Do Not
MF

DOE/MC/16050-1120
(DE82013299)

SYSTEM DESIGN OF THE METC AUTOMATIC DATA
ACQUISITION AND CONTROL SYSTEM

By
David R. Goff
Dan Armstrong

February 1982

Work Performed Under Contract No. AM21-81MC16050

Morgantown Energy Technology Center
Morgantown, West Virginia

and

TRW Energy Engineering Division
Morgantown, West Virginia

TECHNICAL INFORMATION CENTER
UNITED STATES DEPARTMENT OF ENERGY

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

"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."

This report has been reproduced directly from the best available copy.

Available from the National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia 22161.

Price: Printed Copy A09
Microfiche A01

Codes are used for pricing all publications. The code is determined by the number of pages in the publication. Information pertaining to the pricing codes can be found in the current issues of the following publications, which are generally available in most libraries: *Energy Research Abstracts*, (ERA); *Government Reports Announcements and Index* (GRA and I); *Scientific and Technical Abstract Reports* (STAR); and publication, NTIS-PR-360 available from (NTIS) at the above address.

DOE/MC/16050--1120

DE82 013299

SYSTEM DESIGN OF THE METC
AUTOMATIC DATA ACQUISITION AND CONTROL SYSTEM

BY

DAVID R. GOFF
MORGANTOWN ENERGY TECHNOLOGY CENTER
MORGANTOWN, WEST VIRGINIA

and

DAN ARMSTRONG
TRW ENERGY ENGINEERING DIVISION

FEBRUARY 1982

PREPARED FOR
UNITED STATES DEPARTMENT OF ENERGY
MORGANTOWN ENERGY TECHNOLOGY CENTER
MORGANTOWN, WEST VIRGINIA

UNDER CONTRACT NO. DE-AM21-81MC16050

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	vi
1.0 INTRODUCTION	1
2.0 THE DEVELOPMENT OF ADACS	2
2.1 Overview of Data Acquisition Development.	2
2.2 Features and Capabilities of ADACS, Release No. 4.0	3
2.2.1 Operating System	3
2.2.2 ADACS Diagnostic and Maintenance Features.	3
2.2.3 Scanning	4
2.2.4 Project Support	4
2.2.5 I/O Support	5
2.2.6 Online Data Presentation	5
2.2.7 Online Disk Recall	6
2.2.8 Online Data Conversion	6
2.2.9 Tape I/O	6
2.2.10 Input Logging.	6
2.2.11 Sequencers	7
2.2.12 Limit checking	7
2.2.13 Sequential scanners.	7
2.2.14 Gas chromatographs	7
2.2.15 Level detector management.	7
2.2.16 Communications interpreter	7
2.2.17 ADACS multitasking	7
2.2.18 Foreground/background.	7
2.2.19 Crash recovery	7
2.2.20 Temperature monitors	7
2.2.21 User Monitor mechanism	7
2.2.22 Online macro definition.	8
2.2.23 Online Reconfiguration of CRT's.	8
2.3 CURRENT APPLICATIONS.	8

TABLE OF CONTENTS (Cont.)

	<u>Page</u>
3.0 ADACS DESCRIPTION	10
3.1 Design Constraints	10
3.2 ADACS Hardware Features	11
3.3 Software Architectural Features	16
3.3.1 Real Time Operating System	16
3.3.2 Memory Utilization	17
3.3.3 Data Tables and Slot Processing	17
3.3.4 Input Algorithms	24
3.3.5 Data Display	26
3.3.6 Control Methods	34
4.0 ADACS PROGRAM DESCRIPTION	36
4.1 ADACS Control Table Setup	36
4.1.1 System Control Files	36
4.1.2 Slot Processing Control Files	36
4.1.3 Sequencer Control Files	38
4.2 The ADACS Startup Tasks	39
4.3 The Asynchronous Tasks	43
5.0 BIBLIOGRAPHY	56
APPENDIX A - Sequencer Example	A-1
APPENDIX B - Description of ADACS Users	B-1
APPENDIX C - Thermocouple Conversion Tables	C-1
APPENDIX D - Post Processing	D-1
APPENDIX E - Command Summary	E-1
APPENDIX F - Typical ADACS Dialogue	F-1
APPENDIX G - The ADACS Routines	G-1
APPENDIX H - Commons	H-1
APPENDIX I - Magnetic Tape Format	I-1

LIST OF TABLES

		<u>Page</u>
Table 1	List of METC Drawings of the Individual ADACS Block Diagrams	15
Table 2	ADACS Overlay Structures	18
Table 3	Letter Assignments in the Sixteen Bit Reduced Word . . .	23
Table 4	Sequencer Instructions	38
Table 5	The FORTRAN Channels and Their Uses	42
Table 6	A List of the Gases Analyzed for by the B-4 Gas Chromatographs and the Associated Full Scale Percentages	48
Table 7	The Constants Used in the GCOUT Calculations	48
Table 8	Possible Classifications of First Words of the CRT Inputs	50
Table 9	Some of the Overlays Used for Deciphering CRT Input . .	51
Table 10	Some of the Available Post-Processing Calculations . . .	53

LIST OF FIGURES

Figure 1	Block Diagram of the Building B-15 ADACS Hardware . . .	12
Figure 2	The B-15 ADACS	13
Figure 3	A technician using a color graphics terminal	14
Figure 4	Example of a Slot Value Display Format	27
Figure 5	Example of a Graphic Data Display	28
Figure 6	Example of a Flowchart Mode Data Display	29
Figure 7	Example of a Real Time Plot Display	31
Figure 8	Example of a Recall Plot Display	32
Figure 9	Example of a Recall Statistics Display	33
Figure 10	ADACS Arrays	41

SYSTEM DESIGN OF THE METC
AUTOMATIC DATA ACQUISITION AND CONTROL SYSTEM

by

D. R. Goff¹ and D. L. Armstrong²

ABSTRACT

A system of computer programs and hardware was developed by the Instrumentation Branch of the Morgantown Energy Technology Center (METC) to provide data acquisition and control features for research projects at the site. The Automatic Data Acquisition and Control System (ADACS) has the capability of servicing up to eight individual projects simultaneously, providing data acquisition, data feedback, and process control where needed. Several novel software features - including a data table driven program, extensive feedback in real time, free format English commands, and high reliability - were incorporated to provide these functions.

¹Former Project Manager (currently with ITT), Valve Testing and Development Projects, United States Department of Energy, Morgantown Energy Technology Center, Morgantown, West Virginia.

²Member of Technical Staff, TRW, Energy Engineering Division, Morgantown, West Virginia.

1.0 INTRODUCTION

Most research and development projects require some data gathering method. A multi-user computerized system was developed at the Morgantown Energy Technology Center (METC) of the U.S. Department of Energy that provides data acquisition and control functions for its projects. This report documents the system and the unique features and methods that went into developing it. It is felt that similar systems could have many applications in other test and research facilities.

2.0 THE DEVELOPMENT OF ADACS

2.1 OVERVIEW OF DATA ACQUISITION DEVELOPMENT

ADACS is an evolutionary data acquisition system that has grown to match the requirements of the METC research projects that it serves. Early data acquisition systems (ca. 1975-76) consisted of stand-alone hardware systems. These systems were limited to collecting the data and storing it on magnetic tape. They offered no means of real time data feedback, and signal processing was very limited. Additional data, especially concerning signal range changes, had to be collected by the test technicians and kept in log books. After the test run was completed, the tapes would be processed and the data analyzed.

There were several inherent problems with this system. First, there could be communications problems between the person taking the data and the person analyzing it. For example, a signal range would change and it would either not be noted in the log book, or not used during the analysis, thus giving invalid results.

Another problem which happened occasionally was that a critical input signal would cease to operate. The technicians, having no feedback from the system, would not know that the signal was missing and would continue testing, only to find out at the end that there was little or no meaningful data taken.

As the projects became more complex, a second generation of data acquisition systems was purchased (early 1977). These were based on a Data General Nova 840* minicomputer with data acquisition hardware and software by Datum. The software sold with these systems offered a limited amount of signal processing and provided limited feedback to the technicians via CRT monitors. Each project required its own system, which could cost from \$100K to \$250K, depending on its size.

By late 1977, an effort was undertaken to write new software for these computer systems to allow them to provide data acquisition for multiple projects. This was desirable both from an economic and a manpower standpoint. Economically, it would be wasteful to purchase a system for each project on station, especially since projects were generally of limited duration. From the manpower standpoint, individual systems would require too much in the way of maintenance and operation.

*Manufacturers' names on products described herein are given only for technical completeness and do not constitute endorsement by the U.S. Government, its agencies, employees, or contractors.

The new software, which ultimately grew into ADACS, initially provided the same support as the Datum systems, but for multiple users. As the system developed and the needs of the projects expanded, it was seen that new features could be added to the system to enhance its capabilities. Thus, the data processing capabilities were expanded to include complex calculations, the data feedback features were expanded to allow various display modes including graphic displays and plots, and sequential and continuous control features were added. Some of the more important features and capabilities of the present ADACS, release number 4.0, are outlined below.

2.2 FEATURES AND CAPABILITIES OF ADACS, RELEASE NO. 4.0

2.2.1 Operating System. ADACS runs under the Real Time Disk Operating System (RDOS) by Data General. Some of the RDOS inherent features include:

1. Multitasking with priority for up to 256 tasks. This allows the computer to do a large number of jobs simultaneously. ADACS uses about 30 tasks. The priority structure allows the more important tasks, such as data acquisition, to run at their own speed while the lesser tasks, such as data feedback, are run when the computer has time.
2. Overlays. This is a memory conservation method which allows routines which are not used too often to be switched in and out of memory, leaving only the most used code as core resident. About 50 overlays are used in ADACS.
3. Spooling. This allows the computer to run at full speed without waiting on slow I/O devices.
4. ERCC, which corrects all single bit memory errors.
5. Memory protection. Memory blocks of 1024 words (1K) are protected from interference from other grounds (e.g., foreground/background).
6. Data channel protection. Memory blocks of 1K are protected.

2.2.2 ADACS Diagnostic and Maintenance Features. ADACS employs several schemes to ensure that the system hardware and software are functioning properly. They are:

1. A Watchdog timer. This feature performs several functions:
 - A. It reinitializes the software and Datum hardware when abnormal conditions are detected. This allows automatic recovery from minor software bugs and hardware failures.

B. It outputs a pulse to an external hardware timer every five to 10 seconds when conditions are normal. If the timer fails to receive a pulse, it sounds an alarm to indicate a possible computer failure.

2. A Floating Point Unit check. This feature checks nearly all of the single precision instructions on the floating point arithmetic unit once each hour.

3. An analog input channel calibration check. It sounds an alarm if the offset or gain of any multiplexer box exceeds one percent of ideal. It can detect the impending failure of a single analog input channel. This check is performed 3 times each hour.

4. ERCC logging. This feature keeps a record of all single-bit memory failures. This allows failing memory boards to be pinpointed before a multiple bit failure occurs.

2.2.3 Scanning. ADACS can scan up to 2048 high accuracy analog input channels. Each has twelve gain ranges available, ranging from 4mV to 10.24 V full scale. The analog-to-digital converter (ADC) has an autorange feature which automatically selects the most accurate range. After the gain range is set, the analog inputs are converted to digital numbers by a 12-bit ADC. The channels are scanned at a 15kHz rate, giving nearly simultaneous readings between adjacent slots (analog input channels). The data are written on magnetic tape as fast as once per second.

ADACS can handle up to 64 analog outputs. Each digital-to-analog converter (DAC) provides a zero to 10 volt signal with 12 bit resolution. These outputs are updated 1 to 10 times each second.

To monitor switch inputs and to control relays, up to 384 discrete input and/or output combinations can be used. These input and output channels can be updated asynchronously.

2.2.4 Project Support. Up to eight independent research projects may be supported simultaneously. A total of 1,024 slots may be scanned for all users, with a single user maximum of 512. These are software limitations.

2.2.5 I/O Support. A number of I/O devices are supported by ADACS:

1. One graphic CRT operating at any baud rate serving as the master control CRT.
2. One background CRT or teletype operating at any baud rate.
3. Eight graphic CRTs operating at 4800 baud for the users. These may be reconfigured online to allow several CRTs to be assigned to a single project. Each crt has hard copy capability.
4. One lineprinter for hard copy backups of data, messages, etc.
5. Two tape units. ADACS will switch to the backup tape drive in the event of a tape failure or end-of-tape.
6. Two disk units. These are used to store the ADACS programs, control files, and online data records.

2.2.6 Online Data Presentation. ADACS has the capability for several modes of data feedback:

1. Flowchart mode. This provides a predefined tabular listing of slots. Up to 26 flowchart configurations may be defined per user. Updating is automatic and may be scheduled as often as once every 10 seconds.
2. Real time plotting. Up to eight variables (slots) may be plotted on the CRT screen, with updates as often as once per second. Each plot may hold 120 values, and the user has the option of producing automatic hard copy when the plot is filled.
3. Project graphic. Each project may have up to 26 predefined, detailed project graphic displays showing major vessels, piping, and slot values. Graphics are updated by demand only.
4. Recall plots. Any slot (up to 5 simultaneously) may be recalled from disk and plotted on a high precision line plot, with up to 720 points on a plot.
5. Recall lists. A list of values for a given time period can be recalled for up to five slots. The minimum, maximum, average, and standard deviation are computed and listed.

2.2.7 Online Disk Recall. Currently, each project can store up to 65,536 scans on disk, with the scan interval schedulable, as fast as once per second.

2.2.8 Online Data Conversion.

- 1) Thermocouples -- Table look up for type K and T thermocouples accurate to $\pm 1^\circ\text{F}$.
- 2) Sensors -- Any linear conversion function. Fifty linear functions are typically available for each ADACS system.
- 3) Meloy analyzer -- Conversion to PPM of SO_2 using a highly nonlinear relationship.
- 4) Linearization of analyzer signals using up to 3rd order polynomials.
- 5) Flows -- Correction and linearization of flows. The computer fully corrects for pressure and temperature variations for:
 - A) Liquids
 - B) Gases
 - C) Saturated Steam
 - D) Superheated Steam
 - E) Rotameter Flows
- 6) BCD digit assembly from analog channels. Allows signals to be acquired that require more accuracy than can be obtained from a single analog channel.
- 7) Strain gage corrections. Allows individual calibration for each strain gage channel plus automatic compensation for offset drifts.
- 8) General math function can be applied to a slot. Allows up to five slots to be manipulated in a general way.
- 9) A slot can be differentiated to yield rate.
- 10) A slot can be integrated to totalize.

2.2.9 Tape I/O. ADACS allows a single tape unit to be shared among eight users. Data records are explicitly identified as to their content and user. The time and date are recorded in all tape records.

2.2.10 Input Logging. ADACS allows operator entered comments and data to be stored on tape and disk to ease later interpretation of the data.

