System Laboratories, Inc.
CRAY is a registered of Cray Research, Inc.

™ UNISYS is a trademark of UNISYS Corp.

The FFTF data processed by the UNISYS computer were written to magnetic tape. Data processed on the two CRAY computers and the SUN workstations were written to disk files, which were eventually transferred to CFS for permanent storage. Data tapes created by the UNISYS cannot be read by PODM software on the SUN workstations. However, data files created by the CRAY computers can be read using the SUNs even though the file formats are different. The differences are completely transparent to the user.

To access FFTF data processed before the PODM System was operational on the Hanford CRAY (September 1988), the original copies of the data tapes must be obtained from the records center and processed on the SUN system.

2.5 Related Documentation

References 2 - 5, cited earlier, provide more background on the PODM System. Reference 1 described the programs used to process data tapes from FFTF computers and maintain the data base. This document also provides source code listings, version numbers and release dates for all PODM programs.

3.0 FLOW OF INFORMATION INTO THE DATA BASE

The steps involved in transferring operational data on magnetic tape from FFTF plant computers to the data base are outlined in Figure 3-1. Tapes containing data to be entered into the data base are first duplicated at the FFTF site. The originals are stored at the 400 Area and the copies are sent via messenger service to the central computer facility in the Federal Building. There, they are read on the Large Scale Information System (LSIS) and copied to disk. The disk files are then transferred to a SUN workstation (ces1 or ces3) for processing.

Data from the PDS and EDS are translated and partitioned into logical sensor groups during this processing. Data in each group are then used to update separate disk files. In Figure 3-1, two PDS and two EDS groups are shown only for illustrative purposes. In fact, PDS data are divided into four groups and there was one EDS group per experimenter organization having instrumented

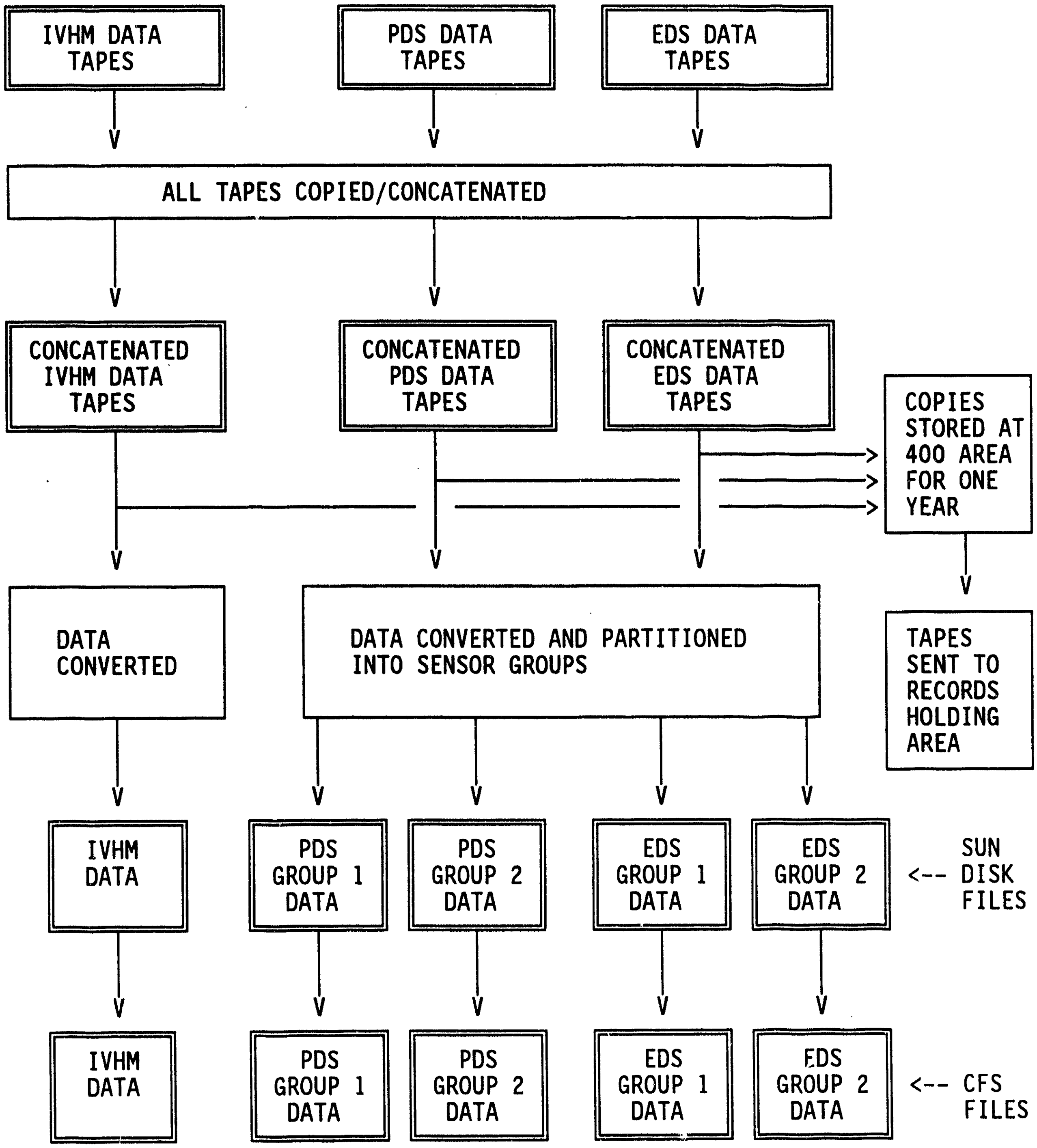


Figure 3-1. Flow of FFTF Operational Data into Data Base

tests in the reactor, plus one group for facility data backed up on the EDS. The general breakdown of PDS facility data into the four groups is summaried in Table 3-1.

Data records from the IVHM computer were reformatted during duplication to be in the same format as PDS and EDS tapes. IVHM data were then processed like PDS and EDS data except that IVHM data were not divided into different groups.

4.0 DATA BASE ORGANIZATION AND CONTENT

As indicated above, sensor records are segregated into groups as they are processed into the data base. Groups 1, 2, 3, and 5 contain the PDS data summarized in Table 3-1. Group 11 contains key facility data recorded by the EDS as backup. Group 20 contains all available data recorded by the IVHM computer.

Each variable record in the data base includes a sensor identifier, a value in engineering units, the date and time recorded and a number of status flags. (A detailed description of the records is given in Appendix A). Complete tabulation of variables recorded by the PDS and EDS is not included in this report because of the large number of sensors involved. Also, there can be changes from cycle to cycle. A current list of sensors recorded by the PDS is available. The available IVHM parameters are listed in Appendix G.

Table 3-1

Breakdown of Plant Sensors Into Groups

Group	Content
1	Subassembly powers
	Subassembly temperature
	Subassembly flow rates
	Total power
	Reactivity
	Flux Deviations
	Inlet and outlet temperatures
	Flow fraction
2	Other temperatures
	• Primary and secondary loops
	• IHX and DHX
	• Instrument tree
	• Head
	• Vessel nozzle
	Data on primary and secondary loop systems
	• Flow
	• Pump speed
	• Sodium level
	• Pressure
	DHX valve positions
	Blower speed and damper positions
3	Control rod positions
	LLFM and EVFM count rates
	Flux source rate
5	Radiation and particle monitors
	Calibration data
	PPS channels
	All System 61 sensors
	All System 81 sensors

5.0 PROGRAMS PROVIDING DIRECT ACCESS TO THE DATA BASE

Sections 5.1 through 6.3 contain brief descriptions of programs that are available for retrieving, displaying and analyzing FFTF operational data. Further details and instructions on these programs can be found in Appendix C. Section 5.1 through 5.5 are devoted to programs that obtain information directly from the data base while Sections 6.0 through 6.3 deal with programs to process operational data that have been previously extracted from the data base and stored on user files.

5.1 Data File Assignment Routine, REQFL

Because of the volume of data produced by multiple FFTF plant computers and the division of data into logical sensor groupings, the operational data base consists of a large number of files. To eliminate the necessity of having to determine which file(s) contain information of particular interest, the REQFL program that automatically selects and assigns required data files was written. A second function of REQFL is to provide data security by controlling access to the data base. Before REQFL assigns any files, a master password must be supplied by the user. This password allows access to all facility data. The security system also has the capability of providing additional security for experimenter-owned data from instrumented test assemblies.

After requesting and checking a user's password entry, REQFL reads the range of dates and times over which data are requested. Finally, REQFL reads identifiers for the required sensors, or the numbers of the groups into which these sensors have been placed. Using this information, the program determines which files contain the required data and initiates instructions for requested files to be transferred from CFS to temporary on-line disk files, if necessary. REQFL also writes a communication file expected by most data retrieval routines.

Even though assignment of data files necessary to complete a requested report is automatic, the user still needs to have an idea of how many files are necessary. No more than 12 data files can be assigned for any data group in a single run. Methods for determining in advance how many input files must be read to complete a desired report are presented in Appendix B. It may require 10 minutes to stage and transfer each file from CFS.

Another feature of the REQFL program allows the user to specify that only the latest file in each specified group is to be assigned. This option is useful if a user wants to interactively look at only the most recent information in the data base without knowing in advance the time period of data included on the files.

Detailed input instruction of REQFL are given in Appendix C.1.

5.2 The Data Review Program, CDREV

The CDREV program is designed to produce correlated data reports in either the batch or interactive mode. CDREV is used in conjunction with the REQFL preprocessor that assigns files containing records for the sensors and the time intervals requested. Input instructions for CDREV are given in Appendix C.2 with examples. A complete batch runstream is listed and described in Appendix E and an annotated printout from a typical interactive session is given in Appendix F. Some of the general features and limitations of the CDREV program are discussed in the following paragraphs.

An example of a report produced by CDREV is shown in Appendix F. A single report can contain information on up to 200 sensors. The number of columns of sensor data on a single page is specified by the user, but cannot exceed ten. The number of data lines on each page is 45. If a report is to list data for more sensors than can be put on one page, consecutive pages will be used to include all sensors at each set of 45 time periods. The time interval between each line in the report is also specified by the user and can be any value between 0.1 sec. and 24 hours. If more than one value is recorded for a particular sensor in a single time interval, the first recording encountered is printed.

The CDREV program requires less than 200K bytes of memory to execute. Running times are job dependent and can range from a few seconds to over several minutes. Computer time is a function of the number of files required, length of the time period over which data re retrieved and the specified retrieval frequency.

The status of sensor values printed by CDREV is indicated by a single character immediately following the value printed. Status flags that could appear include:

H - The value exceeds the high alarm limit.
L - The value is below the low alarm limit.
* - The value is outside the valid sensor range.
N - The scan rate for this sensor has been changed.

When successive recordings of a sensor are invalid, the DDH&DS also labels all following values as being invalid until a reactor operator resets the invalid alarm.

5.3 Program to Generate User Files, GULF

The GULF program was designed to create data files containing user-specified subsets of the data base. This program is similar to CDREV in that a time period and a set of sensor identifiers ($\leq$ 800) are specified via input. Also like CDREV, a time increment to allow regulation of the frequency of data output records is specified. Each sensor record written to the output file by GULF consists of:

- The data and time
- Sensor identifier
- Sensor value in engineering units
- Engineering units code
- DDH&DS status word

where the units codes and status bits are defined in Appendix A.

Three options for the format of the output file are available. The first is a standard FORTRAN binary file with each logical record containing the data for one sensor at one time interval. The other two options are FORTRAN formatted

(ASCII) files. With option 2, each record contains the same information as option 1. A third option is an ASCII file where each 80-character record contains two sensor records.

Memory requirements and running times are similar to those for CDREV. Input instructions for using GULF and a detailed description of the output files are given in Appendix C.3.

5.4 Program to Time Average Sensor Data, CGULF

The CGULF program is similar to GULF in that the data base is accessed and data files are created. The key difference is that CGULF time averages sensor data over specified intervals of time, making it useful for smoothing noisy signals. The length of the averaging interval and the frequency of output are both input parameters. For example, the output could consist of sensor data averaged over a five minute interval each hour on the hour.

Another difference between GULF and CGULF is the content of the output when a standard FORTRAN binary or FORTRAN formatted output file (options 1 or 2) is specified. In these cases, CGULF writes out the observed standard deviation of each sensor in each averaging interval in addition to the five items in a GULF record. When output file format option 3 is called for, CGULF creates a file identical in content and format to GULF. Input instructions for CGULF and a detailed description of the output files are given in Appendix C.4.

5.5 Programs to Summarize Pump Motor Operations, PUMPS and PMPRPT

The PUMPS program summarizes the operating history of the primary and secondary loop pump motors. The operating time of each main motor in each of four pre-selected speed (pump RPM) ranges is accumulated and reported. Also tallied and reported is the operating time of each pony motor.

A file, pmphis, containing operating time accumulated from previous PUMPS runs is read and updated. The PMPRPT program, which can be run to produce summary reports, also reads the pmphis file.

Input instructions for PUMPS and PMPRPT are given in Appendix C.5, along with the format and content of the pmphis file.

6.0 PROGRAMS ACCESSING GULF AND CGULF GENERATED FILES

The GULF and CGULF programs described in the previous section are used to retrieve and store specified subsets of the data base for later processing. General purpose programs that read these data files are described below.

6.1 Program to Average Sensor Data, SENAV

The SENAV program is designed to average sets of two or more sensors together and write the results on disk or tape in GULF format. The sensor data processed by SENAV are read from a single file created by GULF, CGULF or a previous SENAV run.

SENAV is typically used to average redundant sensors together or to average corresponding sensors in each of the three heat transport loops to yield reactor averages. SENAV input instructions are given in Appendix C.6. The program can optionally be instructed to discard from the averaging process sensor records that are flagged invalid or where readings fall outside specified limits. This feature was incorporated to provide a means of preventing a faulty sensor from invalidating the resulting averages.

6.2 Program to Generate Plot Files, GPD

The function of GPD is to read specified data from a GULF, CGULF, or SENAV (GULF-format) file and produce a complete input file for a plotting routine that is not available on the SUN workstations. However, GPD output files may be preferable to GULF-format ASCII files as input to standard plotting or spreadsheet programs.

Input instructions and the output file format for GPD are described in Appendix C.7.

6.3 Programs to Merge GULF-Format files, MERGF

The MERGF program reads and merges two or more files generated by GULF, CGULF or SENAV. The resultant file contains all of the sensor records on the input files in proper time order. MERGF is also useful for concatenating or reformatting GULF-format files.

Instructions for using MERGF can be found in Appendix C.8.

7.0 SUBROUTINE LIBRARY FOR DATA RETRIEVAL AND PROCESSING

In the previous sections brief descriptions of complete programs for retrieving and processing FFTF data were given Also available is PODMLIB, a library of FORTRAN callable subroutines and functions through which a user's program and access and process FFTF data. All programs described in the previous sections call routines in PODMLIB.

A summary data retrieval subroutine in PODMLIB is given in Table 7-1 while other available subroutines and functions are summarized in Table 7-2. Instructions for using the routines in PODMLIB are given in Appendix D.

Table 7-1

Data Retrieval Subroutines in PODMLIB

Subroutine Name	Function
OPENDF	Opens data files assigned by REQFL
INITFO	Positions files previously opened by OPENDF, sets up a table of sensors to be retrieved and performs other initializations
GETMSO	Retrieves data records in chronological order for sensors specified in call to INITFO
SKIT	Repositions previously opened files to a specified time

Table 7-2

Other Subroutines and Functions in PODMLIB

Subroutine Name	Function
CA2SN	Converts a list of sensor identifiers (in ASCII) to a corresponding list of integers used internally in the data base to identify sensors.
NCODT	Encodes a date and time (year, month, day, hours, minutes and seconds) into data base format.
DECODT	Decodes a data base date and time word into six words (year, month, day, hours, minute and seconds).
KUNITS	Extracts the engineering units code from a data base record
ICOND	Interprets the DDH&DS status word and returns the status.
ISTATWD	Returns a word containing the 16 DDH&DS status bits.
IVALID	Determines from a 16 bit status word whether a sensor record is valid or not.
NEXTIM	Adds a time (hours, minutes, seconds) to a data base date and time word. The result is in data base format.
IPAST	Subtracts a time (hours, minutes, seconds) from a data base date and time word. The result is in data base format.
CLEAR/CLEARC	Set all elements of a specified array to an initial value.
KRONO	Determines the chronological order of two pair of date/time words in data base format.
TDIF	Calculates the difference in seconds between two given dates/times.
VERSN	Returns the version number and release date of PODMLIB.

8.0 REFERENCES

1. J. V. Nelson, "The FFTF Plant Operational Data Management System," WHC-SD-FF-SWD-063, Rev. 0, released June 10, 1994.

2. Letter, T. A. Mangelsdorf to Director, FFTFPO, "Functional Requirements for FFTF Operational Data Management," 7952796, September 25, 1979.

3. S. A. Hunt and S. E. Busby, "Report on FFTF Site Computing Study," TC-1071, Hanford Engineering Development Laboratory, April 1978.

4. Letter, T. A. Mangelsdorf to Director, FFTFPO, "FFTF Operational Data Management Plan," 7863267, October 6, 1978.

5. J. V. Nelson and M. R. Gilbert, "Users' Guide to the FFTF Plant Operational Data Management System," HEDL-TC-2506, January 1984.

6. Letter, T. A. Mangelsdorf to Director, FFTFPO, "FFTF Data Output Control and Cost," 7950990, March 1979.

APPENDIX A

Content of Sensor Records in the Operational Data Base

Each data record in the FFTF Operational Data Base consists of 8 bytes (64 bits) containing the following information:

a sensor identifier,
the sensor value in engineering units,
a code indicating the units,
the sensor status.

The engineering units codes are defined in Table A-1. The sensor status is given by a 16 bit status word. The meaning of each of the bits is defined in Table A-2. Only the first eleven status bits that are used are stored in the operational data base.