- 2.2.11 Sequencers. Up to eight sequencer control tasks are allowed. Programs of up to 1000 instructions each can be handled. They are used to automate several METC projects (Appendix A).
- 2.2.12 Limit checking. Validity of data can be checked against two predefined sets of alarms and warning messages generated.
- 2.2.13 Sequential scanners. ADACS can accept input from up to four sequential scanners of up to 100 channels each. The maximum scan rate is one step per second. This feature was incorporated to service pneumatic scanners.
- 2.2.14 Gas chromatographs. ADACS can handle chromatographs with serial component output. Results are output to tape, LPT, memory, and a second TTY. ADACS also computes Btu, specific gravity, molecular weight, etc.
- 2.2.15 Level detector management. ADACS incorporates a task that monitors and reduces data from the producer nuclear level detectors. ADACS estimates the bed mass and looks for voids.
- 2.2.16 Communications interpreter. ADACS has built into it a sophisticated, highly overlayed communications interpreter that accepts input commands from all graphic CRT's and acts on the request that was made. Data flow, manipulation, interpretation, and processing can all be modified and observed on line.
- 2.2.17 ADACS multitasking. Allows new features to be easily added to ADACS without interfering with existing features.
- 2.2.18 Foreground/background. ADACS is written to run in the foreground, allowing RDOS to run in the background to be used to develop programs and analyze data. When in this mode, ADACS will output LPT data to disk so that the LPT is available for use by background.
- 2.2.19 Crash recovery. ADACS allows a data tape to be repositioned after a shutdown or crash. This allows maximum tape utilization.
- 2.2.20 Temperature monitors. Ten internal temperature monitors are allowed - one for each of eight MUX boxes, one for the CPU, and one for UDAM. The monitors typically sound an alarm if any value exceeds 110°F. If an alarm triggers, the threshold is raised 5°F.
- 2.2.21 User Monitor mechanism. ADACS allows the master CRT to monitor or recall operator input on any or all of the user CRT's.

2.2.22 Online macro definition. ADACS allows abbreviated forms of any command to be defined online. Up to 26 shortened commands may be defined by each user.

2.2.23 Online Reconfiguration of CRT's. ADACS allows several CRT's to be assigned to a given project. This allows large projects such as the Gas Producer to use two or three CRT's simultaneously.

2.3 CURRENT APPLICATIONS

ADACS has been applied to a broad range of projects at METC since its implementation. The system was developed so that a single program could be easily adapted to a large variety of applications. There are currently four data acquisition systems in service, each running essentially the same program. The most complex of these is located in building B-4 Annex and serves the 42-inch Gas Producer, the Holmes-Stretford unit, erosion/corrosion testing, the slurry drying system, and two 18-inch fluidized beds.

The gas producer and cleanup system are the major users of the B-4 Annex system, currently feeding about 350 signals into the system. Some complex control signals are returned to the project. These deal primarily with the humidifier nozzle flows. Also, many abstract calculations dealing with previously unmeasurable quantities which are inferred from a number of other signals, such as the producer bed mass, gas humidity, mass and energy balances, etc. are performed.

The Holmes-Stretford gas cleanup system, which is used to remove sulfur compounds from the producer gas, sends about 70 signals to the B-4 Annex System.

The erosion/corrosion project, which is concerned with determining the effects of producer gas on turbine components, uses about 90 signals. This project also requires complex calculations of inferred variables, such as gas velocities.

The slurry drying system is currently inactive but still has about 60 signals associated with it.

There are also two eighteen-inch fluidized beds monitored by the B-4 Annex System, each presenting about 80 signals. In addition there are a number of calculations dealing with inferred quantities.

The second-largest system is located in building B-15. This system is devoted entirely to the four test units of the valve testing and development programs - the Valve Static Test Unit, the Valve Dynamic Test Unit, the Valve Hot Solids Test Unit, and the Valve Slurry Test Unit. Each of these units presents from 50 to 100 signals to ADACS.

These test units are about ninety-five percent automated by the B-15 system through the use of ADACS's sequential control features. Once a valve has been placed in the test train and the proper parameters set up in the computer program, the computer will take over the testing procedure. Actions such as valve position changes and pressure setpoint changes are initiated and the proper response of the test units verified. If an expected result does not occur in a specified time interval the system returns to a safe state and notifies the technician of the problems. With no test unit or test valve failures, testing can continue for days without technician intervention.

The next system is located in building B-1. This system monitors a small research laboratory which is limited in the number of signals that are processed, about twenty-five. However, this system acquires data at a fairly high rate and some complicated calculations are performed.

A fourth system is currently being installed to serve the advanced gasification concepts laboratory. It will initially monitor an entrained bed gasifier and a fluidized bed gasifier.

Additional information on these projects can be found in Appendix B.

3.0 ADACS DESCRIPTION

3.1 DESIGN CONSTRAINTS

A number of constraints were imposed upon ADACS in the early stages of its development. The way that these constraints are met is described under the heading Software Architectural Features.

First and foremost among the constraints is that the system had to be multitasking. This would allow the system to handle all of the programs (I/O, conversions, etc.) for several projects with one computer, running them seemingly simultaneously.

The system had to be able to convert the data from a raw signal to meaningful engineering units prior to recording both the raw signal and the converted signal on magnetic tape. In the earlier systems, the data was simply written on tape as a voltage signal. Then, after the testing was completed, the tape was processed to convert the signals into the appropriate engineering units. This often led to communications problems. For example, when a signal range changed, the person processing the tapes might not pick it up, thus converting the data incorrectly. Also, it was very difficult to look at old tapes and determine the status of each signal at that time.

Another system requirement was that it had to have the capability to display the converted data to the technician in real time. The technicians could then verify that the data was complete and correct. This would give them an indication of how their process was performing and would allow faulty instrumentation or other problems to be spotted before significant amounts of data were lost.

The next constraint was that the system had to have the capability for controlling processes. This would be implemented in two ways. The first is sequential control, which would allow the system to perform a repetitive series of operations involving switching such as opening/closing valves, turning motors on/off, etc. The second, continuous control, would allow the system to control a temperature, pressure, or other variable by means of a complex algorithm based on some independent variable(s).

Finally, the system should be able to perform complex calculations using the data that it acquires. This would free the technicians or engineers from making calculations and allow them to look at variables that cannot be measured directly, such as flow.

3.2 ADACS HARDWARE FEATURES

To convey an idea of the ADACS architectural layout, the reader is referred to Figure 1, which shows the general configuration of a system. A list of METC drawings showing the specific layouts of each of the systems is given in Table 1, and some of the equipment used is shown in Figures 2 and 3. There are some hardware differences between the four systems, but the basic configuration is very similar.

All of the current data acquisition systems are based on Data General minicomputers. Two of the systems use Eclipse S-200 minicomputers each with 128K* words of memory. Another system uses a Nova 840 computer with 64K words of memory, and the fourth uses a Nova 3.

Each of the systems uses a 10 megabyte moving head disk for program and intermediate data value storage. The B-4 Annex system also uses a Zebra moving head disk which provides 192 megabytes of on-line storage. This disk is used primarily to store one to three weeks of data for quick recall by the technicians. This feature allows current conditions to be easily compared to past time intervals.

Each system has either one or two nine-track tape drives, each of which can store 800 bits per inch at a rate of 37 1/2 inches per second. Only one tape is active at a time; the second tape is switched to automatically if the first tape becomes filled or if a problem is encountered with it.

Graphic CRT terminals are used to provide for user input/output (I/O). Typically a system will have a model 4014 Tektronix CRT serving as the master or controlling CRT which is used to control ADACS and oversee all of the projects associated with the system. A model 4006 Tektronix CRT will be dedicated to each project for data retrieval. The Tektronix terminals provide black and white text and high resolution graphics output.

Analog I/O is provided via equipment manufactured by Datum which was designed to operate with Data General computers. The analog to digital converter (ADC) allows up to 2048 input signals which are converted to twelve bit digital values. Ten different gain ranges, ranging from minus 10 to plus 10 volts down to minus 4 to plus 4 millivolts, are available, with the most accurate range being chosen automatically by the ADC. The data is sent to the computer as a 16-bit word, with the extra four bits conveying the range information. The ADC is a direct memory access device (DMA) which allows scan rates up to about 15,000 channels per second.

*1K is equivalent to 1024 words

Figure 1. Block Diagram of the building B-15 ADACS Hardware.

Figure 2. The B-15 ADACS.

Figure 3. A technician using a color graphics terminal.

Table 1. List of METC Drawings of the Individual ADACS Block Diagrams

Drawing #	ADACS
D 800312	B-4
D 800313	B-15
D 800314	B-1
D 800315	Instrumentation Branch

The Datum hardware provides up to 64 digital to analog converter (DAC) outputs which accept a twelve bit digital input and convert it to an analog signal ranging from 0 to plus 10 volts. These outputs are necessary for the control features which are incorporated into ADACS.

The hardware provides for up to 384 discrete digital inputs which are used to sense the on or off status of a valve, motor, etc. and up to 384 discrete digital outputs, which allow the computer to change the on or off status of a valve, motor, etc.

Each system can support an ISC color monitor for system graphic displays. These color displays are normally used to give an overview of the project as a graphic display. By changing the display colors the status of valves, out-of-range slots, alarms, etc., may be shown.

3.3 SOFTWARE ARCHITECTURAL FEATURES

A number of constraints were placed on the operating system in the early stages of ADACS development. To allow the system to respond to data that appears at an uncontrollable moment in time the operating system had to be a 'real time' design. Multitasking was necessary for the computer to perform a large number of tasks simultaneously. Since some of the tasks are more critical than others or must run more frequently than others, the operating system had to allow a priority structure. Finally, since the program was likely to become very large, an overlay structure was deemed necessary.

3.3.1 Real Time Operating System

To fulfill all of the above requirements, the Real time Disk Operating System (RDOS), a standard Data General product, was chosen. RDOS is multitasking, allowing up to 256 tasks to run simultaneously, with up to 256 different priorities, if needed. Overlays are allowed, and the operating system uses spooling to let the computer run at full speed without waiting for slow I/O devices. Additional RDOS inherent features include ERCC (corrects all single bit memory errors), memory protection (protects 1K memory blocks), and data channel protection (protects 1K memory blocks).

3.3.2 Memory Utilization

As the available features increased the memory requirements soon exceeded the available storage. This situation was handled by grouping subroutines into overlays. About fifty overlays are used in ADACS, as shown in Table 2. Overlay area two is initially used by the ADACS master task MAST. Once MAST has run it is replaced by the other routines listed under area two. These routines remain there as long as ADACS is operating. In the other overlay areas the routines are swapped in and out as they are needed.

3.3.3 Data Tables and Slot Processing

As ADACS developed, the need for program flexibility was noted. Due to the large number of projects that would use the system and the limited number of personnel that could be dedicated to software support, it was determined that the program must be versatile enough to support virtually any project with only minor alterations. This would also allow the system to be upgraded at one location and the new version easily distributed to the other locations.

One way that flexibility was achieved was to write the code in a general fashion, making as few assumptions about the project as possible. This meant that assumptions made concerning the number of signals a project would provide, how they would be presented to the computer, the type of processing required on the data, how the data would be presented back to the user, and so on were kept to a minimum. Over 95% of the routines in ADACS are extremely general and are used universally in all of the existing systems. Inevitably, each system would be sufficiently different from the other systems that it wasn't practical to include that respective code in all of the systems. Therefore, about five percent of the code is site specific.

The major method of achieving flexibility was by making the data processing routines - those that perform the conversion of data to engineering units, perform simple calculations, present data back to the technicians, etc. - data table driven. This means that a table of instructions would be set up using a separate set of programs telling the routine exactly what to do in that specific instance. This allows the routine to adapt to the requirements of the individual projects.

Table 2. ADACS Overlay Structures.

<u>Routines</u>	<u>Routines</u>	<u>Routines</u>
OVERLAY <u>0</u>	OVERLAY <u>1</u>	OVERLAY <u>2</u>
MTEH1	all communications	MAST
MTEH2	(COMXX) subroutines	PLOT
MTEH3	GRAPH	LIMCHK
INPUT	LCAL1	POST4
CL1	GCOUT	REPORT
CL2	SCAL1	ZDATA
SMSG	SCAL4	VECTOR
ENDIT		
STFIL		
COM26		
B4MES		
GETSYS		
GETTOK		
FINDIT		
DPL0T		
METRIC		
REDISK		
POP		
GETSL0T		
SMSG2		

Due to the data table driven nature of the routines, a new project can be set up in a matter of days by creating a new set of data tables. The processing of a given signal can be changed in a matter of minutes.

There are four major ways that data tables are used. They are:

1. slot processing
2. slot processing data table size reduction
3. sequential controls
4. CRT displays

The data collected by ADACS or calculated within ADACS is organized into slots. The slot number provides a way to retrieve data stored or calculated by ADACS.

Slots can be classified according to the data processing they require:

1. No Processing Required -- Raw millivolt signal recorded.
2. Linearly Transformed Slots -- Simple slope and intercept conversion.
3. Thermocouples -- Requires evaluation of a piecewise linear function.
4. Switch Function -- Slot compared against a threshold.
5. Post Processed Slots -- Complex calculations of a general nature, e.g., flow calculations.
6. Special Calculations -- Hard coded calculations that are project specific.
7. Dedicated Tasks -- Complex calculations that require real-time response considerations.

Most slot calculations are data table driven (types 1-5), that is, a data table from each slot is kept in memory which determines the type of calculation which is done and contains all parameters needed.

Linear slots are those whose final value is a linear conversion from the transmitter input and only a slope and intercept are needed to obtain the result. Thermocouple slots are very similar to linear slots except that they cannot be expressed as a single slope and intercept. Their working functions consist of a number of linear segments with slopes and intercepts which approximate the actual function of the thermocouple. Switches also are similar to linear slots, but all that is needed is an indication whether the transmitter signal is above or below a specified threshold level.

"Post processing" slots require considerably more processing than a simple conversion from transmitter signal to final value. Post processing permits complex calculations to be done which may require data from several different transmitters. These calculations are used by most projects and consist of such types of calculations as flows, BCD digit conversion, polynomial calculations, integration, etc.

Special slots permit the use of calculations which are of interest to only a single project and where each calculation may pertain to only one or two slots. These calculations are not data driven as are other types, but are hard coded into a separate special program.

In order for a project to use a slot, that slot's definition must be entered in the project Acquisition Control File (ACF). These slot definitions are created, deleted, and modified in the program EDACQ and entered into the project ACF in the program CRACQ (both described later).

Slot definitions consist of a number of pieces of information. Including a slot number (0-5119), this information consists of:

- eight-character identifier

- 32-character description

- four-character engineering unit

- function indicator -- indicates type of processing to be done

- limit indicator -- indicates normal range of data

- range indicator -- indicates, for those slots associated with transmitters, the MV range of the transmitter

In order to minimize the amount of memory required to store the processing requirements of a given slot, a data compression scheme is used. Since engineering units, functions, limits, and ranges typically only take on a limited number of distinct values for a given system, these entities are defined in tables and then referred to by entry number in the table. For instance only 32 different engineering units may exist per ADACS. Similarly linear functions are limited to 50, limits may number 64 and up to 16 range values may exist per ADACS. These restrictions have proven to be adequate thus far on all ADACS projects.