Associated with each data record is an 8 byte date and time entry that gives the time to the tenth of a second. FFTF plant clocks used to time-stamp each data record written to tape are set to local time -- either Pacific Standard Time (PST) or Pacific Daylight Time (PDT). The Operational Data Base uses a constant time base -- PST. When Daylight Time is in effect, recorded time values are converted from PDT to PST before entry into the data base.

The internal structure of the data base, and the format of the data records and date/time entries, which vary depending on whether they were written on the CRAY or a SUN, are described in detail in Reference 1. Regardless, data records are converted to an internal PODM format when they are read in. This format is described later in Appendix D, Section D.1.3.

Table A-1

Engineering Units Codes

Code	Engineering Units	Abbreviation In Printouts
01	degrees Fahrenheit	DEG F
02	gallons per minute	GPM
03	10^3 gallons per minute	KGPM
04	counts per second	CPS
05	percent of full power	% PWR
06	feet	FEET
07	inches	INCHES
08	millirem per hour	MR/HR
09	decades per minute	DPM
10	revolutions per minute	RPM
11	pounds per square inch	PSI
12	pounds per square inch - gauge	PSIG
13	relative weighting (0 to 1)	COEFF
14	volts a-c	VAC
15	counts per minute	CPM
16	hertz	HZ
17	value logic (0/1/2/3=moving/closed/open/invalid)	VLOGIC
18	millivolts	MV
19	PPS relay logic (0/1=open/closed)	NLOGIC
20	percent open	% OPEN
21	rad	RAD
22	megawatts thermal	MWT
23	dollars per hour	$/HOUR
24	megawatt days	MWDAYS
25	percent of full flow	% FLOW
26	kilowatts	KWT
27	cents	CENTS
28	percent full	% FULL
29	gallons per minute (approximate)	AP-GPM
30	no units	
31	hours	HOURS
32	dollar	DOLLAR
33	days	DAYS
34	seconds	SECOND
35	inches of water	IN H2O
36	percent of full flow pressure drop	%FFPD
37	megawatts	MW
38	no units	
39	no units	
40	amperes	AMPS
41	milliamperes	MA

Table A-2

DDH&DS Status Bits

DDH&DS Bit No.[a]	Meaning when bit is set (=1)	Normal value[b]
1	high alarm limit exceeded	0
2	low alarm limit exceeded	0
3	invalid readings (2 consecutive)	0
4	single invalid reading (noise)	0
5	alarm acknowledged	0
6	alarm logged	0
7	processing	1
8	limit checking	1
9	recorded	1
10	increased data rate	0
11	I/O active	1
12-16	Not Used	0

[a] Bit #1 is the leftmost (highest order) bit in the word.

[b] Normal value of the 16 bit status word is 928 decimal.

APPENDIX B

Content of FFTF Operational Data Base

A program, LSTMFD, is available to give a table of contents of the FFTF operational data base maintained in CFS. An example of the reports produced by LSTMFD is given in Table B-1. From such a table, a user can determine the time span included in the data base and also can determine how many files would be necessary to generate a particular report.

The LSTMFD program can be executed by entering:

```
   /apps/sh/lstmfd
```

anywhere in either a batch or interactive run.

LSTMFD has four optional input data cards (lines) which are described in Table B-2. These cards are used to obtain a listing of only a specified subset of the available data files.

Another method for determining the number of files required to produce a report is to simply execute the REQFL program per instructions in Appendix C.1 in an interactive run. The REQFL program will indicate how many data files are required for a report. These files will be automatically assigned at that point. If the user does not want to assign the files, hit the Control and C keys simultaneously immediately after the message, "x FILES CONTAIN REQUESTED DATA."

Table B-1

Sample LSTMFD Report

```
CONTENTS OF FFTF OPERATIONAL DATA BASE ON  2-Jun-94 AT 09:19:16

  FILE     GROUP  STORAGE     BEGINNING          ENDING
  NAME      NUM.   DEVICE   DATE     TIME     DATE     TIME   RECORDS  BLOCKS
--------   -----  -------  --------------    --------------   -------  ------
D01AMR      01     CFS     1/16/94  7: 0     2/18/94  0:19   1396592    3137
D01AMQ      01     CFS    12/ 2/93 18:40     1/16/94  7: 0   1869652    4209
D01AMP      01     CFS    10/30/93 17:56    12/ 2/93 18:39   1469840    3287
D02AJZ      02     CFS     2/14/94  1:20     2/18/94  0:19    640281    1768
D02AJY      02     CFS     2/ 1/94  9:35     2/14/94  1:20   1937894    5297
D02AJX      02     CFS     1/20/94  6:53     2/ 1/94  9:35   1963077    5378
D03AJU      03     CFS     2/10/94  4: 1     2/18/94  0:19    634405    2106
D03AJT      03     CFS     1/24/94  7:32     2/10/94  4: 1   1315987    4346
D03AJS      03     CFS     1/ 8/94  7:54     1/24/94  7:14   1345036    4439
D05AGT      05     CFS     2/14/94  1:20     2/18/94  0:19    319074     692
D05AGS      05     CFS     1/20/94  6:53     2/14/94  1:20   1932939    4185
D05AGR      05     CFS    12/24/93 13: 9     1/20/94  6:53   2037615    4404

 THIS IS A LIST OF THE MOST RECENT FILES
 FOR GROUPS:  1  2  3  4  5
```

Table B-2

LSTMFD Input Cards

Card 1: List type

Col. 1: Enter S, N, D or L.

S will result in a list containing only the three most recent files in each group.

N will result in a list containing files in the active library.

D will result in a list of files containing data recorded between dates specified on Card 2.

L will result in a list containing files in both the active library and the archives.

Col. 2: Enter G or leave blank.

If G is entered, only files for groups specified on cards 3 and 4 will be listed.

If Col. 2 is left blank, cards 3 and 4 are not read and a list containing files in all groups is produced.

Card 2:[a] Dates to be included in the report. The format is MMDDYY, MMDDYY. If the first entry (the starting date) is zero, then all files containing data up to the ending date (the second entry) will be listed. If the second entry is zero, then all files containing data from the starting date up to the present will be listed

Card 3:[b] Total number of groups to be listed (free field format)

Card 4:[b] Groups whose files are to be listed (free field format). The number of entries on this card is specified on Card 3.

[a] Card 2 is read only if column 1 of Card 1 contains a "D".

[b] Cards 3 and 4 are read only if Col. 2 of Card 1 contains a "G".

APPENDIX C

Instructions for Using Programs in the PODM Library

Descriptions of available PODM programs, instructions for their use and examples are contained in this appendix. These programs that are operational on the SUN Workstations cesl and ces3 include:

REQFL	(C.1)
CDREV	(C.2)
GULF	(C.3)
CGULF	(C.4)
PUMPS	(C.5)
PMPRPT	(C.5)
SENAV	(C.6)
GPD	(C.7)
MERGF	(C.8)

C.1 Data File Assignment Routine, REQFL

REQFL is used to assign and stage files containing the operational data for the sensors and time period required by a user. This routine must be executed prior to running PODM data retrieval programs or programs invoking the PODM data retrieval subroutines. REQFL requires less than 170K bytes of memory and only a second of CPU time to execute. However, it could take 10 minutes to transfer each file from CFS. LSTMFD (Appendix B) can be used to determine how many files are required for a given run. To determine if any of the needed files already exist on SUN disk space, enter "ls /tmp/podm."

To run reqfl enter "/apps/sh/reqfl" and press the RETURN key. Then enter the REQFL input. A description of this input is given below in Section C.1.2. When executed in the interactive mode, the program will prompt the user for input.

C.1.1 Files Used by REQFL

REQFL reads the user input described in Section C.1.2 from the standard input file (FORTRAN unit 5) and writes summary information to the standard output device (unit 6). REQFL also reads the master variable file (mvardic) and the master file directory (mfildir) through logical units 39 and 40. The master variable file contains, for each valid sensor, the sensor name, the number of the group to which it belongs, a single digit number identifying the recording data system, and the number used to identify the sensor in the data base. A listing of a small part of mvardic is given in Table C-1. The master file directory contains, for each data file, the information displayed in Table B-1.

The format and content of mfildir and mvardic are described in detail in Reference 1. The location of these files is known to REQFL, and they are automatically assigned.

REQFL writes the list of commands to retrieve needed data files from CFS to file ASG_COM through unit 35. Also, a file, ASG_DATA, giving data directory names and listing basic information about each assigned data file, is written through unit 36. Finally, logical units 41 through 60 are used for assignment of data base files and index files, as needed.

C.1.2 Data Cards for REQFL

Card 1: Access code. All facility and other unclassified data can be accessed by entering the master password in columns 1 and 2. Experimenter-owned data that have been classified as proprietary* can be accessed by entering a code word of the form: XXGGYY, where XX is the master password, GG is a proprietary group number (two digits) and YY is the special two-character password for group GG. Contact the PODM System custodian to obtain the required passwords.