All of the information about a slot is memory resident at execution time except the 32 character description, which is loaded from a disk file only when needed. Nine 16 bits words serve to define each slot. The layout is:

WORD 1	=	"IN"	=	Slot number (0-5119)	
WORD 2	=	"NF"	=	Bits 11-15 → engineering units pointer (0-31)	
				Bits 0-7 → function pointer (-5 to 110)	
WORD 3	=	"LIM"	=	Bits 4-7 → range pointer (0-15)	
				Bits 8-15 → limit pointer (0-63)	
WORDS 4-7	=	"INDEN"	=	8 character identifier	
WORD 8	=	"CRTX"	=	X coordinate for CRT display	} used for real time display updating
WORD 9	=	"CRTY"	=	Y coordinate for CRT display	

If a slot requires "post processing", an additional 20-word table is set up for the slot. The layout of this table is discussed later. Also if the slot value is to be output to a DAC, a 6-word driver table is created for the slot.

In order to determine the type of processing that a slot is to receive, it will look up the second word of that slot's control file. This tells ADACS which of the pre-stored engineering units to associate with the slot and what type of processing is needed by the signal. The processing may consist of one of a number of linear conversions, one of two nonlinear conversions, or post processing. Then, depending on the processing required, ADACS will go to additional data tables to obtain the necessary information to do the processing. For example, if a linear conversion is specified, ADACS will use the function pointer from the slot description to look up the proper slope and intercept from an array in memory. It will then do the calculation.

There are several types of slot processing available.

Linear Transmitters. Linear transmitters return voltage signals that are very simple to convert to engineering units, requiring a single slope and intercept. When defined in EDACQ, these signals are given slot numbers between 0 and 2047. Required are an 8-character identifier, a 32-character descriptor, an engineering units pointer, and a positive function pointer in the range 0 to 50 which determines the slope and intercept of the function. Usually applied, but optional, are the limits pointer and range pointer. Function 0 is a special case that does not modify the slot value, e.g., SLOPE = 1.0, INTERCEPT = 0.0.

Thermocouples (Nonlinear Transmitters). ADACS is able to translate the output from thermocouples (or other nonlinear transmitters) in much the same way that linear transmitter values are converted.

A lookup table is used to locate a slope and intercept for a small section of the nonlinear function which corresponds to the value sent by the thermocouple. This slope and intercept is then used to convert the value to temperature.

The lookup tables for different thermocouples consist of slopes, intercepts, and boundaries for a number of segments (≤ 15) which divides the voltage to temperature curve into linear segments.

When defined in EDACQ, along with an 8-character identifier and a 32-character descriptor, thermocouple slots are given negative function pointers (-1 to -5) which define the particular lookup table to use. These slots are usually given limit pointers but not range pointers.

Currently, only two thermocouple functions are defined in ADACS. They are:

1. COPPER-CONSTANTAN
2. CHROMEL-ALUMEL

The conversion tables (listed in Appendix C) for the thermocouples are derived through an iterative curve fitting procedure. First, one decides what range of temperature should be covered. A data file is then entered into the computer with entries spaced at about 20°F intervals. Then a curve fitting routine is executed to do a linear fit to the data values. The object is to include as many data values as possible in each segment while retaining a $\pm 0.5^\circ\text{F}$ conversion accuracy. For copper-constantan and chromel-alumel, 10 linear segments are needed to span the full temperature range of the thermocouples. The conversion tables also take into account the 150°F reference ovens used on ADACS.

Switches. Switches are defined where it is desired to know only when a voltage is above or below a specified threshold point. They are defined in EDACQ as any other analog channel slot except the function pointer is given as a value from 100 to 110 where the amount over 100 is the threshold for the switch in volts.

Post Processing. Post processing allows the use of a set of predefined general purpose algorithms which are of use to several projects. These algorithms permit the calculation of values which are not a simple conversion to engineering

units from the millivolt value of a single transmitter. Often, these calculations require the combining of values from more than one transmitter.

The use of these algorithms requires that certain parameters be defined. These parameters are different for each algorithm, but are defined and stored similarly in arrays based upon a master array called CPOST. This CPOST array consists of a number of words into which all the parameters needed for a specific algorithm are placed in a specified manner. These parameters and their placement in the CPOST array are described later.

The definitions for post-processed slot parameters are placed in a file named POSTX.YY (X = User No., YY = Building No.). The definitions for each slot are placed in the file in numerical order and consist of several lines giving the parameter definitions. The first line of a definition consists of the type of processing to be done (i.e., "FLOW"), the slot number the value is to be placed into, the 8-character ID for the slot, and an optional comment. The following lines consist of a number determining the data type of the third field (0 = integer, 1 = real, 2 = slot number, and 3 = octal number in first position of next line), the word in CPOST array for the storage of the data (one-half the first location for a real number, which requires two words for storage), the data value (integer, real, slot number, or a dummy when an octal number is called for), and an optional comment. The indexing into the CPOST array does not need to be in order and any word which is not used in a CPOST layout may be left out of the definition. Each definition ends with a line of three minus ones. The POSTX.YY file is usually created with the RDOS Text Editor.

When defined in EDACQ, post-processed values are given slot numbers in the range 3072 to 4095. Required are the 8-character identifier, 32-character description, an engineering pointer, and normally a limit pointer. A function pointer and range pointer are not used and are set to 0.

NOTE: Not all of the post processing functions are available on all ADACS systems. Rather only the required functions are loaded in to conserve memory. In order to find out exactly what complement of functions exists for a given system, examine file "DRG:POSTX" where X is the ADACS site number, i.e., B4 is "DRG:POST4", B15 is "DRG:POST15"). If a post processing function is specified that is not implemented, a message, "POSTP ERROR ZZZZ" will appear on the system console, where "ZZZZ" is the post processing function, e.g., ZZZZ = FLOW.

The layouts for the various types of post processing algorithms are given in Appendix D.

3.3.4 Input Algorithms

The next requirement imposed on ADACS during its development was in terms of human factors design - the program had to be very easy to interface (communicate) with. Most of the personnel that would use the system were METC engineering technicians, most of whom had only limited typing experience and generally had no prior experience with computers. In order to make the system conveniently usable, a number of things had to be done to tolerate input errors.

First of all, to make the computer easier to use for those with no experience, the commands were made to be verbalizations of what the technician would want the computer to do. If the technician wanted the computer to start scanning on some project the command would be "START SCANNING", or if the technician wanted to look at the value of a given slot he would simply type in "DISPLAY SLOT" and then the slot number. A comprehensive listing of commands is presented in Appendix E. A sample of ADACS dialog is given in Appendix F.

Next, to compensate for typical typographic errors, a hashing algorithm was devised to take the user's command and reduce it according to a series of rules which would allow the computer to interpret the command entered. The algorithm was designed to allow the system to tolerate some of the most common typing errors.

Any degree of letter transpositions (i.e., S-T-R-A-T instead of S-T-A-R-T) can be tolerated, including having more than one letter transposed. Also, the system will ignore any duplication of letters where the duplication was not required (i.e., S-T-T-A-R-T) or tolerate non-duplication where duplication was required (i.e., S-T-A-R).

The hashing algorithm works by taking each word (token) within the command and reducing it to sixteen bits, one computer word of information. Each bit of the sixteen bits is associated with one or more letters and if that particular letter is present in the token being analyzed that bit is set to one; otherwise it is set to zero.

The letters that are associated with each bit in the sixteen-bit word are shown in Table 3. The order in which this was chosen corresponds basically with the frequency of use of the letters. The letter E is generally the most frequently used letter, followed by T and the others. The last four bits are associated with infrequently used letters.

Table 3 - Letter assignments in the sixteen bit reduced word.

Bit	Letter	Bit	Letter	Bit	Letter	Bit	Letter
0	E	4	R	8	H	12	F,P,U,M
1	T	5	S	9	L	13	V,W,G
2	A	6	N	10	D	14	J,K,X,Y
3	O	7	I	11	C	15	D,Q,Z

By studying the way the command word is reduced one can easily determine how the algorithm eliminates mistakes due to letter transpositions, duplications, or non-duplications. If a command has two T's in it, the first T sets bit one and any following T's have no effect. Also, it doesn't matter where the T appears in the word as long as it is there, since the bit will still be set to one. One constraint on a system like this is that the frequency tables are based on a wide range of English literature. In some cases, a letter which is infrequently used in general may be used extensively in a particular application. For example, if graphical work is being performed, it may be necessary to distinguish X, Y, and Z. To resolve command words that generate the same code a second 16-bit word is generated that is simply the sum of the ASCII codes associated with each letter in the command. This result is used only to resolve ambiguities.

Also, to make program interfacing easier for the technicians the commands were designed to be input in a free format. There is no requirement on the number of spaces between words of a command - there can be one or ten or as many as the technician desires.

3.3.5 Data Display

Another constraint placed upon ADACS was that it must be able to present data back to the technicians in real time. This was necessary to allow for the detection of any problems with the experiment as it was being performed and to give the technicians better control of the testing.

ADACS has several means of accomplishing real time data feedback. The first of these is by simply displaying the current value of a given slot. An example of the display format is presented as Figure 4.

Next are project graphic displays showing pictures of process piping, major vessels, sensors, and current slot values. Each project may have up to 26 predefined project graphics. Typically, the first of these is an overview of the entire process showing only the most critical piping, vessels, and data values, while the remainder are used to detail each individual vessel. Each update of a graphic display must be requested by the technician. An example of a typical graphic display is presented as Figure 5.

In the flowchart mode, tabular listings of slot values are displayed. Up to 26 predefined preformatted lists of slots may be used per project. Slot value updating is automatic and may be scheduled as often as every ten seconds. An example of a tabular listing is presented as Figure 6.

The project graphic and the flowchart display formats are set up in data tables prior to run time. When a particular display is needed ADACS gets the proper control table from disk storage. This table tells ADACS where to place lines and slot values on the CRT to create the display.


```

DISPLAY SLOT 48 10
SLOT 48 " 991.749 MU (USER 1)
SLOT 49 " 983.634 MU (USER 1)
SLOT 50 " 1.960 PSIG (USER 1)
SLOT 51 " 29.501 MU (USER 1)
SLOT 52 " 1014.429 MU (USER 1)
SLOT 53 " -37.331 PSIG (USER 1)
SLOT 54 " 30.613 MU (USER 1)
SLOT 55 " 969.902 MU (USER 1)
SLOT 56 " 1.804 PSIG (USER 1)
SLOT 57 " 8.049 MU (USER 1)
OK

READY
D S 4S 10
SLOT 48 " 990.501 MU (USER 1)
SLOT 49 " 976.144 MU (USER 1)
SLOT 50 " 1.335 PSIG (USER 1)
SLOT 51 " 29.520 MU (USER 1)
SLOT 52 " 1014.429 MU (USER 1)
SLOT 53 " -37.175 PSIG (USER 1)
SLOT 54 " 30.930 MU (USER 1)
SLOT 55 " 981.972 MU (USER 1)
SLOT 56 " 1.960 PSIG (USER 1)
SLOT 57 " 8.010 MU (USER 1)
OK

READY

```

Figure 4. Example of a slot value display format.

Figure 5. Example of a Graphic Data Display.

TIME = 13:52: 6 OVERLAYS 0

570 TIR-1	68.84 DEGF	474 UIH-2	-2.72 LBS
571 TIR-2	66.35 DEGF	475 UIH-1	-3.04 LBS
572 TIR-3	66.91 DEGF	484 FI-1A	-.02 RU
573 TIR-4	67.14 DEGF		
574 TIR-5	67.48 DEGF	476 A.CO.1	16.33 XFS
575 TIR-6	68.04 DEGF	477 A.CO2.1	8.58 XFS
473 PD-2	-25.00 °H2O	3084 R.COAL	8.78 \$/HR
577 TIR-8	65.22 DEGF	3085 R.LINE	7.83 \$/HR
578 TIR-9	65.33 DEGF		
579 TIR-10	66.24 DEGF	482 A.CH4.1	9.60 PPM
580 TIR-11	67.93 DEGF		
581 TIR-12	61.60 DEGF	3073 A.CO2.1	1.25 %
582 TIR-13	63.30 DEGF	3075 FI-42x	.00 SCFH
583 TIR-14	74.15 DEGF	481 A.S02.1	-249.96 PPM
584 TIR-15	76.53 DEGF	3072 A.CO.12	.11 %
585 TIR-16	76.86 DEGF		
586 TIR-17	75.06 DEGF	3077 FI-13x	.00 SCFH
587 TIR-18	75.85 DEGF	3076 FI-1Ax	.00 SCFH
588 TIR-19	62.05 DEGF	3081 FI-8x	.00 \$/HR
589 TIR-20	60.69 DEGF	3080 FI-6x	.00 SCFH
472 PD-1	-25.00 °H2O	3078 FI-4x	.00 GPM
591 TIR-22	74.49 DEGF	3082 FI.TOTAL	.00 SCFH
494 UI-40	-.51 LBS	3079 FI-5x	.00 GPM
495 UI-39	-4.23 LBS	3083 SU-1	.00 FT/S
492 PDI-37	-2.50 °H2O		
493 PDI-38	-12.50 °H2O		

Figure 6. Example of a Flowchart Mode data display.

The last means of real time data feedback is by plotting each data value on a background grid as it occurs. Up to eight slots may be plotted simultaneously, and the slot values may be updated as often as once per second. Each plot can hold 120 values. The technician can have automatic hard copy of each plot when it is full if it is desired. An example of a real time plot is presented as Figure 7.

As an example of how the real time plot might be used, on the producer the technicians might set up a plot of the crossover temperature with an update every few seconds. This would allow them to see what type of trend the temperature has been establishing over the last few minutes.

The next major feature of ADACS is the data recall ability. This group of features retrieve data from the moving head disks and present it to the technicians in various formats to show how the process has run in the past. The technician establishes the time window that he wishes to examine by defining the beginning time and the width of the window. The ADACS will then search through the disk file, find the data associated with that time interval, and present it in the required manner.

There are two formats available for recall displays. The first of these is a plot. This is a high resolution line plot of up to five slots. Up to 720 values can appear on a plot. An example of a recall plot is presented as Figure 8.

Also, statistical summaries for up to five slots can be recalled. This shows the minimum, average, maximum, and standard deviation for the time frame selected. An example of the statistical recall is presented as Figure 9.

The recall feature has a number of important uses. Anyone associated with data analysis can obtain large amounts of data while the testing is still in progress. Typically, a person will go to the system every few days throughout the run and obtain eight or 24-hour plots and statistical summaries of the slots they consider important. At the end of the run, they have a stack of plots covering the entire run. Before the recall feature was available, it was necessary to wait days or weeks until the run was completed and the data was processed off-line in order to obtain such data.

Figure 7. Example of a Real Time Plot Display.

32

TIME	SLOT 48	SLOT 49	SLOT 50	SLOT 51	SLOT
13:47: 2	1015.74	1104.38	1.97	29.68	
MINIMUM	1004.650	1042.101	1.187	29.069	
AVERAGE	1016.991	1084.383	1.884	29.569	
MAXIMUM	1028.369	1114.507	2.610	30.103	
STANDEV	5.194	16.004	.360	.204	
OK					

READY

READY
RECALL USER 1 STAT 48 49 50 51

Figure 9. Example of a Recall Statistics Display.

Also, it provides a very good diagnostic capability for determining the cause of any unexpected upset to the process. For example, there have been several instances on the producer that have created potentially hazardous situations or have caused the 42-inch gas producer to go down suddenly and no one knew exactly what had happened. By using the recall feature, they could look at a time interval around the time the crash occurred and generally determine what went wrong. In several cases, it was found that valves that should have been closed were left open; had ADACS not been available, it would not have been possible to pin down exactly where the problems had occurred or at least not in a manner which allowed a timely restart.

The most recent addition to ADACS is the capability for color CRT displays. Using a color CRT a graphic image similar to the black and white project graphic is displayed on the screen depicting major vessels, valves, piping, and slots. The update interval for the data values is typically ten to thirty seconds.

Using a color display allows more information to be presented to the technicians with same amount of display area. For instance, a data value that is outside the alarm limits can be shown with blinking red characters. The status of a valve can be shown by drawing its body in green if it is open and red if it is closed.

3.3.6 Control Methods

As ADACS developed, it became desirable to use the computer to control the various projects. This was implemented in two ways - continuous control and sequential control.

Continuous control involves calculating a value and outputting it to a digital-to-analog converter. This allows a process to be controlled by complex algorithms which are not easily obtainable with normal analog controllers. The algorithms are also easily and quickly changed. For example, the humidifier on

the gas producer cleanup system can be controlled with this feature. The water flow to each of the 4 nozzles must be maintained above 50% of the rated flow or it must be shut off altogether. The control algorithm used selects the number of nozzles and the flow rate to each so that the overall flow rate is achieved within the limits of the individual nozzles.

Sequential control is achieved through the sequencer, a special section of code in ADACS which reads an instruction from a previously defined data table and executes it. An example of a sequencer instruction listing is given in Appendix A. The sequencer can be used to control any process that consists of a series of steps, such as operating lockhoppers. The sequencer can be programmed to open the bottom valve, wait a specified period of time, and then close the valve. It could then pressurize the lockhopper to a specified pressure and open the top valve. After a period of time, the sequencer could close the top valve, depressurize the lockhopper, and start the sequence over. The logic used also has the controller expect feedback indicating successful completion of an operation (i.e., limit switch closure for a valve or a process suitable change for a set point adjustment). If the expected feedback does not occur, an error message is given and the system returns to a stable state. Up to 1000 instructions are allowed for each of the 8 possible sequencer tasks.

4.0 ADACS PROGRAM DESCRIPTION

Since ADACS is a very complex collection of computer programs, it is desirable to show the ways in which the routines interact. More detailed descriptions of the routines and common storage areas are presented in Appendices G and H.

4.1 ADACS CONTROL TABLE SETUP

Before ADACS can be run, the control tables must be created, placed in the proper format, and stored on disk to be read by the program. There are three basic types of control tables that must be set up before running - system files, slot processing control files, and sequencer control files.

4.1.1 System control files. These control files dictate the types of engineering units, conversions, limits, and alarms that are to be available for the slots on the system. The files are created in the proper format by typing them in through special setup programs (EDACQ). They then are stored on disk to be picked up by the programs ADACSX and MAST.

4.1.2 Slot processing control files. The slot processing control files are user specific. There are two types required for each user - a slot description file and a slot post processing file.

In order for a project to use a slot, that slot's definition must be entered in the project Acquisition Control File (ACF). These slot definitions are created, deleted, and modified in the program EDACQ and entered into the project ACF in the program CRACQ. These programs are described in detail in a METC Information Release^{1*}.

Slot definitions consist of a number of pieces of information. Including a slot number from zero to 5119, the data presented in a definition are:

- 1) an eight-character identifier, such as FI-304
- 2) a 32-character description, such as PRODUCER HUMIDIFIER OUTLET TEMP
- 3) a four-character engineering unit, such as DEGF
- 4) a function indicator to describe the type of processing to be done
- 5) a limit indicator to show the normal boundaries of the data values
- 6) a range indicator to show, for those slots associated with transmitters, the millivolt range of the transmitter.

To minimize the amount of memory needed to store the processing data of a given slot, a data compression scheme is used. Since engineering units,

* Superscript refers to listing in Bibliography.

functions, limits, and ranges each take only a limited number of distinct values, they are defined in system control tables and then referred to by entry number. Thus, each slot description is reduced to nine words.

All of the information about a slot is memory resident during data acquisition except the 32-character description, which is loaded in from disk only when needed.

Post processing allows the use of a set of predefined general purpose algorithms. These algorithms permit the calculation of values which are not simple conversions to engineering units from the millivolt values of single transmitters. Often, these calculations require the combining of values from more than one transmitter.

If a slot requires post processing, a twenty-word table is set up for it and stored in file POSTX.YY, where X is the user number and YY is the system number. The definitions for each slot are placed in the file in numerical order and consist of several lines giving the parameter definitions. The first line of a definition consists of the type of processing to be done (i.e., "FLOW"), the slot number the value is to be placed into, the 8-character ID for the slot, and an optional comment. The following lines consist of a number determining the data type of the third field (0 = integer, 1 = real, 2 = slot number, and 3 = octal number in first position of next line), the word in CPOST array for the storage of the data (one-half the first location for a real number, which requires two words for storage), the data value (integer, real, slot number, or a dummy when an octal number is called for), and an optional comment. The indexing into the CPOST array does not need to be in order and any word which is not used in a CPOST layout may be left out of the definition. Each definition ends with a line of three minus ones. The POSTX.YY file is usually created with the RDOS Text Editor.

4.1.3 Sequencer control files. These files are needed only by users that need sequencers. ADACS Sequential Control and Testing (SCAT) is described in detail in a METC Information Release².

Initially two source files must be created in RDOS, the SCAT source file and the Association name file. The SCAT source file consists of a list of SCAT commands, which are shown in Table 4.

The Association name file, which is used to correlate discrete input and output addresses to eight-character ASCII names, consists of a list of the discrete addresses and the associated names.

Table 4. Sequencer instructions

<u>Accumulator operations</u>	<u>Miscellaneous operations</u>
INCREMENT ACCUMULATOR	
DECREMENT ACCUMULATOR	WAIT BIT
CLEAR ACCUMULATOR	WAIT CARD
TEST ACCUMULATOR	OUTPUT BIT
COMPARE ACCUMULATOR	OUTPUT CARD
LOAD ACCUMULATOR	NO OPERATION
ADD ACCUMULATOR	DELAY
OUTPUT ACCUMULATOR	SCAN USER
LOAD SLOT	STOP
LOAD PARAMETER	IF BIT
DIVIDE	PAUSE
STORE ACCUMULATOR	PRINT
MULTIPLY ACCUMULATOR	LOAD TABLE
	PULSE
	EXECUTE
	WRITE DATA
<u>Accumulator - accumulator operations</u>	<u>Branching Operations</u>
ADD ACCUMULATOR ACCUMULATOR	BRANCH
MULTIPLY ACCUMULATOR ACCUMULATOR	GOTO SUBROUTINE
MOVE ACCUMULATOR ACCUMULATOR	COMPUTED GOTO
SUBTRACT ACCUMULATOR	RETURN
DIVIDE ACCUMULATOR	

The source files are then compiled into binary files. The SCAT file is compiled by the program VSEQ and the Association name file is compiled by the program VCOMPILE.

4.2 THE ADACS STARTUP TASKS

At startup the root program of ADACS, a routine called ADACSX, where X is the system number, is run. Its primary job is to establish the system parameters; it sets limits on what the system is capable of doing, such as the number of slots it has, the number of linear conversions available to it, and the features it will have.

To establish the parameters, the routine inputs them from a system file which is stored on magnetic disk and places them in the common areas SYSF0 and SYSACT. The common area SYSF0 contains eleven one-word arrays describing the extent of the system. The arrays are:

KBOX	-	number of analog input boxes
KCHAN	-	number of analog input channels
KLIM	-	number of limit values
KALM	-	number of alarm values
KFLF	-	number of linear functions
KFNF	-	number of engineering units
KUSR	-	maximum number of users allowed
KSEQ	-	maximum number of sequencers allowed
KDAC	-	number of DAC's
KRANGE	-	number of ranges

Common area SYSACT consists of a single twenty-word array called XACT. Each word of the array is used to define whether or not special features are allowed on the system and their extent:

Word 1	-	sequencers allowed? 1=yes, 0=no
Word 2	-	DAC's allowed? 1=yes, 0=no
Word 3	-	number of 0.1 second periods between DAC updates
Word 4	-	discrete (single bit) I/O allowed? 1=yes, 0=no
Word 5	-	ADACS system temperature monitors allowed? 1=yes, 0=no
Word 6	-	clock output available? 1=yes, 0=no
Word 7	-	watchdog features available? 1=yes, 0=no
Word 8	-	digital address used with the watchdog output
Word 9	-	maximum number of crt terminals allowed

- Word 10 - recall resolution - the number of points shown on a recall plot
- Word 11 - number of special slots available for dedicated calculations
- Word 12 - the calibration channel on each analog input box
- Word 13 - the number of tape drives available
- Word 14 - pneumatic scanners on the system? 1=yes, 0=no
- Word 15 - the ADACS system number
- Word 16 - Zebra disk on the system? 1=yes, 0=no
- Word 17 - maximum number of macros per terminal
- Word 18 - buffer size for buffering tape output
- Word 19 - size of the post processing control table
- Word 20 - number of user commands retained by ADACS

ADACSX also sets up system arrays where system control data will be stored by other routines. Most of these are of variable size, with the final size determined by the parameters above. For example, the array ENGU in common SYSF1 would be set up with the number of rows determined by the value in KENG and the number of columns as two words. Other commons and arrays set up by ADACSX are shown on the system wide array side of Figure 10.

The final ten percent of code in ADACSX is used to call in the ADACS startup task MAST.

MAST, which is the ADACS master task, has many duties associated with initializing ADACS. These include opening I/O channels, inputting control files, and sending startup messages to the users.

MAST starts by allocating extended memory. This is thirty two thousand (32,000) words of memory that are set aside by the operating system for data storage. MAST determines how this memory should be used.

Next MAST rewinds all of the tapes and opens the console channels. The console channels are the communications links to the master CRT which oversees all of ADACS. Additional channels are opened to the line printer and the user CRT's. The channels opened are summarized in Table 5.

Figure 10. ADACS Arrays.

Table 5. The FORTRAN channels and their uses.

<u>Channel</u>	<u>Use</u>
0	tape I/O
2	overlay file
3	writes users status hourly to disk for recovery
10	console output
11	console input
12	line printer
13	auxiliary teletype output
20-27	user 0-7 CRT I/O
30-37	user 0-7 recall files
40	master task temporary (to read control files)
41	command file storage
42	sequencer message storage
43	alarm storage
50-57	temporary user 0-7 restart files
62	temporary communications
63	master task and temporary communications

Startup messages are then printed along with the date and time that ADACS was brought on line. The digital inputs and outputs and their addresses are defined to the system.

MAST determines whether or not a Zebra disk(192 Mbyte) unit is associated with the system. If there is one it deletes any recall files that exist on it. If there isn't one, it reverts back to the ten magabyte Diablo disk unit.

Next, the tape drive units are set up. It starts this by requesting the operator to state which tape unit is to be used for initial data acquisition by responding with the proper unit's serial number. It asks for the amount of tape that is on the mounted tape and if the other tape unit is available in case problems occur with the first unit. It also asks whether the mounted tape is new or if it is to be a continuation of data on an old tape.

Log files and technician-entered data files are set up. These are used to store commands concerned with log messages.

Startup messages are written to tape. Header records, which specify which ADACS system is using the tape, what control files are used, etc., are written onto the tapes. The format for the header and data storage is provided in Appendix I.

MAST then reads the control files into the arrays set up by ADACSX. These are the data lookup tables that determine the types of processing available for slots, the available engineering units, etc.

Next, files are set up to retain the last 1000 sequencer messages and 1000 alarm messages.

Finally, MAST initiates the operation of various asynchronous tasks that will run until ADACS is shut down. These are routines that are necessary to keep the system running, and have to do primarily with I/O operations.

4.3 THE ASYNCHRONOUS TASKS

There are two basic types of asynchronous, core resident tasks in ADACS - those that are system oriented, which are started at system startup, and those that are user oriented, which are called in when they are needed to service a user. The difference between the two types is not always readily discernable for all routines - i.e., the USER tasks handle CRT input from both the master CRT and the user CRT's.

Each of the asynchronous tasks is given a priority based on its relative importance to the system. A routine with a high priority will have more computer time devoted to it. A task's priority is indicated by a priority number between zero and 255, zero being the highest priority.

Each task is also given an ID number between zero and 255. This is the only way that the operating system can tell one task from another. Different tasks may have the same priority but each task must have a unique ID number.

CLEANUP. The first of the system asynchronous tasks is CLEANUP. It is a watchdog task that performs periodic system diagnostic tests on the system and it cycles with a priority of 55. CLEANUP is also used to perform numerous small tasks that there is no place else to do.

CLEANUP's first job is to call a subroutine called CL2 which logs all of the single-bit memory errors which occur within the computer system so that a failing memory board can be detected and repaired or replaced before it can cause a data loss or a system crash.

The Eclipse computer operates with a sixteen-bit memory word and five parity bits are maintained to facilitate error detection within the sixteen bits. When a data value is written into a memory location, the memory hardware encodes the associated parity bits based on the data value. When a memory location is read, the hardware again computes the parity and compares it to the five parity bits. If there is a difference a memory error has occurred and based on the differences, the hardware can determine whether it is a single or multiple-bit error. If it is a single-bit error, the hardware will use the parity differences to correct it. It will then generate an interrupt to the system to indicate that the error occurred and then processing will resume unimpaired. If a multiple-bit error occurs, generally the hardware cannot correct for it and the system usually crashes.

CL2 uses entry point ERCCON in routine MUXINIT to detect the interrupt generated by a memory error. If the interrupt is detected CL2 will print a message describing the error, thus maintaining a log of all single-bit memory errors.

CLEANUP's next task is to check ADACS' internal temperature monitors to see if any of them are above the threshold temperature, which is set initially at 105°F. If a monitor exceeds the threshold CLEANUP will call CL1 to print a message stating that the threshold was exceeded. Then CLEANUP will raise the threshold for that monitor by 10°F.

Perhaps the most important task that CLEANUP performs is called a watchdog check. This is a check to make sure that the system is running normally.

At least once every ten seconds, each data acquisition task must go to a specific memory location and set a bit. Every fifteen or twenty seconds CLEANUP will check the status of the flag bits. If they are all set the routine will assume that everything is running properly and it will clear the bits. If for any reason a data acquisition task gets hung up it will fail to set its flag bit and CLEANUP will detect this on its next check. CLEANUP will then give the task one more chance to set its bit. If it fails to do so, CLEANUP will call CL1, which will print messages about the deadlock. Then CL1 will go through a number of steps to clear software flags and, using entry point MUXINIT in routine MUXINIT, try to reinitialize the data acquisition hardware in an attempt to get the system running again. This feature clears over 80% of the system deadlocks and, in many cases, compensates for programming errors.

To make sure the other asynchronous tasks are running properly, CLEANUP checks some of the common system resources such as the line printer, which must be shared by all of the users. It is possible for two tasks to each require two common resources at one time and both need the same resource. For example, task A needs the tape unit and the line printer and it gets the tape unit, and task B needs the tape unit and the line printer and it gets the printer. The two routines get deadlocked because they cannot get both of the resources they require. Every effort is made to program against such occurrences, but with 20 simultaneous tasks and 10-20 common resources, deadlocks are difficult to totally eliminate.

If CLEANUP finds that a piece of equipment has been tied up too long, it will clear the request flags and release all of the tied up equipment. Thus, in the example above, CLEANUP would release the tape unit and the line printer and the two tasks would have to start over.