* The control and distribution of data from instrumented tests are discussed in Reference 6 and Section 6 of EI-504.

Card 2: Group numbers or sensors required. Entries are in free-field format separated by one or more blanks or a comma and terminated by $. Entries can be made in columns 1 through 72 only, but any number of cards (lines) may be used. Group number are one or two digits.

Sensor identifiers consist of five or six alphanumeric characters. A mixture of group numbers and sensor identification numbers may be entered. Lists containing available sensors and their group numbers are also available from the PODM System custodian.

Card 3: Beginning and ending dates and times (MMDDYY HHMM, MMDDYY HHMM, where MMDDYY is a six-digit entry for month, day and year and HHMM is a four-digit entry for hours and minutes on a 24-hour clock). In place of the beginning and ending dates and times, card 3 may also contain the entry CURRENT$ if only the latest disk file for each specified group is wanted.
NOTE: Date and time entries should be in Pacific Standard Time.

C.1.3 Examples of REQFL Input

1. To assign files containing data on sensors 21047, 21011A and 21001A recorded between Nov. 3, 1989 at 6AM and Nov. 5, 1989 at 3PM:

```
MK
21047 21011A 21001A$
110389 0600,110589 1500
```

MK in the above example, and in other examples in this appendix, is assumed to be the master password.

2. To assign files containing group 1 data plus data for sensors R10801, R10802, and R10803 recorded on Jan. 30, 1989:

```
MK
1,R10801,R10802,R10803$
013089 0000,013089 2359
```

Card 2 of this example is equivalent to:

```
        1,R10801$
```

since R10801, R10802 and R10803 are in the same group (i.e., on the same file).

3. To assign the latest group 1 and group 2 files:

```
        MK
        1,2$
        CURRENT$.
```

4. To assign the necessary group 1 files and the group 13 proprietary files having a special password LM when data recorded between 0700 and 1300 (PDT) on May 15, 1990, are needed.

```
        MK13LM
        1,13$
        051590 0600, 051590 1200.
```

C.2 The Correlated Data Report Generator, CDREV

The CDREV program can be used in either the interactive or batch mode to produce correlated data reports as described in Section 5.2. Before CDREV can be executed, the REQFL routine must be run to assign the necessary data files. A complete batch run stream involving REQFL and CDREV can be found in Appendix E.

To run CDREV, enter:

```
   /apps/sh/xqt cdrev
```

or

```
   /apps/sh/xqt cdrev < in_file
```

where in_file is a file containing the data described in Section C.2.2.

C.2.1 Files Used by CDREV

CDREV reads the input data described in section C.2.2 from the standard input file (logical unit 5). All output is written to the standard output device (unit 6). CDREV automatically assigns and reads several other files including:

* the ASG_DATA file, created by REQFL, through unit 36,
* a file named eunits through unit 34,
* the mvardic file (described in C.1.1) through unit 39,
* one or more data base files assigned by REQFL.

The eunits file contains a list of valid engineering units and their associated units codes (see Table A-1).

C.2.2 Data Cards for CDREV

The following information is read from the standard input file.

Card 1: Number of columns of output per page. Maximum value is 10. Recommended values are 10 for a full-width line printer, and six for 80 column screens.

Card 2: Beginning and ending dates and times to be included in the report. The format is:

MMDDYY HHMM, MMDDYY HHMM

where MMDDYY is a six-digit entry for month, day and year and HHMM is a four-digit entry for hours and minutes on a 24-hour clock.
NOTE: Date and time entries should be in Pacific Standard Time.

Card 3: Time interval between lines of output. The format is HHMMSS.T where HH is hours, MM is minutes, SS is seconds and T is tenths of a second. The input value, read as a free-field floating point number, must range between 0.1 (1/10 of a second) and 240000.0 (24 hours).

Card 4: List of identifiers for sensors to be included in the report. Entries must be separated by one or more blanks and/or a comma and terminated with $. Only Columns 1 through 72 may contain input. More than one card (line) may be used with any number of sensor identifiers per card.

If more than one report is desired, repeat cards 2, 3 and 4. If CDREV is run interactively, enter a line with 4 zeroes separated by commas or spaces to terminate.

C.2.3 Examples of CDREV Input

1. The input to produce a report on the behavior of sensors 21047, 21011A and 21001A between Nov. 3, 1989 at 6AM and Nov. 5, 1989 at 3PM, with a one-hour time interval between each line in the report is:

```
6
110389 0600,110589 1500
10000.0
21047 21011A 21001A$
```

(assuming REQFL, example 1, has been run to assign the files).

2. The input to produce a report on sensors C00000, C00014, C00010, C00001, R10801, R10802 and R10803 on Jan. 30, 1989 between 10AM and 11AM with a one second time interval between print lines is:

```
6
013089 1000,013089 1100
1.0
C00000,C00014,C00010,C00001,R10801,R10802,R10803$
```

(assuming REQFL, example 2, has been run to assign the necessary files).

3. The input to produce a by-the-minute record on sensors 10001A, 10101 and C10001 from data on the latest disk files is:

```
6
010179 0000,123199 2359
100.0
```

```
        10001A, 10101, C10001$
```

(assuming REQFL, example 3, has been run to assign the disk files). Note, a wide time range on card 1 was used to be assured of getting all the data on the files without having to know what dates/times are included on the latest files.

C.3 Program to Generate User Files, GULF

The GULF program allows the creation of special user-owned files containing selected subsets of the operational data base. A primary application of this program has been in sending data to users in computer readable form. Also, programs such as MERGF and SENAV read GULF generated files. A typical runstream for GULF is:

```
/apps/sh/reqfl << EOI
[Input for REQFL defined in Appendix C.1.2]
EOI
/apps/sh/xqt gulf <<EOI
[Input for GULF.  See Section C.3.2].
EOI
mv senout perm_file
```

where perm_file is the name of the file, in which the GULF output is to be permanently stored (see sections C.3.1, C.3.2 and C.3.3).

C.3.1 Files Used By GULF

GULF reads input data described in section C.3.2 from the standard input file (logical unit 5). Summary information is written to the standard output device (unit 6). However, the main GULF output (i.e, requested sensor records) is written to file senout through logical unit 10. Like CDREV, GULF automatically assigns and reads several other files, including ASG_DATA, mvardic, and one or more data base files.

C.3.2 GULF Input Instructions

The following information is read from the standard input file.

Card 1: Structure of output file. Enter 0, 1 or 2 in Col. 1.

0 = FORTRAN binary file.

1 = Standard FORTRAN formatted (ASCII) file containing one data record per line

2 = Packed ASCII file containing two data records per line.

See section C.3.3 for details on the file formats.

Card 2: Beginning and ending dates and times of data to be included in output file. The format of this card is:

MMDDYY HHMM,MMDDYY HHMM

where MMDDYY is a six-digit entry for the month, day and year and HHMM is a four-digit entry for hours and minutes on a 24-hour clock using Pacific Standard Time.

Card 3: Time interval for output of sensor data (HHMMSS.T, where HH is hours, MM is minutes, SS is seconds and T is tenths of seconds). If the entry, which is read as a floating-point number in free-field format, is less than or equal to 0.1, all data are written. Otherwise, only one record for each sensor in each time interval is written.

Card 4: List of identification numbers for sensors to be included in the output file. Entries should be separated by one or more blanks and/or a comma and terminated with $. Only Columns 1 through 72 may be used. More than one card (line) may be used with any number of sensor identifiers per card.

C.3.3 Format of Output Files

In all cases, the information written for each sensor recording consists of the date and time, the sensor identifier, a floating point value for the sensor, an integer number indicating the units of the floating point value and the DDH&DS status word. The engineering units codes are defined in Appendix A.

The DDH&DS status word in GULF output records is an integer with value ranging from 0 to 65504. To interpret the status word, first convert it to a 16 bit binary number. Label the 16 bits from 1 to 16 with the most significant (leftmost) bit being 1. Then refer to Table A-2 in Appendix A to interpret the status bits. The bit numbers, as identified above, are listed in column 1 of this table.

As an example, suppose the status word in a GULF file has a value of 12288. The binary representation of this number is:

```
  0011000000000000
  ↑              ↑
bit 1          bit 16
```

In this is case, bits 3 and 4 are set. From Table A-2 it can be seen that these bits indicate the occurrence of invalid readings.

Use of the "0" option for data output results in the generation of a standard FORTRAN binary file. Each logical record described in Table C-1 contains the data for one sensor at one time period. Use of the "1" option results in the generation of a standard FORTRAN formatted (ASCII) output file. Each record (line) consists of 41 characters as shown in Table C-2. Use of the "2" option also results in the generation of an ASCII file. However in this case, two sensor records are stored in each character line, as shown in Table C-3.