CLEANUP's next task is to call CL1 in once an hour to perform a check on the floating point arithmetic hardware unit, which performs all of the floating point calculations for the computer. CL2 goes through a set of calculations with the unit that tests each of its functions and then compares the answer it receives from the unit to the number it should get. If any errors are noted, the routine sounds an alarm.

Finally, once an hour CLEANUP will call CL2 in to write messages concerning the status of each user to disk. In the event of a system crash, these will be used to restore the users to the point they were at within an hour of the crash.

USER. This routine takes input from the CRT terminals and places it in a buffer area for further processing. USER may be run up to nine times simultaneously to service the master CRT and the eight user CRT's. The priorities for the different versions of user range from 79 for the one servicing the master CRT to 87 for the one servicing the user seven terminal.

When a terminal key is pressed an interrupt is generated. The operating system senses the interrupt and identifies the terminal that generated it. It then routes the input to the proper version of USER, as identified by its ID number, which then places the input in a buffer.

SCANR. This is a very small task that scans the signals from the ADC's and places them into memory. SCANR cycles with a priority of 10.

LEVDET. This routine is run only on the B-4 system. It is used to perform a number of special functions associated with the gas producer and it cycles with a priority of 120.

LEVDET's first task is to determine if a rupture disk has blown. There are currently four rupture disks on the producer that have thermocouples associated with them on the low pressure side. If a rupture disk blows, hot gas will flow over the thermocouple, causing its temperature to rise. By scanning the slots associated with the rupture disk thermocouples and comparing the signal with a threshold temperature, currently set at 100°F, LEVDET can determine if any of the disks have blown. If any of the thermocouples is sensing a temperature above the threshold LEVDET will sound an alarm.

The next job performed by LEVDET is to make sure that the producer flare is lit. There are three thermocouples surrounding the flare in a horizontal plane, each placed sixty degrees from the other two. Theoretically, no matter which direction the wind is blowing at least one of the thermocouples will still be in the flame and, therefore, heated. LEVDET examines the slots for these thermocouples and if at least one of them is above the threshold temperature, which is currently set at 130°F, the flare is assumed to be lit. If all of the thermocouples are below the threshold LEVDET assumes that the flare has gone out and it sounds an alarm.

Next, LEVDET is used to control the humidifier outlet temperature by controlling the water flow to the spray nozzles. Once each second LEVDET compares the humidifier gas outlet temperature to the setpoint temperature, which is set by technicians in the producer control room. If there is a difference between the two temperatures it will adjust the nozzle flows in order to make them the same, thus attempting to maintain a constant outlet temperature. The nozzles must be turned on in a sequential order from nozzle number one at the top down to nozzle four such that no nozzle operates at less than fifty percent of its rated output.

LEVDET's final task is to scan the outputs of the five nuclear density detectors which are located on the producer shell and perform some calculations based on the data. First, it computes an estimate of the total coal and ash mass within the bed of the producer. It prints this estimate on the line printer and displays it in the producer control room so that the technicians can monitor the bed mass. Then it tries to determine if there are any voids developing within the bed. If a void is detected LEVDET will sound an alarm.

CHROM. CHROM is the chromatograph handling routine. It is run on the B-4 system only and it cycles with a priority of 50.

Producer gas samples are analyzed for each of the eight gas components shown in Table 6, by one of four gas chromatographs. As it analyzes the gas, the chromatograph passes the values to CHROM via a pair of slots. The first slot passes a signal representing the relative proportion of the particular gas and the second slot conveys the identity of the particular gas. The second slot is also used to indicate when the first slot is stable enough to use. CHROM can handle all four of the chromatographs simultaneously by polling each of the slot pairs individually.

Table 6. A list of the gases analyzed for by the B-4 gas chromatographs and the associated full scale percentages.

Hydrogen (H ₂)	0-40%
Carbon Dioxide (CO ₂)	0-20%
Ethane (C ₂ H ₆)	0-3%
Hydrogen Sulfide (H ₂ S)	0-3%
Oxygen (O ₂)	0-5%
Nitrogen (N ₂)	0-80%
Methane (CH ₄)	0-10%
Carbon Monoxide (CO)	0-40%

CHROM scans the outputs of a chromatograph once each second and stores the millivolt value and the identity of the gas associated with it. Once it has the readings for all eight gases it calls the subroutine GCOUT to convert the signals into meaningful numbers.

GCOUT takes the millivolt readings and compares each one to the full scale readings for that gas as shown in Table 6, thus converting them into relative percentages. It then uses these numbers along with the constants in Table 7 to compute the gas's average molecular weight, gross Btu content, net Btu content, specific gravity relative to air, and density in grams per liter.

Table 7. The constants used in the GCOUT calculations.

Gas	Molecular Weight	Density, g/cm ³	Net Btu/ft ³	Gross Btu/ft ³
H ₂	2.016	0.0899	275	325
CO ₂	44.01	1.9770	0	0
C ₂ H ₆	30.07	1.3423	1641	1792
H ₂ S	34.08	1.5390	596	647
O ₂	32.00	1.4290	0	0
N ₂	28.02	1.2506	0	0
CH ₄	16.04	0.7160	913	1013
CO	28.01	1.2500	322	322

After GCOUT performs its calculations CHROM takes the results and formats a report which it sends to the line printer and to an auxiliary teletype which is located in the chromatograph room. It also writes records on the tape and disk units.

SPECIAL. This routine is used to set up for and call the subroutines that do the special slot calculations. It runs with a priority of 130 which is the lowest priority in ADACS.

The special slot calculations are set up to enable a user to compute quantities which are often abstract in nature and are not measurable. The computations are usually very complex and unique, not lending themselves to the data table techniques. Therefore, they are hard coded into their own subroutines.

SPECIAL first determines which slots are needed for a particular set of calculations. It then has those slots scanned and places the data in an array where the subroutines can access it. SPECIAL then calls the proper subroutine to do the calculations. The name of the subroutine is SCALX, where X is the user number.

There are currently two users of the special slots, both on the B-4 system. The first of these is the gas producer, which is user one. Some of the things calculated by SCAL1 are humidifier efficiency, cyclone velocities, and heat loss to cooling water. The other user is the Erosion/Corrosion project, which uses SCAL4 to calculate such quantities as the gas molecular weight, the mass flow of combustion air into the facility, and air-to-fuel ratios in the combustion cells.

DACIO. This routine has a priority of 20. It has the responsibility of outputting data values to the digital-to-analog converters (DACs). DACIO constantly scans an array, pulling data values out and outputting them to the proper DACs. If a routine needs to output something to a DAC it places the value in the proper location in the array. The next time DACIO cycles the correct voltage will be outputted by the DAC. Typically DACIO cycles from 1 to 10 times per second.

DACIO also provides output to the ADACS clocks in the project control rooms. The clock accepts 6 binary-coded-decimal digits to yield a clock precisely synchronized with the ADACS clock.

SCAN. This is a small routine with a priority of 5 that updates some SCAN counters within ADACS on the B-1 system.

TGATASK. This routine is used only on the B-4 system and it cycles with a priority of 16. Its task is to accept data from a thermogravimetric analyzer and put it in an appropriate format.

RCOMM. This is the core-resident portion of the communications interpreter. It has a priority of 70.

RCOMM polls the various USER tasks to see if one has a full line of input. When a full line is found RCOMM will place it in a buffer. It will then call INPUT for interpretation.

INPUT looks at the first word of the input line and uses a hashing algorithm to reduce the word to sixteen bits. It then classifies the word as belonging to one of several categories, which are listed in Table 8.

Table 8. Possible classifications of first words of the crt inputs.

<u>Category</u>	<u>Interpretation</u>
-1	unrecognizable input
0	primary command word
1	integer number
2	valid command word but not a primary command word
3	real number
4	octal number

In deciphering a word as a primary command INPUT compares it against a list of the 68 possible primary commands. If a match is made INPUT returns the number of the overlay that is associated with that command then returns control to RCOMM. RCOMM then calls the overlay for further processing of the input. A listing of some of the possible overlays is given in Table 9.

Table 9. Some of the overlays used for deciphering CRT input.

<u>Overlay Routine</u>	<u>Associated Commands</u>	<u>Overlay Routine</u>	<u>Associated Commands</u>
COM01	Stop	COM26	Calibration commands
COM02	Start or Restart	COM27	Identify
COM03	Schedule	COM28	Recall
COM04	Shutdown	COM29	Disk file commands
COM05	Defining macros	COM30	Recall
COM06	Output	COM31	Load
COM07	Crt mode	COM32	Set Recall
COM08	Update and Two-step	COM33	On-line calculator
COM09	Add and Remove	COM34	Open and Close
COM11	Print system	COM35	Control Sequencer
COM12	Printer user & Print file	COM36	Analyze
COM14	Status	COM37	Ultrasonic commands
COM15	Setcal	COM38	Move, Restore and Reset
COM16	Executing macros	COM39	Lookup
COM17	Dump	COM40	Recall
COM19	Display	COM41	Status
COM21	Suspend and Activate	COM42	File
COM23	Dump	COM43	Snapshot
COM24	Log and Data	COM44	Manual file data commands
COM25	Modify		

If INPUT doesn't recognize the command a message will be written to the user's CRT stating this.

CRT. CRT cycles with a priority of 100. Periodically it determines when output is needed at one of the user's CRT terminals. When it senses that output is required, it sets a flag that activates REPORT, which prepares the data and places it in a common. When REPORT finishes it sets another flag and CRT picks up the data for outputting. Once the data has been outputted CRT clears the flags and frees the data storage area for the next request.

CRT services all eight users on a round-robin polling basis.

LPOUT. This routine, which cycles with a priority of 75, is the line printer output task. It polls each of the users to determine when one needs output at the line printer. When it senses that output is required, it sets a flag that is sensed by REPORT, which prepares the data. When REPORT finishes it sets a flag that LPOUT senses. LPOUT then prints the data on the line printer and clears all of the flags.

REPORT. This is the user's main data acquisition task. It can be running up to eight times simultaneously, once for each user. Each version has a system identification number corresponding to the user it serves - 40 for user 0 to 47 for user 7. All versions run with the same priority, 40.

REPORT is initiated when ADACS receives a command to start scanning a project. It immediately goes through a series of startup tasks. First, it allocates its portion of extended memory and BUF, a buffer area. Next, it reads in the slot processing control tables from disk and stores them in memory. It also writes the control files onto the tape to assist in later processing the tape. It writes headers on the tape concerning the project. It sets up the disk recall files. Finally, it writes startup messages on the CRTS and the line printer.

REPORT then enters an infinite loop that prepares data for outputting. First it checks a series of flag words to determine where output is needed. REPORT maintains a group of scan counters, one for each output device, which it compares against the system clock to determine when output is due. Then it loads the control table containing the list of slots that need to be scanned and calls XFER to do the scanning. Next, it calls REDUC to convert the data, which is now a digital representation of an analog signal, into meaningful engineering units.

REDUC looks at the slot processing control tables to determine the type of processing needed for each slot. REDUC handles three of the six types of slot processing. The first is linear conversions, in which the signal is simply multiplied by a slope and added to the intercept. Next are nonlinear conversions, where REDUC must use the range of the signal to look up the slope and intercept of a piecewise linear function. Finally, REDUC handles switch functions, in which the signal is compared to a threshold value and a judgment is made as to its on or off state. After processing a slot REDUC places it in a buffer with the raw signals.

Next, REPORT collects the special slots that were created by SPECIAL, which runs concurrently, in the hard coded calculations and places them in the buffer. It then calls the routine POSTX, where X is the system number, which does the post processing calculations.

POSTX is a system specific routine but all systems' versions have a similar structure. To process a slot, POSTX first goes to the post processing control table and pulls out a twenty-word control list for that slot. The first two words of this list specify the type of processing required by the slot, usually with a four-character ASCII mnemonic; a partial list of the available calculations is presented as Table 10. The last eighteen words specify data for the calculation, and are defined differently for each calculation. Some of the possible post processed calculations are presented in Appendix D.

Table 10. Some of the available post-processing calculations

ASCII mnemonic	Calculation
FLOW	gas flow, liquid flow, sat. steam w/pressure correction, sat. steam w/temperature correction, rotameter calculation, superheated steam w/temperature and pressure
POLY	polynomial calculations
SO2	MELOY SO ₂ logarithmic conversion
MATH	general math calculations
MTHX	special math calculations w/log option
BCD	binary coded decimal conversion
INTG	slot integrator
RATE	rate calculations
SUMS	slot sums
ENBL	decodes slots w/time multiplexed data

After the post processing is finished REPORT starts outputting the data. It outputs to the DACs by changing the necessary values in an array. It sends output to the CRTs and line printer if it is needed. REPORT also performs slot limit checking at this point if it is required. Finally, it fills in the tape and disk header messages and outputs to those units if it is required.

If ADACS is commanded to stop scanning the project, a flag bit is set, and REPORT calls the routine ENDIT. ENDIT writes shutdown messages to the tape and disk units, the CRTs and the line printer. It then shuts down REPORT and itself.

SEQUENCE. This routine is used to implement the sequential control feature of ADACS. It runs with a priority of 75 and can be used up to eight times simultaneously, once for each user.

Before this task can be run, a sequencer control file, which is a data table of instructions each consisting of four sixteen-bit words, must be loaded into extended memory from disk by giving ADACS the load sequencer command. Then SEQUENCE will run when ADACS receives the start sequencer command.

SEQUENCE loads an instruction from the data table and decodes it. It then performs the action dictated by the instruction. It will continue to decode instructions until it either encounters a STOP instruction or ADACS receives a command to stop the sequencer. SEQUENCE is limited to processing ten instructions per second to keep it from using all of the available real time, thus keeping the lower priority tasks from running.

There are two major subroutines associated with SEQUENCE, SEQMES and SEQ2MES. These routines are concerned primarily with sequencer messages. SEQMES takes a control flag from SEQUENCE and, based on its value, performs one of nine functions, ranging from outputting a value to a DAC to selecting one of a number of messages and printing it. SEQ2MES performs jobs associated with outputting data or messages to tape and line printer units.

LCAL. This is the ADACS analog input calibrator. It resides in an overlay and when running, normally every twenty minutes, it cycles with a priority of 13.

LCAL is called by a loop in MUXID which counts clock interrupts, thus timing the next occurrence of LCAL. LCAL first calls in a small subroutine called TSK 13 to obtain the necessary resources. Then it calibrates the analog input boxes and determines the appropriate calibration factor to ensure a conversion accuracy of $\pm 0.1\%$.

The inputs are checked by applying a known voltage to each of the input channels and determining the calibration factors. Since the gains of the input channels are known, the theoretical value can be compared to the actual value. If the difference is greater than one percent, the channel is assumed to be going bad. This routine can detect a single bad channel out of hundreds, usually allowing it to be repaired or replaced before it goes completely bad.

RMTEH. This is the magnetic tape error handling routine. It is called in as needed by TWRITE and it cycles with a priority of 35.

If TWRITE encounters a problem in writing data to tape it calls RMTEH. RMTEH immediately calls the first of three subroutines associated with it, MTEH1.