Table C-1

Content of Gulf Records Written to a FORTRAN Binary File* (Option 0)

Word	Content
1	The date, a six-digit integer value (YYMMDD, where YY is the year, MM is the month, and DD is the day)
2	The time of day to the 1/10 second, a seven digit integer (HHMMSST, where HH is the hours on a 24-hour clock, MM is the minutes, SS is the seconds and T is the tenth of seconds)
3	The sensor identifier (6 characters)
4	Sensor value in engineering units (a floating point word)
5	Units code (see Appendix A)
6	The 16-bit DDH&DS status word right justified in the 32-bit integer word

* The WRITE statement used to create the FORTRAN binary files is of the form:

```
WRITE(8) NDT,NTM,IDSEN,VAL,IUNITS,ISTAT
```

Table C-2

Content of Gulf Records Written to a FORTRAN Formatted File* Using Option 1

Characters	Content
1 - 6	The date (YYMMDD, where YY is the year, MM is the month and DD is the day)
7	blank
8 - 14	The time of day to the 1/10 second (HHMMSST, where HH is the hours on a 24-hour clock, MM is the minutes, SS is the seconds, and T is the tenths of seconds)
15	blank
16 - 21	Sensor identifier
22	blank
23 - 32	Sensor value in engineering units. The floating point format is: ±x.xxxE±yy
33	blank
34 - 35	The units code (see Appendix A)
36	blank
37 - 41	The integer value of the 16-bit DDH&DS status word (see Appendix A)

* The WRITE and FORMAT statements used to create the formatted files are:

```
      WRITE(10,10) NDT,NTM,IDSEN,VAL,IUNITS,ISTAT
10    FORMAT(I6,1X,I7,1X,A,1X,1PE10,3,1X,I2,1X,I5)
```

Table C-3

Content of GULF Records Written to a Packed ASCII File Using Option 2

Columns	Content
1 - 6	The date (YYMMDD, where YY is the year, MM is the month and DD is the day)
7 - 13	The time of day to the 1/10 second (HHMMSST, where HH is the hours, MM is the minutes, SS is the seconds and T is the tenths of seconds)
14	blank
15 - 20	Sensor identifier
21 - 30	Sensor value in engineering units. The floating point format is: ±x.xxxE±yy.
31	blank
32 - 33	Units code (see Appendix A)
34	blank
35 - 39	Integer value of the 16-bit DDH&DS status word (see Appendix A)
40	blank
41 - 80	Same as col.s 1 - 40 for the second sensor record on a card image

C.4 Program to Produce Files of Time-Averaged Data, CGULF

The function of CGULF is to produce files containing time-average sensor data. As with the GULF program (Appendix C.3), the total time span to be covered and the frequency at which sensor records are written out are user specified. The output frequency in CGULF determines the time step between successive calculations of sensor averages. In addition, the length of averaging interval must be specified. For example, if the output frequency is set to one hour and the averaging interval is set to five minutes, all sensors specified will be averages over a five minute period each hour.

Other CGULF input items include, for each sensor, the type of data validity checking to be performed and the DDH&DS scan interval. The data checking options are included because a single bad sensor signal will result in an erroneous average if it is not screened out.

Three options are available:

- no data checking
- use of the data validity flags in the DDH&D status word (discussed in Appendix A)
- input of an acceptable range for a sensor reading

When the third option is selected, upper and lower limits must be read in. If a sensor value retrieved from the data base does not fall between these limits, that value is rejected and not used in the sensor averaging process.

When the second option is selected, the DDH&DS status word in each retrieved sensor record is checked. If either of the validity flags (bits 3 and 4) are set, the record is rejected. Also, records of PDS origin are rejected if either the "processing" bit (7) or their "I/O active" bit (11) is not set to 1.

If the first option is specified, all sensor records retrieved from the data base are used in computing time averages.

In most CGULF runs, a number of sensors will be called for. Since data checking option is specified for each sensor, more than one option can be specified in a single run.

The DDH&DS scan rates are used only in the computation of the standard deviations. The scan rates are needed because the DDH&DS uses a data compression scheme to reduce the volume of data recorded. If a value acquired by the DDH&DS for a particular sensor differs from the last recorded value for that sensor by less than a specified amount (a compression window), the current value is not recorded. Each sensor is, however, recorded at least once per hour, even if it is constant. The compression windows, which normally range from 0.2% to 3% of full scale, are listed in the variable description table available from the PODM System custodian. This table gives the sensor scan rates needed to compute the standard deviations. Since small data fluctuations are not recorded due to data compression, the standard deviations computed will normally be slightly low.

A typical CGULF runstream is given below:

```
/apps/sh/reqfl << EOI
[data cards for REQFL defined in Appendix C.1.2]
EOI
/apps/sh/xqt cgulf << EOI
[data cards for CGULF.  See Section C.4.2].
EOI
mv senout perm_file
```

C.4.1 Files Used By CGULF

CGULF used the same files as GULF (see section C.3.1).

C.4.2 Input Instructions

Card 1: Structure of output file. Enter 0, 1 or 2 in Col. 1.

0 = FORTRAN binary file.

1 = Standard FORTRAN formatted (ASCII) file.

2 = Packed ASCII file containing two sensor records per line. (See section C.4.3.)

Card 2: Beginning and ending dates and times of data to be included in output file. The format of this card is:

MMDDYY HHMM, MMDDYY HHMM

where MMDDYY is a six-digit entry for the month, day and year and HHMM is a four-digit entry for hours and minutes on a 24-hour clock using Pacific Standard Time.

Card 3: Two entries in free field format.

Item 1 -- Frequency for computing averages and writing sensor data (HHMMSS.T, where HH is hours, MM is minutes, SS is seconds and T is is tenths of seconds).

Item 2 -- Length of each averaging interval (HHMMSS.T). This time interval must be between 0.1 seconds and the time step specified in item 1.

Card 4: List of identification numbers for sensors to be included in the output file. Entries should be separated by one or more blanks and/or a comma and terminated with $. Only Cols. 1 through 72 may be used. More than one card (line) may be used with any number of sensor identifiers per card. The number of sensors must be less than 800.

Card 5: The DDH&DS scan rate (seconds) for each sensor given in Card 4 (decimal numbers in free-field format).

Card 6: The type of data rejection criteria to be used for each sensor given in card 4 in free field integer format.

0 = No test (accept all data records)

1 = Reject all data records flagged in the DDH&DS status word as being invalid

2 = Read sensor limits and reject all sensors values outside these limits

Card 7: The low limit for data acceptability. One entry is required for each sensor having a type 2 data rejection criterion on Card 5. Data are read in free-field floating point format.

Card 8: The high limits for data acceptability. One entry is required for each sensor having a type 2 data rejection criterion. Data are read in free-field floating point format.

Note: Cards 7 and 8 are read only if type 2 data checking is specified for one or more sensors.

C.4.3 Format of Output Files

The content of CGULF output files is the same as GULF files (section C.3.3) except that:

1) the status word is set to zero in all cases, and
2) the standard deviation of sensor data is added to the output records if FORTRAN binary or standard FORTRAN formatted files are specified (options 0 or 1).

CGULF records then consist of:

1) the date and time at the beginning of an averaging interval
2) sensor identifier
3) a time averaged sensor value in engineering units
4) the units code
5) a zero entry in place of the status word to preserve compatibility with GULF records
6) the standard deviation of sensor data over the time interval

As in GULF, there are three options for the format of the output files: FORTRAN binary, Standard ASCII, and packed ASCII. Because the formats available in CGULF are nearly identical to those in GULF, only the differences are identified here.

A GULF Binary output record consists of six words as given in Table C-1. CGULF binary records contain these six words plus a seventh word representing the standard deviation. A GULF standard ASCII record consists of the 41 characters described in Table C-2, plus the standard deviation in character positions 42 through 52. A CGULF packed ASCII record does not contain a standard deviation entry. Thus, a CGULF packed ASCII file is identical in format to a GULF packed ASCII file (Table C-3).

Even though there are differences between CGULF and GULF output files, programs that required GULF-format files as input (e.g. SENAV, GPD, and MERGF) also will read CGULF files without any difficulty.

C.5 Programs to Summarize Pump Motor Operations, PUMPS and PMPRPT

The PUMPS program summarizes the operating times of the primary and secondary loop pumps in seven selected speed (RPM) ranges. The program assumes that the first range specified defines the pump speeds expected when the pump motors are not operating. This range typically is from a small negative number (e.g., -10 RPM) up to the lower limit expected when a pump is operating on its pony motor.

The second speed range is assumed to define the pumps speed range when operating on pony motors. The third range is assumed to be an interval of undefined operation between pony and main motor speed ranges. The lower limit of this range should be set to the upper limit of pump speed while operating on the pony motor. The upper limit of this interval should be the lower limit of pump speed on the main motor.

The last four ranges are assumed to encompass pump speed range on the main motor. A complete set of ranges is read for each pump.

A typical runstream for PUMPS is:

```
cd /tmp/podmwd              #Change to a working directory
[Commands to copy pmphis (see Section C.5.1) into the working directory]
cp pmphis pmphis.old        #Save copy of pmphis before updating
/apps/sh/reqfl  << EOI
[Input for REQFL defined in Section C.1.2]
EOI
/apps/podm/bin/pumps.x  << EOI
[Input for PUMPS.  See Section C.5.2]
EOI
nl pmphis                   #list updated pmphis file
[Commands to copy pmphis back to a permanent storage location.]
```

C.5.1 PUMPS Files

PUMPS reads the input data described in Section C.5.2 from the standard input device. An ASCII file, pmphis, is also read and updated. This file is used to accumulate pump operating data (the total operating time in each speed range for each pump). The format of pmphis is given in Table C-4. Updates made to pmphis during a PUMPS run are based on data records (pump speed versus time) read from one or more operational data files assigned by REQFL. PUMPS rejects records read from the operational data files if they are flagged as invalid by the DDH&DS.