The first thing that MTEH1 does is to sound an alarm, announcing that there is a problem with the tape unit. It then tries to determine whether the failure is soft or hard. A soft failure indicates that probably only a few data values are bad and that data acquisition should continue with the same tape. A hard failure, however, indicates that the tape unit has a serious malfunction and that data acquisition should continue with the other unit if it is available.

If the problem is in the hardware MTEH1 will determine if another tape drive is available. If there isn't the routine will print out error messages and return to RMTEH. If a new drive is available MTEH1 will call MTEH2.

MTEH2 closes the channel to the bad tape drive and rewinds the tape. It then initializes the new tape, makes sure it is working properly, and then writes a continuation record onto the tape indicating that this tape is continued from a previous one. It then calls in MTEH3.

If the tape switch was successful MTEH3 will send the appropriate messages indicating this to the crts and line printer. It then releases the old tape unit and returns to RMTEH, which then would cancel itself until the next tape problem.

5.0 BIBLIOGRAPHY

1. METC Information Release #936, "ADACS Slot Processing" by David R. Goff and Ralph Crane, 2/09/81.
2. METC Information Release #693, Rev. 1, "ADACS Sequential Controls and Testing (SCAT)" by David R. Goff, 12/01/80.
3. METC Information Release #384, "ADACS Features and Capabilities" by David R. Goff, 8/01/79.
4. METC Information Release #160, Rev. 5, "Updated Magnetic Tape Format for the Automatic Data Acquisition and Control System (ADACS) by David R. Goff, 4/25/80.
5. METC Information Release #661, "ADACS Command Line Manual" by David R. Goff, 2/15/80.

APPENDIX A
Sequencer Example
Taken from IR No. 693

TER 4 = 0.0; TOTAL CYCLE C
STEP 119

DIR:DPO FILE:VSLTURED APR 20, 1982 16: 3:32 MEIC PRT 0 10

PARAMETER 4 = 0.0 ; TOTAL CYCLE COUNT SETTING
PARAMETER 5 = 0.0 ; TCYCNT UPDATE SUBROUTINE OR PAUSE FLAG
PARAMETER 6 = 100.0 ; LTCYCNT -PRESET LIMIT
PARAMETER 7 = 0.0 ; LTCYCNT -VALUE TO DATE
PARAMETER 8 = 0.0 ; STARTING TCYCNT
PARAMETER 9 = 0.0 ; STARTING TACOUNT
PARAMETER 10 = 0.0 ; STARTING TPOCOUNT
PARAMETER 11 = 5.1 ; VALVE NUMBER (I.E. 3.1->A3, 5.2->B5, ETC)
PARAMETER 12 = 4 ; TEST RUN NUMBER
PARAMETER 13 = 10.0 ; SLURRY TRANSFER SECTION DELAY TIME (SECONDS)
PARAMETER 14 = 3.0 ; SLURRY TRANSFER VALVE DELAY TIME (SECONDS)
PARAMETER 15 = 1000.0 ; SYSTEM PRESSURIZING OR TEMPERATURE RATE

THIS IS THE COMPUTER CONTROLLED TEST SEQUENCE FOR THE VALVE SLURRY TEST UNIT (VSLTU)

VERSION 6-REVISION 3-APR 15, 1982 (DRS)

***** REVISION HISTORY *****

1-20-81 V 5-REV 2 ADDED CODE TO EXERCISE FLOW LEG VALVES EVERY 5 AUTO CYCLES
1-27-81 V 5-REV 3 UPGRADED 'WRITE DATA' STATEMENTS TO INCLUDE 14PE & CYCLE COUNT
1-30-81 V 5-REV 4 REDEFINED PARAMS 11 & 12, ADDED VALVE NUMBER & RUN NUMBER
3-9-81 V 6-REV 0 REFINE CODE, COMBINE SUBROUTINES & VALVE MOVEMENTS
11-18-81 V 6-REV 1 REMOVED FV211 REFERENCES
12-31-81 V 6-REV 2 RED VALVE MODIFICATIONS--DELAY AFTER TEST VALVE ACTUATION
4-15-82 V 6-REV 3 TV CLOSE ONLY IF LSH208 OFF, CLOSE FV202 & OPEN FV201 SIMUL,
CHANGE MESS. 3 & 11, SET FLOW LEG VALVES AFTER SCAN USER, & DELETE
USAGE OF BOTH TV EXERCISE SUBROUTINES (AMB. & PRESS.)

DISCRETE INPUTS ARE ON CARDS WITH DEVICE CODES 61 OCTAL (61K) AND 71 OCTAL (71K)

DISCRETE OUTPUTS ARE ON CARD WITH DEVICE CODE 65 OCTAL (65K)

REGISTER A OF 61K FOR VALVE OPEN (OP) IS ASSIGNED AS FOLLOWS: BIT 0 THRU BIT 15:
COMPON=BIT0, FV201-OP=BIT1, FV202-OP=BIT2, FV203-OP=BIT3, FV204-OP=BIT4, FV205-OP=BIT5,
TV7-OP=BIT6, FV210-OP=BIT7, FV211-OP=BIT8, FV246-OP=BIT9, FV247-OP=BIT10,
FV248-OP=BIT11, FV244-OP=BIT12, FV245-OP=BIT13, FV226-OP=BIT14, FV227-OP=BIT15

REGISTER B OF 61K FOR VALVE OPEN (OP), MOTOR ON (ON), OR SWITCH HIGH (HI) IS ASSIGNED AS
FOLLOWS: BIT 0 THRU BIT 15; FV209-OP=BIT0, FV228-OP=BIT1, FV229-OP=BIT2, FV230-OP=BIT3,
FV231-OP=BIT4, FV232-OP=BIT5, FV233-OP=BIT6, FV234-OP=BIT7, FV235-OP=BIT8, FV236-OP=BIT9,
FV237-OP=BIT10, A2-ON=BIT11, P2-ON=BIT12, LSH208=BIT13, LSH234HI=BIT14, P204-HI=BIT15

REGISTER C OF 61K FOR VALVE CLOSED (CL) IS ASSIGNED AS FOLLOWS: BIT 0 THRU BIT 15:
FV209-CL=BIT0, FV201-CL=BIT1, FV202-CL=BIT2, FV203-CL=BIT3, FV204-CL=BIT4, FV205-CL=BIT5,
TV7-CL=BIT6, FV210-CL=BIT7, FV211-CL=BIT8, FV246-CL=BIT9, FV247-CL=BIT10,
FV248-CL=BIT11, FV244-CL=BIT12, FV245-CL=BIT13, FV226-CL=BIT14, FV227-CL=BIT15

REGISTER A OF 71K FOR VALVE CLOSED (CL) OR MOTOR OFF (OF) IS ASSIGNED AS FOLLOWS: BIT 0
THRU BIT 15; FV228-CL=BIT0, FV229-CL=BIT1, FV230-CL=BIT2, FV231-CL=BIT3, FV232-CL=BIT4,
FV233-CL=BIT5, FV234-CL=BIT6, FV235-CL=BIT7, FV236-CL=BIT8, FV237-CL=BIT9, A2-OF=BIT10,
P2-OF=BIT12

REGISTER A OF 65K FOR VALVE OPERATION IS ASSIGNED AS FOLLOWS: BIT 0 THRU BIT 15:
FV201=BIT0, FV202=BIT1, FV203=BIT2, FV204=BIT3, FV205=BIT4, FV206=BIT5, SPARE=BIT6, FV246=BIT7,
FV247=BIT8, FV248=BIT9, FV244=BIT10, FV245=BIT11, FV226=BIT12, FV227=BIT13, FV228=BIT14,
FV229=BIT15. IF BIT IS SET TO ZERO, VALVE GOES TO NORMAL POSITION, I.E. ON OR NO.

TER 4 = 0.0; TOTAL CYCLE C

DIR:DPO FILE:VSLTURED APR 20, 1982 16: 3:32 MEIC PRT 0 10

TER 4 = 0.0; TOTAL CYCLE C DIR:DPO FILE:VSLTURED APR 20, 1982 16: 3:32 METC PRT 0.10
STEP 119

```
13 IF (PARAMETER 0 LT CONSTANT 0) GOTO STEP 16 ; PERFORM ATMOS/PRESS IF NON-NEGATIVE
14 IF (PARAMETER 3 LT CONSTANT 1) GOTO STEP 16 ; DO LEAK SERIES IF TEST PRESSURES POSITIVE
15 CALL SUBROUTINE 222 ; CLOSE THE TEST VALVE
15 GOTO STEP 133
16 IF (PARAMETER 0 LT CONSTANT 0) GOTO STEP 229 ; PERFORM OTHER IF PARAMETER 0 IS NEG
C ; PERFORM ATMOS CYCLE IF PARAMETER 0
17 IF (PARAMETER 0 GE CONSTANT 1) GOTO STEP 19 ; IS ZERO, PRESS CYCLE IF POSITIVE
18 IF (ACCUM 1 GE PARAMETER 4) STEP 119 ELSE STEP 21 ; ATMOS CYCLES COMPLETE ?
19 IF (ACCUM 2 GE PARAMETER 4) GOTO STEP 119 ; PRESS CYCLES COMPLETE ?
21 CALL SUBROUTINE 222 ; CLOSE THE TEST VALVE
22 PRINT MESSAGE 19 ACCUM 1 ; "TOTAL ATMOSPHERIC CYCLE COUNT= "
22 PRINT MESSAGE 20 ACCUM 2 ; "TOTAL PRESSURE CYCLE COUNT= "
22 PRINT MESSAGE 24 ACCUM 3 ; "LYCYCNT= "
23 OUTPUT ACCUM 11 DAC 8 RATE=PARAMETER 15 ; SET DAC 8 TO 1.0 VOLT
24 CLOSE PUEN1 ; ENABLE FV227 PURGING ON CLOSING
C
C **** THE FOLLOWING ADDED 1-20-81 TO EXERCISE FLOW LEG VALVES EVERY 5 AUTO CYCLES ***
C
24 INCREMENT ACCUM 12 ; BUMP COUNT BETWEEN EXERCISES
24 IF (ACCUM 12 LT CONSTANT 5) GOTO STEP 25 ; TRANSFER IF NOT TIME TO EXERCISE
24 CLEAR ACCUM 12 ; CLEAR THE COUNTER
24 OUTPUT CARD 65K REG A MASK 000760K STATE 000760K ; CLOSE FV244, 245, 246, 247, 248
24 OUTPUT CARD 65K REG B MASK 170000K STATE 170000K ; OPEN FV230, 231, 232, 233
24 WAIT CARD 61K REG C MASK 000174K STATE 000000K ; CHECK FV244, 245, 246, 247, 248 CLOSED
24 WAIT CARD 61K REG B MASK 017000K STATE 000000K ; CHECK FV230, 231, 232, 233 OPEN
C
C *****
C
25 OUTPUT CARD 65K REG A MASK 122766K STATE 022002K ; CHECK OF VALVE STATUS, FV201, 203, FV209,
25 OUTPUT CARD 65K REG B MASK 177761K STATE 000000K ; 246, 247, 248, 249, 245, 227,
26 WAIT CARD 61K REG A MASK 150175K STATE 006400K ; 228, 234, 235, 237, PREPU OPEN
27 WAIT CARD 61K REG B MASK 057777K STATE 017107K ; FV210, FV230, 231, 232, 233, 236,
28 WAIT CARD 61K REG C MASK 050175K STATE 150175K ; PVENA CLOSED, AGITATOR AND PUMP ON,
29 WAIT CARD 71K REG A MASK 057770K STATE 040670K ; LSH208, LSH231, PDAH LOW, COMPON
31 OUTPUT CARD 65K REG A MASK 040011K STATE 040001K ; OPEN FV202, 226, 229
32 WAIT BIT FV202-OP TRUE ; WAIT FOR FV202 TO OPEN
32 WAIT BIT FV226-OP TRUE ; WAIT FOR FV226 TO OPEN
32 WAIT BIT FV229-OP TRUE ; WAIT FOR FV229 TO OPEN
33 IF (PARAMETER 5 LT CONSTANT 1) GOTO STEP 35 ; PAUSE FOR OPERATOR SYSTEM CHECK ?
34 CALL SUBROUTINE 310 ; GO PAUSE & UPDATE CYCNT
35 CLOSE FV226 ; BLOCK DRAIN FOR FILLING OF
36 DELAY PARAMETER 14 ; CHARGE TANK, TIME DELAY TO PREVENT
37 CLOSE PREPU ; SLURRY TRAP, ENABLE FV227 PREPURGING
37 CLOSE FV227
37 OPEN PREPU ; DISABLE FV227 PREPURGING
40 CLOSE FV201 ; ALLOW FILLING OF CHARGE TANK BY
C ; CLOSING BYPASS
40 PRINT MESSAGE 5 ; "FILLING CHARGE TANK"
41 WAIT BIT LSH208 TRUE ; SIGNIFIES CHARGE TANK FULL
42 OUTPUT CARD 65K REG A MASK 140000K STATE 000000K ; OPENS SLURRY BYPASS AND CLOSSES INLET
43 WAIT BIT FV201-OP TRUE ; TO CHARGE TANK
43 WAIT BIT FV202-CL TRUE
44 IF (PARAMETER 0 LT CONSTANT 1) STEP 45 ELSE STEP 77 ; ATMOS IF ZERO, PRESS IF POSITIVE
C
C THE FOLLOWING CODE PERFORMS THE ATMOSPHERIC CYCLE
C
45 PRINT MESSAGE 6 ; "PERFORMING ATMOSPHERIC CYCLE"
```