The PUMPS standard output (print) file lists the data read from pmphis, and the results after adding in the data from the operational data files.

C.5.2 PUMPS Input

PUMPS reads the following input from the standard input device. Table C-5 lists the input for a typical case.

Card 1: Beginning and ending dates and times of data to be processed. The format of this card is:

```
MMDDYY HHMM,MMDDYY HHMM
```

As in Card 2 of Sections C.2.2, C.3.2, and C.4.2.

Card 2: Data retrieval time interval. The format is HHMMSS.T, as in Card 3 of Sections C.2.2 and C.3.2. A typical value is 500.0.

Card 3: List of identifiers for pump speed indicators, separated by one or more blanks and/or a comma, and terminated with $. The usual list of six identifiers is given in Table C-5.

Card 4: Lower and upper RPM limits of each of the seven speed ranges for the pump associated with the first identifier on Card 3. Seven pair of floating point numbers are read in free-field format. Repeat Card 4 for the other identifiers on Card 3.

Normal entries for Card 4 are given in Table C-5. Note that there are slightly different entries for the primary and secondary loop pumps.

Card 5: Pump motor associated with each of the seven speed ranges. Each entry begins on a separate line starting in column 1. See Table C-5 for the normal Card 5 input.

C.5.3 Description of PMPRPT

PMPRPT reads the pmphis file through the standard input device and produces a one-page summary report of pump motor operating times. After reading pmphis, PMPRPT will optionally read one or more records of text information that is included in the output Table as foot notes. The summary report is written to the standard output file. A typical UNIX script for a PMPRPT run is given in Table C-6.

Table C-4

Format of File pmphis

Record	Description
1	Ending date and time of most recent PUMPS run. The format, starting in column 1, is: YYMMDD HHMMSST where YY, MM, and DD are 2 digit entries for the year, month, and day, respectively, and HH, MM, SS, and T is the time (hours, minutes, seconds, and tenths of a second, respectively).
2	Total operating times (hours)[a] of pump 1 motors (6 entries in free field format): 1. Pony motor total 2. Main motor, first speed range 3. Main motor, second speed range 4. Main motor, third speed range 5. Main motor, fourth speed range 6. Main motor, total[b] Record 2 is repeated for each of the other pumps.
3	Status code for each pump at the end of the most recent PUMPS run. The status codes (in free field format) can have the following values: 0 - no valid records read 1 - motors not operating 2 - pony motor operating 3 - RPM in undefined range 4 - main motor operating in range 1 5 - main motor operating in range 2 6 - main motor operating in range 3 7 - main motor operating in range 4 8 - Pump overspeed

[a] PUMPS writes operating times to the nearest 0.01 hours.

[b] Because of initialization of the pmphis file, the sum of main motor operating times in the four speed ranges may not equal the main motor total hours.

Table C-5

Typical PUMPS Input

```
040193 0000 010199 0000
500.0
21047,22047,23047,21538,22538,23538$
-10.0 70.0  70.0 100.0  100.0 500.0  500.0 700.0  700.0 900.0
 900.0 1000.0  1000.0 1110.0
-10.0 70.0  70.0 100.0  100.0 500.0  500.0 700.0  700.0 900.0
 900.0 1000.0  1000.0 1110.0
-10.0 70.0  70.0 100.0  100.0 500.0  500.0 700.0  700.0 900.0
 900.0 1000.0  1000.0 1110.0
 -10.0 90.0  90.0 110.0  110.0 500.0  500.0 700.0  700.0 900.0
 900.0 1000.0  1000.0 1110.0
 -10.0 90.0  90.0 110.0  110.0 500.0  500.0 700.0  700.0 900.0
 900.0 1000.0  1000.0 1110.0
 -10.0 90.0  90.0 110.0  110.0 500.0  500.0 700.0  700.0 900.0
 900.0 1000.0  1000.0 1110.0
OFF
PONY
UNDF
MAIN
MAIN
MAIN
MAIN
```

Table C-6

Typical PMPRPT Runstream

```
cd /tmp/podmwd
cp pmphis PMPRPT.INP
cat >> PMPRPT.INP <<EOI
Operating times given for each of the four main motor speed ranges include
only data since 3/13/81.  Main and pony motor totals are for the lifetime
of the pumps.

On 6/3/82 at 1430 hrs PDT, P1 main motor replacement was completed.  At
that time, P1 main motor operating totals were:

    500 -  700 RPM =   16.39 HRS
    700 -  900 RPM =  162.45 HRS
    900 - 1000 RPM = 1698.77 HRS
   1000 - 1100 RPM =  950.35 HRS
        MAIN TOTAL = 9194.93 HRS
EOI
/apps/podm/bin/pmprpt.x < PMPRPT.INP
rm PMPRPT.INP
```

C.6 Program for Averaging Sensor, SENAV

SENAV is used to average data from two or more (usually redundant or similar) sensors. The results are written to a file, senout, in GULF format. SENAV reads the data described in section C.6.1 from the standard input file (unit 5). Sensor records to be averaged are read from a single GULF format file, seninp.

A typical SENAV runstream includes the following:

```
[Instruction(s) for linking or copying file seninp into the user's
working directory.]
/apps/sh/xqt/senav << EOI
[SENAV input data described below]
EOI
mv senout perm_file
```

The last line in the runstream is an example of one way to move (rename) the output file, senout, to a permanent file.

Input Data

Card 1: Four entries in free field format:

1) Number of averages to be made (≤ 30)
2) Input file type (0 = binary, 1 = FORTRAN formatted)
3) Output file type (0 = binary, 1 = FORTRAN formatted)
4) Output option (0 = write out only the averages, 1 = write out the averages plus all sensor data used in computing the averages)

Card 2: The identifiers for each average. Identifiers must be six characters or less in length separated by a comma or space and terminated by "$". Only columns 1-72 may be used, but multiple lines are permitted. The number of identifiers expected is the value of entry 1 on Card 1.

Card 3: The lower acceptance limit for the sensor data used in computing each average, in free field format. One entry is read for each average called for on card 1, entry 1.

Card 4: The upper acceptance limit for sensor data used in computing each average. If a sensor value is outside the acceptance range, that input record is rejected and not used in the averaging process.

Card 5: The identifiers for all sensors to be used in computing the first average. Entries must be separated by a comma or space and terminated with "$".

Repeat Card 5 for all averages called for on Card 1, entry 1.

C.7 Program to Generate Plot Files, GPD

GPD was designed to read a user file containing sensor data and generate an input file for a plot program that is not available on SUN Workstations. GPD may still be useful for generating files for importing to a spreadsheet or general purpose plot program. GPD reads data from a GULF-format file (seninp) and generates a file (pdatout) for one or more plots, each containing up to nine curves.

GPD Input Instructions

The following information is read from the standard input device (unit 5).

Card 1: Beginning date and time on the plot (MMDDYY HHMM) where MMDDYY is a six digit number representing the month, day and year, and HHMM is a four digit number representing the time (hours and minutes) on a 24-hour clock.

Card 2: Length of plot in units of seconds, minutes, hours or days (for example, 7 DAYS). A space must appear between the numerical value and the units. The time units may be abbreviated as follows:

SECONDS - SEC, S
MINUTES - MIN, M

HOURS	- HRS, H
DAYS	- DYS, D

Card 3: Four entries in free field format.

1) The number of points to be plotted for each sensor. If a value of 0 is entered all available data will be plotted.
   If a positive value, N, is entered the time axis will be divided into N equal intervals and all available data in each interval will be averaged to obtain a single data point for each interval. If no input data points exist in an interval, the last input value read will be used for that interval. A maximum of 1000 points can be plotted.

2) Input file type (0 = binary, 1 = FORTRAN formatted) as described in C.3.3.

3) Plot type (not used - enter 0).

4) Data rejection criterion (0 = accept all data, 1 = reject all data flagged by the DDH&DS as being invalid, 2 = reject data that are off-scale, 3 = read validity limits for each sensor to be plotted.

Card 5: Identifiers for each sensor to be included on this plot. Identifiers must be separated by a comma and/or one or more spaces and terminated with "$". More than one line may be used. Columns 73 - 80 are ignored.

Card 6: Read only if entry 4 on card 3 was "3". The lower and upper validity limits for each sensor in free field format. Limits are read in pairs for each sensor. If a sensor value is outside the validity limits, it is ignored.

Card 7: Plot title -- 40 characters or less terminated with "$".

Card 8: Plot subtitle -- 40 characters or less terminated with "$". Up to three subtitles can be entered.