TER 4 = 0.0; TOTAL CYCLE C DIR:DPO FILE:VSLTURED APR 20, 1982 16: 3:32 METC PRT 0.10

TER 4 = 0.0; TOTAL CYCLE C
STEP 119

DIR:DPO FILE:VSLTURED

APR 20, 1982 16: 3.32 RETC PRT 0.10

```
45 OUTPUT CARD 65K REG A MASK 040001K STATE 000000K ; PREPARE TO PRESSURIZE SECTION 1
45 WAIT BIT FV202-CL TRUE ; AND CHARGE TANK BY CLOSING MIXTURE
45 WAIT BIT FV229-CL TRUE
46 DELAY PARAMETER 14 ; INLET PATH TO CHARGE TANK
47 OUTPUT CARD 65K REG A MASK 020002K STATE 000000K ; CLOSE FV203, 228
48 WAIT BIT FV203-CL TRUE
49 WAIT BIT FV228-CL TRUE ; OPEN FV234, 236, 237
50 OPEN FV234
51 CLOSE FV235
52 OPEN FV236
53 OPEN FV237
57 CALL SUBROUTINE 345 ; PRESSURIZE SYSTEM
58 OPEN FV227 ; OPEN CHARGE TANK OUTLET
59 OPEN FV226
60 CLOSE FV236 ; PREVENT SLURRY CLOGGING OF FV236
61 DELAY PARAMETER 13 ; MAINTAIN PRESSURE ON SECTION 1 AND
C ; CHARGE TANK FOR VARIABLE PERIOD
64 CALL SUBROUTINE 360 ; DEPRESSURIZE SYSTEM
67 CALL SUBROUTINE 216 ; ALLOW SLURRY TRANSFER TO MIX TANK
67 OPEN FV228 ; OPEN TV7, FV228 AND 229
67 OPEN FV229
67 DELAY PARAMETER 13 ; TIME DELAY FOR SLURRY TRANSFER
68 INCREMENT ACCUM 1,3 ; BUMP TACOUNT, L1CYCND
70 CALL SUBROUTINE 222 ; CLOSE THE TEST VALVE
73 IF (ACCUM 3 GE PARAMETER 6) GOTO STEP 133 ; PERFORM LEAK SERIES 2
74 IF (ACCUM 1 LT PARAMETER 4) GOTO STEP 22 ; REPEAT ATMO'S CYCLES IF INCOMPLETE
76 IF (PARAMETER 3 LT CONSTANT 1) STEP 119 ELSE STEP 133 ; DO LEAK SERIES IF TEST PRESSURES SPECIFIED
C
C THE FOLLOWING CODE PERFORMS THE PRESSURE CYCLE
C
77 PRINT MESSAGE 7 ; "PERFORMING PRESSURE CYCLE"
78 OUTPUT CARD 65K REG A MASK 040001K STATE 000000K ; PREPARE FOR PRESSURIZING SECTIONS
79 WAIT BIT FV202-CL TRUE ; 1 AND 2 AND CHARGE TANK BY CLOSING
79 WAIT BIT FV229-CL TRUE
80 DELAY PARAMETER 14 ; MIXTURE INLET PATH TO CHARGE TANK
81 OUTPUT CARD 65K REG A MASK 020002K STATE 000000K ; AND INLET PATH TO MIX TANK. CLOSE
82 WAIT BIT FV203-CL TRUE ; FV202, 229, TIME DELAY, CLOSE FV203, 228
82 WAIT BIT FV229-CL TRUE
87 OPEN FV234 ; PREPARE FOR SECTIONS 1 AND 2 AND
87 OPEN FV235 ; CHARGE TANK PRESSURIZING. OPEN
88 OPEN FV236 ; FV234, 235, 236, 237
89 OPEN FV237
93 CALL SUBROUTINE 345 ; PRESSURIZE SYSTEM
C ; PRESSURIZE SECTIONS 1 AND 2 AND
C ; CHARGE TANK
C ; OPEN CHARGE TANK OUTLET
94 OPEN FV227
95 OPEN FV226
96 CLOSE FV236 ; PREVENT SLURRY CLOGGING OF FV236
97 DELAY PARAMETER 13 ; TIME DELAY FOR SLURRY TO SECTION 1
98 CALL SUBROUTINE 216 ; OPEN THE TEST VALVE
98 DELAY PARAMETER 13 ; ALLOW SLURRY TRANSFER TO SECTION 2
99 INCREMENT ACCUM 2,3 ; BUMP TPCOUNT, L2CYCND
103 CALL SUBROUTINE 222 ; CLOSE THE TEST VALVE
106 CALL SUBROUTINE 360 ; DEPRESSURIZE SYSTEM
111 OPEN FV228 ; ALLOW SLURRY TRANSFER TO MIX TANK
112 OPEN FV229
113 DELAY PARAMETER 13 ; TIME DELAY FOR SLURRY TRANSFER
```

TER 4 = 0.0; TOTAL CYCLE C

DIR:DPO FILE:VSLTURED

APR 20, 1982 16: 3.32 RETC PRT 0.10

TER 4 = 0.0; TOTAL CYCLE C
NBSTEP 119

DIR DPO FILE:VSLTURED APR 20,1982 16: 3:32 MEIC PRT 0.10

115 IF (ACCUM 3 GE PARAMETER 5) GOTO STEP 133 ; PERFORM LEAK SERIES IF IT IS TIME
116 IF (ACCUM 2 LT PARAMETER 4) GOTO STEP 22 ; REPEAT PRESSURE IF TOTAL CYCLES NOT COMPLETE
118 IF (PARAMETER 3 LT CONSTANT 1) STEP 119 ELSE STEP 133 ; DO LEAK SERIES IF TEST PRESSURES POSITIVE
C
C THE FOLLOWING CODE PERFORMS THE HALT SEQUENCE
C
119 PRINT MESSAGE 12 ; "ENTERING THE HALT SEQUENCE"
121 CALL SUBROUTINE 360 ; DEPRESSURIZE SYSTEM
124 PRINT MESSAGE 17 ; "SETTING TEMPERATURE" TO ZERO
125 OUTPUT ACCUM 11 DAC 9 ; SET DAC 9 TO 1 VOLT
126 OUTPUT CARD 65K REG A MASK 174763K STATE 000000K ; PLACE VALVES INTO FAIL MODES,
127 OUTPUT CARD 65K REG B MASK 177411K STATE 000000K ; SHUT OFF AIR, VENT OFF PRESSURE,
128 OUTPUT CARD 65K REG C MASK 140000K STATE 000000K ; OPEN FV201, TV7, FV246, 247, 248, 244,
C ; 245, 234, 235, 237, PREPU,
C ; PUEN1, PUEN2, CLOSE FV202, 203, 204, 205
C ; 228, 229, 230, 231, 232, 233, 235,
C ; PVENA REMAIN IN EXISTING STATES
129 DELAY 10 ; TIME DELAY TO ALLOW FV229 TO CLOSE TO
130 OUTPUT CARD 65K REG A MASK 174777K STATE 000000K ; PREVENT SLURRY LOSS, OPEN FV226 AND 227,
C ; SET OTHER VALVES TO FAIL MODES AS ABOVE
131 OUTPUT BIT COMPMODE FALSE ; RELEASE COMPUTER CONTROL
131 PRINT MESSAGE 8 ; "VSLTU SEQUENCER STOPPED"
131 STORE ACCUM 0 PARAMETER 8 ; LOAD TCYCNT VALUE TO DATE
131 STORE ACCUM 1 PARAMETER 9 ; LOAD TACOUNT VALUE TO DATE
131 STORE ACCUM 2 PARAMETER 10 ; LOAD TPCOUNT VALUE TO DATE
131 STORE ACCUM 3 PARAMETER 7 ; LOAD LTCYCNT VALUE TO DATE
131 PRINT MESSAGE 4 ACCUM 0 ; "TOTAL CYCLE COUNT= "
131 PRINT MESSAGE 19 ACCUM 1 ; "TOTAL ATMOSPHERIC CYCLE COUNT=
131 PRINT MESSAGE 20 ACCUM 2 ; "TOTAL PRESSURE CYCLE COUNT=
131 PRINT MESSAGE 24 ACCUM 3 ; "LTCYCNT=
132 STOP
C
C THE FOLLOWING CODE PERFORMS THE LEAK TEST SERIES
C
133 WRITE DATA RT=CONSTANT 5 ; RECORD TYPE
133 WRITE DATA C1=ACCUM 0 ; WRITE THE TOTAL CYCLE COUNT
133 WRITE DATA ST=SLOT 433 ; SCAN THE SLURRY TEMPERATURE
133 LOAD ACCUM 4 1.0 ; SET COUNTER FOR SECTION 1 TO 2
133 GOTO STEP 135 ; LEAK TEST ONLY
134 LOAD ACCUM 4 0.0 ; SET COUNTER FOR SECTION 2 TO 1 AND
C ; 1 TO 2 LEAK TESTS
135 LOAD ACCUM 13 1.0
136 PAUSE MESSAGE 3 ACCUM 0 ; "ENT. LEAK. CL PD VAL. CONT WHEN COMPTCYCNT=
137 CALL SUBROUTINE 360 ; DEPRESSURIZE SYSTEM
138 OUTPUT CARD 65K REG A MASK 040011K STATE 000010K ; CLOSE FV202, 226, 229
138 WAIT BIT FV202-CL TRUE
138 WAIT BIT FV226-CL TRUE
138 WAIT BIT FV229-CL TRUE
138 DELAY PARAMETER 14 ; TIME DELAY FOR SLURRY TRANSFER
138 CLOSE FV203 ; ENABLE FV227 PREPURGING
138 CLOSE PREPU ; DISABLE FV227 PREPURGING
138 CLOSE FV227
138 OPEN PREPU
138 CLOSE FV228
138 LOAD ACCUM 9 0.0
138 IF (ACCUM 13 LT ACCUM 9) STEP 275 ELSE STEP 193 ; CYCLE TEST VALVE WHILE PRESSURIZED ?
140 OUTPUT ACCUM 11 DAC 8 RATE=PARAMETER 15 ; SET DAC 8 TO 1 VOLT

TER 4 = 0.0; TOTAL CYCLE C

DIR: DPO FILE: VSLTURED APR 20,1982 16: 3:32 MEIC PRT 0.10

TER 4 = 0.0; TOTAL CYCLE C
NBSTEP 119

DIR:DP0 FILE:VSLTURED

APR 20, 1982 16: 3: 42 METC PRT 0.10

140 OUTPUT CARD 65K REG A MASK 014000K STATE 010000K
141 OUTPUT CARD 65K REG B MASK 177400K STATE 143400K
142 WAIT CARD 61K REG A MASK 006000K STATE 002000K
143 WAIT CARD 61K REG B MASK 017740K STATE 003240K
144 WAIT CARD 61K REG C MASK 006000K STATE 004000K
145 WAIT CARD 71K REG A MASK 017740K STATE 014500K
149 PRINT MESSAGE 9
150 IF (ACCUM 5 LT ACCUM 9) GOTO STEP 152
151 LOAD ACCUM 9 1.0
151 CALL SUBROUTINE 222
152 OUTPUT CARD 65K REG A MASK 000760K STATE 000720K
152 WAIT CARD 61K REG A MASK 000174K STATE 000164K
152 WAIT CARD 61K REG C MASK 000174K STATE 000010K
156 LOAD TABLE INDEX=5 ACCUM 6
156 PRINT MESSAGE 2 ACCUM 6
156 ADD ACCUM 6 500.
157 MULTIPLY ACCUM 6 0.002
158 PRINT MESSAGE 15
158 OUTPUT ACCUM 6 DAC 8 RATE=PARAMETER 15
C *****
158 DELAY 70
C *****
159 LOAD ACCUM 8 244
159 LOAD SLOT 395=ACCUM 7, SLOT 3393=ACCUM 15
160 LOAD ACCUM 6 1160.
161 IF (ACCUM 7 GE ACCUM 6) GOTO STEP 187
164 OPEN FV245
165 CLOSE FV244
165 DELAY 10.0
166 INCREMENT ACCUM 8
166 LOAD SLOT 396=ACCUM 7, SLOT 3393=ACCUM 15
168 IF (ACCUM 7 GE ACCUM 6) GOTO STEP 187
171 OPEN FV246
172 CLOSE FV245
172 DELAY 10.0
173 INCREMENT ACCUM 8
173 LOAD SLOT 392=ACCUM 7, SLOT 3393=ACCUM 15
175 IF (ACCUM 7 GE ACCUM 6) GOTO STEP 187
178 OPEN FV247
179 CLOSE FV246
179 DELAY 10.0
180 INCREMENT ACCUM 8
180 LOAD SLOT 393=ACCUM 7, SLOT 3393=ACCUM 15
182 IF (ACCUM 7 GE ACCUM 6) GOTO STEP 187
185 OPEN FV248
186 CLOSE FV247
186 DELAY 10.0
186 INCREMENT ACCUM 8
186 LOAD SLOT 394=ACCUM 7, SLOT 3393=ACCUM 15
187 LOAD TABLE INDEX=5 ACCUM 6
188 PRINT MESSAGE 21
189 WRITE DATA RT=CONSTANT 4
189 WRITE DATA C1=ACCUM 0
189 WRITE DATA TP=ACCUM 6
189 WRITE DATA AP=SLOT 399
189 WRITE DATA FV=ACCUM 8
189 WRITE DATA FL=ACCUM 15

; PREPARE TO PRESSURIZE SECTION 1;
; CLOSE FV205, 232, 235, 236, 237;
; OPEN FV204, 230, 231, 234, 236
;
; "PERF. SECT. 1 TO 2 LEAK TEST SEPTER"
; IS THE TEST VALVE ALREADY CLOSED?
; RESET VALVE CLOSED SELECTION
; CLOSE THE TEST VALVE
; CHECK VALVE STATUS AND PREPARE FOR
; CORRECT SECTION PRESSURIZING: OPEN FV244
; CLOSE FV246, 247, 248, 249
; LOAD A TEST PRESSURE
; "LEAK TEST-PRESET PRESSURE=" "
; SHIFT FOR 500 TO 1500 PSI
; CONVERT TO VOLTS
; "PRESSURIZING SYSTEM"
; SET UP PRESET PRESSURE
;
; PRESSURE RAMPING SETTLING TIME
;
; LOAD 244 FLOW LEG IDENTITY
; IS FLOW GT 20 PER CENT OF 244 SCALE?
; IF NO JUMP TO NEXT FLOW LEG
; IF YES, MAKE LEAK MEASUREMENT
; FLOW LT 20 PER CENT OF FULL SCALE
; , SET UP NEXT HIGHEST SCALE
; ALLOW FOR FLOW SETTLING TIME
; SET 245 FLOW LEG IDENTITY
; IS FLOW GT 20 PER CENT OF 245 SCALE?
; IF YES, MAKE LEAK MEASUREMENT
; FLOW LT 20 PER CENT OF FULL SCALE
; , SET UP FOR NEXT HIGHEST SCALE
; ALLOW FOR FLOW SETTLING TIME
; SET 246 FLOW LEG IDENTITY
; IS FLOW GT 20 PER CENT OF 246 SCALE?
; IF YES, MAKE LEAK MEASUREMENT
; FLOW LT 20 PER CENT OF FULL SCALE
; , SET UP FOR NEXT HIGHEST SCALE
; ALLOW FOR FLOW SETTLING TIME
; SET 247 FLOW LEG IDENTITY
; IS FLOW GT 20 PER CENT OF 247 SCALE?
; IF YES, MAKE LEAK MEASUREMENT
; FLOW LT 20 PER CENT OF FULL SCALE
; , SET UP FOR NEXT HIGHEST SCALE
; ALLOW FOR FLOW SETTLING TIME
; SET 248 FLOW LEG IDENTITY
; LOAD THE RAW MILLIVOLT
; LOAD TEST PRESSURE VALUE
; LEAK TEST VALUES HEADING
; RECORD TYPE FLAG
; RECORD CYCLE COUNT
; DESIRED TEST PRESSURE
; ACTUAL TEST PRESSURE
; FLOW LEG #
; FLOW LEG SCFM