C.8 Program for Merging or Concatenating GULF Format Files, MERGF

MERGF provides the means to merge or concatenate GULF-format files. The UNIX cat utility can also be used to concatenate ASCII files, but not binary files. MERGF can also be used to change the format of a GULF file (e.g. from binary to ASCII). In some large data retrieval tasks, it is impossible to complete the job in a single run because the required data are spread over a large number of data base files. A solution to this problem is to make a series of GULF runs to retrieve all required data, and then run MERGF to merge the separate GULF files onto a single file.

C.8.1 Files Used by MERGF

MERGF writes sensor records to a GULF-format file, senout, through logical unit 10. Up to 10 GULF-format files can be read in a single run. A typical MERGF runstream includes the following sequence of control statements and data entries:

```
[Instruction(s) for assigning first input file to senin1]
[Instruction(s) for assigning second input file to senin2]
              •
              •
              •
/apps/sh/xqt/ mergf << EOI
[data cards defined in C.8.2]
EOI
mv senout perm_file
```

C.8.2 Input Instructions

MERGF input is consistent, as far as possible, with GULF input. Some details associated with the input items listed below are omitted here, but can be found in C.3.2. The following information is read from the standard input device.

rd 1: Type of each input file in free field format (0 = binary, 1 = FORTRAN formatted). Enter one value (0 or 1) for each input file.
Note: packed ASCII files cannot be read.

ard 2: Output file type (0, 1 or 2) in free field format. (0 = binary, 1 = standard FORTRAN formatted, 2 = packed ASCII)

ard 3: Beginning and ending date and time of data to be retrieved (MMDDYY HHMM, MMDDYY HHMM)

APPENDIX D

Instructions for Using the PODM Subroutine Library, PODMLIB

A subroutine library, podmlb.o, maintained in SUN directory /apps/bin is available to facilitate retrieval, encoding, decoding, and processing of information from the operational data base by a user's program. All routines in the library were written in FORTRAN 77. The library can be obtained by adding /apps/bin/podmlb.o to the list of parameters passed to the FORTRAN computer (f77). For example, the command:

```
f77 -o gulf.x -03 gulf.f /apps/bin/podmlb.o
```

would compile source file gulf.f, link the PODMLIB library and create an executable file gulf.x.

D.1 Data Retrieval Subroutines

Four subroutines are described in this section. Three (OPENDF, INITFO, and GETMSO are required in order for a FORTRAN program to retrieve information from the operational data base. The fourth, SKIT, which is optional, was designed to efficiently locate sensor data when relatively long retrieval intervals (> 5 minutes) are called for.

Before a program can access the operational data base, REQFL must be run to assign the necessary data files. Subroutine OPENDF is called once to open the data files assigned by REQFL. Subroutine INITFO positions the data files assigned by REQFL to a specified initial date and time. Generally, INITFO, which must be called after OPENDF, is referenced only once in a program, but can be called whenever the files need to be repositioned to a new initial date and time.

Subroutine GETMSO controls the actual data retrieval. Each call to GETMSO, which is supplied with a list of required sensors, results in one sensor record being returned, unless there are no more data. Sensor records are returned strictly in chronological order rather than being ordered by sensor. Any required sorting of the retrieved data by sensor is left to the user.

Subroutines OPENDF, INITFO and GETMSO and other PODMLIB routines that they call require 36k bytes.

Subroutine SKIT positions the assigned operational data files to a specified date and time. It is generally called after all data for one time interval have been retrieved. SKIT is then used to position the files at the start of the next interval. It is not necessary to use SKIT, but it can greatly reduce retrieval time, particularly with retrieval intervals of one hour or more. Its use is recommended for retrieval intervals over five minutes.

All parameters listed in the following sections are 32 bit integer or floating point words unless specified otherwise.

D.1.1 Subroutine OPENDF

Function
OPENDF reads a communication file ASG_DATA created by REQFL and opens the files assigned by REQFL.

Usage
CALL OPENDF(JPR).

Parameters
JPR Flag indicating if a summary of assigned files is to be printed. If JFLAG = 1, the summary is printed; otherwise, it is not.

PODMLIB routines called
CLEAR, CLEARC, DECODT, NORMDT

D.1.2 Subroutine INITFO

Function

The primary function of INITFO is to position the data files to a specified initial date and time. It also performs initialization and setup task. OPENDF must be called prior to INITFO.

Usage

CALL INITFO (JITIM, JLTIM, JNSEN, JISEN).

Parameters

JITIM Initial date and time for data retrieval coded in data base format.

JLTIM Ending date and time for data retrieval coded in data base format.

JNSEN Number of sensors for which data are to be retrieved.

JISEN An array containing the 5 or 6 character identifier name of each sensor.

Notes

The initial and final dates and times (JITIM and JLTIM) must be coded into the operational data base format prior to calling INITFO. PODMLIB routines NCODT (Appendix D.2.2) is available to perform the necessary encoding.

PODMLIB routines called

CA2SN, CLEAR, KRONO, LHASH, POSIT, PREPDF

D.1.3 Subroutine GETMSO

Function

GETMSO accesses the assigned data files and returns a record for one sensor from the list specified in the call to INITFO. Each call to GETMSO results in the next requested record being returned in chronological order.

Usage

CALL GETMSOMREC,ISQ)

Parameters

MREC Four-word array (32 bits/word) containing the returned sensor record. The content of each 32-bit word is:

1. The date, formatted as described in Table D-1.
2. The time - the number of 1/10 seconds since midnight (0-863,999).
3. The sensor value in engineering units (an IEEE floating point word).
4. The units code, status bits and sensor number (see Table D-2).

ISQ An integer value returned by GETMSO indicating the position in the sensor list occupied by the sensor for which data were returned in MREC. (The sensor list is defined in a call to INITFO.)

Notes

The first call to GETMSO must be preceded by calls to OPENDF and INITFO. Because the ordering of data records returned on successive calls to GETMSO is strictly chronological, the sensor returned on a particular call could be any one in the list specified in the call in INITFO. The identity of the record can be determined from the ISQ parameter. It can also be determined by decoding MREC(4).

If a program continues calling GETMSO until no more data are left on the file, MREC(1) and MREC(2) are set to FFFF base 16 (65,535 decimal) and MREC(2) is set to zero.

PODMLIB routines call
GETNXØ, KRONO

D.1.4 Subroutine SKIT

Function
SKIT positions assigned operational data files to a specified date and time. Before SKIT can be used, OPENDF and INITFO must be called.

Usage
CALL SKIT (NEXT)

Parameters
NEXT Two word array containing the date and time to which the data files are to be positioned. This date and time are coded as described in Section D.1.3 and Table D-1.

PODMLIB routines called
FREEFL, KRONO, POSIT, SWIFL

D.2 Other PODMLIB Routines

In addition to the data retrieval routines described in Section D.1, there are a number of other PODMLIR routines that are essential or may be useful in writing programs related to the PODM System. For example, NCODT, described in D.2.1, is required by programs that actually retrieve sensor records from the operational data base.

The following sections also describe other useful PODMLIB routines. Not all routines in PODMLIB are described below, or in D.1. These routines are called by other PODMLIB routines, and hence program designers normally do not need to know what they do, or how to use them. However, if needed, all PODMLIB routines are listed in Reference 1.

D.2.1 Subroutine CA2SN

Function
Given a list of sensor identifers, CA2SN returns a list of corresponding sensor numbers (integers) obtained from the mvardic file described in Ref. 1.

Usage
CALL CA2SN(IDA, ISN, NSEN)

Parameters
NSEN Number of sensor identifiers in IDA.

IDA List of sensor identifiers. IDA is an array of six character variables.

ISN List of sensor numbers (integers) corresponding to each entry in IDA. ISN is a one-dimensional integer array with NSEN entries.

Notes
Data retrieval subroutines require the integer sensor numbers. INITFO calls CA2SN to generate this list given a list of user-specified sensor names.

D.2.2 Subroutine NCODT

Function
NCODT codes a date and time into an internal PODM date and time word.

Usage
CALL NCODT(IYR,IMO,IDY,IHR,IMI,SEC,NDT)

Parameters

NDT Two word array word containing the date and time coded in the internal PODM format (Section D.1.3 and Table D-1) by NCODT.

IYR Year

IMO Month

IDY Day of month

IHR Hours on a 24-hour clock

IMI Minutes

SEC Seconds (a floating point word)

Notes

The INITFO subroutine required for data retrieval has as two of its parameters, the beginning date and time and the ending date and time for data retrieval. The date and time parameters must be in internal PODM representation. NCODT can be used to make the necessary conversions.

D.2.3 Subroutine DECODT

Function

DECODT decodes an internal PODM date and time word into six words (year, month, day, hours, minutes and seconds).

Usage

```
CALL DECODT (NDT,IYR,IMO,IDY,IHR,IMI,SEC)
```

Parameters

NDT Internal PODM representation of the date and time to be decoded (a two word array).

IYR Year

IMO Month of year

IDY Day of month

INR Hours on a 24-hour clock

IMI Minutes

SEC Seconds to 0.1 seconds (floating point word)

(IYR,IMO,IDY,IHR,IMI and SEC are the parameters computed by DECODT)

D.2.4 Function KUNITS

Function

KUNITS is used to extract the code for the engineering units of a sensor from the fourth word of a retrieved sensor record. The units codes are defined in Appendix A.

Usage

```
KEU = KUNITS(MREC(4))
```

Parameters

KEU Engineering units code.

MREC Array containing a sensor record returned by GETMSO. The engineering units code is embedded in the fourth word of this array.

D.2.5 Function ICOND

Function

ICOND returns the status of a sensor record.

Usage

ST=ICOND(MREC(4))

Parameters

ST Status of sensor record (a character*1 variable). ICOND returns one alphanumeric character. The possible status characters and their meanings are:

'*' = invalid signal
'H' = high alarm limit exceeded.
'L' = sensor value is under the low alarm limit
'N' = increased scan rate
' ' = status normal

MREC Array containing a sensor record. The status bits are embedded in the fourth word of the record.

D.2.6 Function ISTATWD

Function

ISTATWD returns an integer word containing the 16 status bits.

Usage

IST = ISTATWD(MREC(4))

Parameters

IST an integer word containing the 16 status bits right justified.

MREC Array containing a sensor record. The status bits are embedded in the fourth word of this array.

D.2.7 Subroutine SENREC

Function

SENREC returns a sensor value, units code and status word given a coded sensor record.

Usage

Call SENREC (MREC(3), VAL, IUN, ISTT)

Parameters

VAL Sensor value in engineering units (a 32-bit IEEE floating point word)

IUN Engineering units code

IST 16 status bits right justified

MREC Array containing a coded sensor record. The sensor value in the third word of this array, and the units code and status bits are embedded in the fourth word.

D.2.8 Function IVALID

Function

IVALID determines from a 16-bit sensor status word whether a sensor record is valid or not.

Usage

L = IVALID (IST)

Parameter

IST An integer word containing the 16 status bits right justified, as obtained using ISTATWD (D.2.6) or SENREC (D.2.7).

L Set to 1 if the status bits indicate an invalid signal, and 0 otherwise.

D.2.9 Subroutine NEXTIM

Function

NEXTIM adds a time increment to an internal PODM formatted date and time word to yield another internal PODM formatted date and time (Table D-1).

Usage

CALL NEXTIM(NDT,IH,IM,DS,LDT)

Parameters

NDT Two element base date and time in internal PODM format.

IH Time increment, nonnegative integer hours.

IM Time increment, nonnegative integer minutes.

DS Time increment, nonnegative floating point seconds.

LDT Array containing the result of adding IH hours, IM minutes and DS seconds to date and time NDT. LDT is in internal PODM format.

D.2.10 Subroutine IPAST

Furction

IPAST subtracts a time increment from an internal PODM formatted date and time to yield another internal PODM formatted date and time.

Usage

CALL IPAST(NDT,IH,IM,DS,LDT)

Parameters

NDT Two element array containing the base date and time in internal PODM format.

IH Time increment, nonnegative integer hours.

IM Time increment, nonnegative integer minutes.

DS Time increment, nonnegative floating point seconds.

LDT Result of subtracting IH hours, IM minutes, and DS seconds from NDT. LDT is in internal PODM format.

D.2.11 Subroutine CLEAR and CLEARC

Function

CLEAR is used to initialize a numeric array to one specified value. CLEARC is used to initialize a character array to one specified character string.

Usage

```
CALL CLEAR(X,Y,N)
CALL CLEARC(X,Y,N)
```

Parameters

Y The array to be initialized.

N The length of array Y.

X A constant to which array Y is to be initialized.

D.2.12 Function KRONO

Function

KRONO determines the chronological ordering of two internal PODM formatted date and time arrays.

Usage

```
K = KRONO (T1, T2)
```

Parameters

T1,T2 Two element arrays each containing an internal PODM formatted date and time.

K If T1 is chronologically after T2, K is set to 1.
If T1 is chronologically before T2, K is set to -1.
If T1 = T2, K is set to 0.

D.2.13 Function TDIF

Function

TDIF computes the difference (in seconds) between two given times. The result is real.

Usage

S = TDIF(LDATE, TIMEL, IDATE, TIMEI)

Parameters

LDATE Ending date expressed as a six-digit integer YYMMDD where YY is the year, MM is the month and DD is the day.
LDATE = DD+100*MM+10000*YY.

TIMEL Ending time expressed as a floating point (real) number.
TIMEL = S+100*M+10000*H
where S is the number of seconds, M is the minutes and H is the hour on a 24-hour clock.

IDATE Starting date in same form as LDATE.

TIMEI Starting time in same format as TIMEL.

If the specified ending date and time is earlier than the starting date and time, the result will be negative.

D.2.14 Subroutine VERSN

Function

VERSN returns the version number and release date of the PODMLIB library.

Usage

```
CALL VERSN(VSN, RELS)
```

Parameters

VSN The version number (a 6 character variable).

RELS The release date (an 8 character variable).

Table D-1

Internal PODM Representation of a Date

Bits*	Content
1 - 17	0 (not used)
18 - 23	The year biased by 1976 (i.e., add 1976 to this value of these bits to get the actual year)
24 - 27	The month (1-12)
28 - 32	The day of the month (1-31)

Table D-2

Content of Fourth Word of Data Record Returned by GETMSO

Bits*	Content
1 - 2	Not used
3 - 9	Units code (see Table A-1)
10 - 20	11 most significant status bits in Table A-2 (bits 12-16 are not used, so are not stored).
21 - 32	Sensor number (an integer converted from ASCII by PODMLIB subroutine CA2SN.

* Bit no. 1 is the leftmost bit in the 32 bit word.

APPENDIX E

Sample REQFL/CDREV Run in the Batch Mode

Table E-1 contains a listing of a runstream for a typical CDREV case. This runstream can be used as a guide in setting up other cases. Changes that users need to make for their own runs are described below.

<u>Lines 1 and 6</u>

- The runstream in Table E-1 directs all output to a file cdrev.pr. Users may want to specify a different file, or no file.

<u>Line 2</u>

- Enter a valid access code. "MK" is only an example.

<u>Line 3</u>

- Enter the sensor identification numbers required and/or numbers of groups containing required sensors; terminate list with $. (See REQFL input instruction in Appendix C.1)

<u>Line 4</u>

- Enter starting and ending date and time for your report. Or, enter CURRENT$ if only the latest data are required. (See REQFL input instructions in Appendix C.1.)

<u>Line 5</u>

- No changes necessary.

<u>Card 6</u>

- Changes described above.

<u>Card 7</u>

- No changes necessary if output (cdrev.pr) is routed to a device having at least 128 character/line capability.

Line 8

- Enter starting and ending date and time for your report. (See input instructions for CDREV in Appendix C.2.)

Line 9

- Enter the time interval between lines in the report. In the sample case, one hour is specified. (See input instructions for CDREV in Appendix C.2.)

Line 10

- Enter sensor identification numbers required. Terminate list with $.

Line 11

- No change necessary.

Table E-1

Sample CDREV Runsteam

```
 1.   /apps/sh/reqfl > cdrev.pr <<EOI
 2.   MK
 3.   1,2$
 4.   031892 0000,031992 2359
 5.   EOI
 6.   /apps/sh/xqt cdrev >> cdrev.pr <<EOI
 7.   10
 8.   031892 0000,031992 2359
 9.   10000.0
10.   C00000,21047,21011A,21011B,21014,21003A,21042,21001A$
11.   EOI
```

APPENDIX F

IVHM Parameters Stored in the Data Base

Sensor Identifier Description

IVHMGE Grapple elevation in inches

IVHMGF Grapple force in pounds

IVHMSN Subassembly serial number (Serial numbers are generally not available on most drives and therefore are frequently zero in the data base)

IVHMOC Operation code used by IVHM to identity the type of drive:

+400 = UUR (up unloaded restricted)
-400 = DUR (down unloaded restricted)
+500 = UUN (up unloaded)
-500 = DUN (down unloaded)
+600 = ULS (up loaded storage)
-600 = DLS (down loaded storage)
+700 = ULC (up loaded core)
-700 = DLC (down loaded core)

IVHMSQ IVHM move sequence number

IVHMCP Four digit number indicating the core lattice position, IVS cell or transport that the grapple is over. For core positions, the normal four-digit identifiers are used. However, for other cells and ports, special four-digit codes were created. These codes are listed below.

Cell or Port	Data Base Identifier	Cell or Port	Data Base Identifier	Cell or Port	Data Base Identifier
xA	x001	xH	x008	xU	x015
xB	x002	xJ	x009	xV	x016
xC	x003	xK	x010	xW	x017
xD	x004	xL	x011	xX	x018
xE	x005	xM	x012	xY	x019
xF	x006	xR	x013	xFTP	x020
xG	x007	xS	x014	xCRP	x021

where, x is the trisector number (1, 2 or 3),
xA through xY are in-vessel storage (IVS) cells,
FTP = fuel transport port,
CRP = core restraint pin.

DATE
FILMED
10/5/94

END