A-8

TER 4 = 0.0; TOTAL CYCLE C

DIR:DP0 FILE:VSLTURED

APR 20, 1982 16: 3: 32 METC PRT 0.10

TER 4 = 0.0; TOTAL CYCLE C
STEP 119

DIR:DPO FILE:VSLTURED

APR 20, 1982 16: 3:32

METC PRT 0.10

189 WRITE DATA FT=SLOT 434 ; FLOW LEG EXIT TEMP
189 WRITE DATA FP=SLOT 402 ; FLOW LEG EXIT PRESSURE
189 WRITE DATA RW=ACCUM 7 ; RECORD RAW MILLIVOLTS
189 SCAN USER ; RECORD ALL SLOTS ON TAPE
189 OPEN FV244 ; OPEN FV244 & CLOSE FV245,
189 WAIT BIT FV244-OP TRUE ; FV246, FV247, & FV248
189 OUTPUT CARD 65K REG A MASK 000720K STATE 000720K
189 WAIT CARD 61K REG C MASK 000164K STATE 000000K
190 CALL SUBROUTINE 360 ; DEPRESSURIZE SYSTEM
192 INCREMENT ACCUM 5 ; BUMP THE TABLE INDEX
192 CALL SUBROUTINE 216 ; OPEN THE TEST VALVE
193 IF (ACCUM 5 GT PARAMETER 3) GOTO STEP 195 ; HAVE WE DONE ALL THE PRESSURES YET ?
194 IF (ACCUM 4 LT CONSTANT 1) STEP 197 ELSE STEP 140 ; SET UP TO REPEAT
195 INCREMENT ACCUM 4 ; BUMP LCYCOUT
195 LOAD ACCUM 5 1.0 ; RESET TABLE INDEX
C ; IF LCYCOUT EQ 2, LEAK TESTS ARE
196 IF (ACCUM 4 LT CONSTANT 2) STEP 194 ELSE STEP 206 ; COMPLETE, GO TO SET VALVE STATUS
197 OUTPUT ACCUM 11 DAC 8 RATE=PARAMETER 15 ; SET DAC 8 TO 1 VOLT
197 OUTPUT CARD 65K REG A MASK 014000K STATE 004000K ; PREPARE TO PRESSURIZE SECTION 2
198 OUTPUT CARD 65K REG B MASK 177400K STATE 035400K ; CLOSE FV204, 230, 231, 234, 237
199 WAIT CARD 61K REG A MASK 006000K STATE 004000K ; OPEN FV205, 232, 233, 235, 236
200 WAIT CARD 61K REG B MASK 017740K STATE 014440K
201 WAIT CARD 61K REG C MASK 006000K STATE 002000K
202 WAIT CARD 71K REG A MASK 017740K STATE 003300K
204 PRINT MESSAGE 10 ; "PERF. SECT. 2 TO 1 LEAK TEST SERIES"
205 GOTO STEP 150 ; PREPARE FOR SECT 2 TO 1 LEAK TEST
206 CLOSE FV205 ; CLOSE FV205
206 OUTPUT CARD 65K REG A MASK 020766K STATE 020002K ; SET VALVE STATUS, CLOSE FV230, 231,
207 OUTPUT CARD 65K REG B MASK 177400K STATE 000000K ; 232, 233, 236, OPEN FV203, 244, 245,
208 WAIT CARD 61K REG A MASK 010175K STATE 000000K ; 246, 247, 248, 227, 228, 234, 235, 237
209 WAIT CARD 61K REG B MASK 057740K STATE 017100K
210 WAIT CARD 71K REG A MASK 057740K STATE 040640K
212 OUTPUT CARD 65K REG A MASK 040C11K STATE 040001K ; OPEN FV202, 226, 229
213 WAIT BIT FV202-OP TRUE
213 WAIT BIT FV226-OP TRUE
213 WAIT BIT FV229-OP TRUE
214 CALL SUBROUTINE 216 ; OPEN THE TEST VALVE
214 LOAD ACCUM 3 0.0 ; RESET LEAK CYCLE COUNTER
214 PAUSE MESSAGE 11 ACCUM 0 ; "LEAK SER COMP, OPEN PU VAL-TOYONT=" "
215 GOTO STEP 16
C
C THE FOLLOWING SUBROUTINE OPENS THE TEST VALVE AND RESETS THE STOP CLOCK
C
SUBROUTINE 216
216 PULSE BIT RESETCLK (TRUE, FALSE) ; RESET THE STOP CLOCK
218 OPEN TV7 ; OPEN THE TEST VALVE
219 DELAY 15 ; *** RED VALVE DELAY ***
221 RETURN
C
C THE FOLLOWING SUBROUTINE CLOSSES THE TEST VALVE, RESETS AND SCANS THE STOP CLOCK,
C INCREMENTS THE TOTAL CYCLE COUNTER, AND SELECTS USAGE OF TOYCNT UPDATE SUBROUTINE
C
SUBROUTINE 222
222 WAIT BIT LSH208 FALSE ; SIGNIFIES CHARGE TANK EMPTY
222 PULSE BIT RESETCLK (TRUE, FALSE) ; RESET THE STOP CLOCK
223 PRINT MESSAGE 4 ACCUM 0 ; "TOTAL CYCLE COUNT=" "
223 IF (PARAMETER 5 LT CONSTANT 1) GOTO STEP 225 ; ENTER NEW TOYCNT VALUE?

TER 4 = 0.0; TOTAL CYCLE C

DIR:DPO FILE:VSLTURED

APR 20, 1982 16: 3:32

METC PRT 0.10

TER 4 = 0.0; TOTAL CYCLE C
NBSTEP 119

DIR: DPO FILE: VSLTURED APR 20, 1982 16: 3:32 METC PRI 0.10

A-10

```
224 CALL SUBROUTINE 310 ; UPDATE TCYCNT
225 CLOSE TV7 ; CLOSE THE TEST VALVE
226 DELAY 15 ; *** RED VALVE DELAY ***
227 INCREMENT ACCUM 0 ; BUMP TCYCNT
228 RETURN
C
229 GOTO STEP 119 ; FOR FUTURE SUBROUTINE ADDITIONS
C
C THE FOLLOWING SUBROUTINE SETS VALVE STATES FOR VALVE ACTUATION TIME AND TORQUE MEASUREMENTS
C
SUBROUTINE 260
260 OUTPUT CARD 65K REG B MASK 177400K STATE 001000K ; CLOSE FV230, 231, 232, 233
C ; OPEN FV234, 235, 236, 237
261 OUTPUT CARD 65K REG A MASK 000760K STATE 000000K ; OPEN FV244, 245, 246, 247, 248
262 WAIT CARD 61K REG A MASK 000174K STATE 000000K ; CHECK FV244, 245, 246, 247, 248
263 WAIT CARD 61K REG B MASK 000740K STATE 000000K ; CHECK FV234, 235, 236, 237
264 WAIT CARD 71K REG A MASK 017000K STATE 000000K ; CHECK FV230, 231, 232, 233
265 RETURN
C
C THE FOLLOWING CODE OPENS AND CLOSES THE TEST VALVE UNDER SYSTEM PRESSURE
C
275 CALL SUBROUTINE 260 ; SET VALVE STATUS
285 CALL SUBROUTINE 345 ; PRESSURIZE SYSTEM
C ; PRESSURIZE SECTIONS 1 AND 2
288 LOAD PARAMETER 2 ACCUM 6 ; GET THE PRESET PRESSURE LIMIT
288 CALL SUBROUTINE 216 ; OPEN THE TEST VALVE
290 WRITE DATA RT=CONSTANT 2 ; RECORD TYPE FLAG
290 WRITE DATA CI=ACCUM 0 ; RECORD TOTAL CYCLE COUNT
291 WRITE DATA TP=ACCUM 6 ; DESIRED TEST PRESSURE
292 WRITE DATA AP=SLOT 399 ; ACTUAL TEST PRESSURE
292 WRITE DATA FO=SLOT 290 ; RECORD TIME TO OPEN VALVE
293 CALL SUBROUTINE 222 ; CLOSE THE TEST VALVE
294 WRITE DATA RT=CONSTANT 3 ; RECORD TYPE FLAG
294 WRITE DATA CI=ACCUM 0 ; RECORD CYCLE COUNT
295 WRITE DATA TP=ACCUM 6 ; DESIRED TEST PRESSURE
296 WRITE DATA AP=SLOT 399 ; ACTUAL TEST PRESSURE
296 WRITE DATA FO=SLOT 290 ; RECORD TIME TO CLOSE VALVE
297 INCREMENT ACCUM 13 ; BUMP SELECTION ACCUM
298 PAUSE MESSAGE 13 ; "PERFORM TORQUE MEASUREMENTS? CONTINUE"
299 GOTO STEP 137
C
C THE FOLLOWING SUBROUTINE IS USED FOR UPDATING THE TEST VALVE TOTAL CYCLE COUNT VALUE
C DURING SEQUENCER OPERATION
C
SUBROUTINE 310
310 PAUSE MESSAGE 22 ; "ENTER PARAMETER VALUES"
311 LOAD PARAMETER 8 ACCUM 0 ; LOAD TCYCNT VALUE TO DATE
312 PRINT MESSAGE 4 ACCUM 0 ; "TOTAL CYCLE COUNT= "
313 RETURN
C
C THE FOLLOWING SUBROUTINE OPENS AND CLOSES THE TEST VALVE UNDER AMBIENT PRESSURE
C
SUBROUTINE 320
320 CALL SUBROUTINE 360 ; DEPRESSURIZE SYSTEM
324 CALL SUBROUTINE 260 ; SET VALVE STATUS
326 CALL SUBROUTINE 216 ; OPEN THE TEST VALVE
328 LOAD SLOT 290 ACCUM 15 ; SAVE THE TIME TO OPEN
```

TER 4 = 0.0; TOTAL CYCLE C

DIR: DPO FILE: VSLTURED APR 20, 1982 16: 3:32 METC PRI 0.10

TER 4 = 0.0; TOTAL CYCLE C
STEP 119

DIR: DFO FILE: VSLTURED

APR 20, 1982 16: 3:32

METC PRT 0 10

```
331 CALL SUBROUTINE 222 ; CLOSE THE TEST VALVE
332 WRITE DATA RT=CONSTANT 1 ; RECORD TYPE FLAG
332 WRITE DATA C1=ACCUM 0 ; RECORD CYCLE COUNT
332 WRITE DATA SC=SLOT 290 ; TEST VALVE ACTUATION TIME
333 WRITE DATA SO=ACCUM 15 ; RECORD TIME TO OPEN
335 PAUSE MESSAGE 13 ; "PERFORM TORQUE MEASUREMENTS? CONTINUE"
336 RETURN
```

C
C THE FOLLOWING SUBROUTINE IS USED TO DEPRESSURIZE THE SYSTEM

```
4C
SUBROUTINE 345
345 LOAD PARAMETER 2 ACCUM 6 ; GET THE PRESET PRESSURE LIMIT
346 ADD ACCUM 6 500. ; SHIFT FOR 500 TO 2500 PSI
347 MULTIPLY ACCUM 6 0.002 ; CONVERT TO VOLTS
350 OPEN FV204 ; OPEN AIR SUPPLY VALVE
351 OPEN FV205
352 PRINT MESSAGE 15 ; "PRESSURIZING SYSTEM"
353 OUTPUT ACCUM 6 DAC 8 RATE=PARAMETER 15 ; SET UP PRESET PRESS (500 PSIG-MIN)
354 LOAD PARAMETER 2 ACCUM 6 ; LOAD PRESSURE
355 MULTIPLY ACCUM 6 0.95
356 LOAD SLOT 404 ACCUM 15 ; LOAD THE CURRENT PRESSURE
357 IF (ACCUM 15 GE ACCUM 6) GOTO STEP 359 ; TRANSFER IF PRESSURE > 95% SETPOINT
357 DELAY 2 ; DELAY 2 SECONDS
358 BRANCH STEP 356 ; WAIT FOR THE PRESSURE
359 RETURN
```

C
C THE FOLLOWING SUBROUTINE IS USED TO ENSURE THE SYSTEM HAS DEPRESSURIZED TO 20 PSIG

A-11

```

SUBROUTINE 360
360 PRINT MESSAGE 16 ; "DEPRESSURIZING SYSTEM"
361 OUTPUT ACCUM 11 DAC 8 RATE=PARAMETER 15 ; RAMP THE PRESSURE DOWN
365 LOAD ACCUM 10 20.
366 LOAD SLOT 404 ACCUM 14 ; IS PT224 MEASURING 20 PSIG OR LESS?
367 IF (ACCUM 14 GT ACCUM 10) GOTO STEP 375 ; IF YES, CONTINUE.
368 LOAD SLOT 400 ACCUM 14 ; IS PT213 MEASURING 20 PSIG OR LESS?
369 IF (ACCUM 14 GT ACCUM 10) GOTO STEP 375 ; IF YES, CONTINUE.
370 LOAD SLOT 401 ACCUM 14 ; IS PT214 MEASURING 20 PSIG OR LESS?
371 IF (ACCUM 14 GT ACCUM 10) GOTO STEP 375 ; IF YES, CONTINUE.
372 LOAD SLOT 399 ACCUM 14 ; IS PT209 MEASURING 20 PSIG OR LESS?
373 IF (ACCUM 14 GT ACCUM 10) GOTO STEP 375 ; IF YES, CONTINUE.
374 RETURN
```

C
375 DELAY 2 ; 2 SECOND TIME DELAY
376 GOTO STEP 366
C

```
MESSAGE 1 SEQUENCER STARTED
MESSAGE 2 LEAK TEST-PRESET PRESSURE=
MESSAGE 3 ENT. LEAK SER, CLOSE PURGE & HAND VALVES, CONT WHEN COMP-TCYCNT=
MESSAGE 4 TOTAL CYCLE COUNT=
MESSAGE 5 FILLING CHARGE TANK
MESSAGE 6 PERFORMING ATMOSPHERIC CYCLE
MESSAGE 7 PERFORMING PRESSURE CYCLE
MESSAGE 8 SEQUENCER STOPPED
MESSAGE 9 PERFORMING SECTION 1 TO 2 LEAK TEST SERIES
MESSAGE 10 PERFORMING SECTION 2 TO 1 LEAK TEST SERIES
MESSAGE 11 LEAK SER COMP., OPEN PURGE & HAND VALVES, CONT WHEN COMP-TCYCNT=
MESSAGE 12 ENTERING THE HALT SEQUENCE
```

TER 4 = 0.0; TOTAL CYCLE C

DIR: DFO FILE: VSLTURED

APR 20, 1982 16: 3:32

METC PRT 0 10

TER 4 = 0.0; TOTAL CYCLE C
NRSTEP 119

DIR:DPO FILE:VSLTURED APR 20, 1982 16: 3:32 METC PRT 0.10

MESSAGE 13 PERFORM TORQUE MEASUREMENTS? CONTINUE WHEN COMPLETE.
MESSAGE 15 PRESSURIZING SYSTEM
MESSAGE 16 DEPRESSURIZING SYSTEM
MESSAGE 17 SETTING TEMPERATURE
MESSAGE 19 TOTAL ATMOSPHERIC CYCLE COUNT=
MESSAGE 20 TOTAL PRESSURE CYCLE COUNT=
MESSAGE 21 LEAK SERIES DATA FOLLOWS :
MESSAGE 22 ENTER PARAMETER VALUES (MUST ENTER TCYCNT)
MESSAGE 23 PARAMETERS LOADED
MESSAGE 24 LTCYCNT=
C
TABLE 50, 100, 150, 200, 250
C
END

A-12

TER 4 = 0.0; TOTAL CYCLE C

DIR:DPO FILE:VSLTURED APR 20, 1982 16: 3:32 METC PRT 0.10

APPENDIX B
Description of ADACS Users

The Morgantown Energy Technology Center's (METC) Automatic Data Acquisition and Control System (ADACS) currently provides support to many METC projects. This is an overview of the projects connected to each ADACS.

THE B-4 SYSTEM

The B-4 ADACS is the largest of the four current systems. It presently serves six test units.

The 42-inch Gas Producer and Cleanup System - This project consists of a 24-ton per day stirred, fixed-bed coal gasifier and a novel "dry" tar removal system. At 200 psig the gasifier is capable of producing 120,000 scfh gas with a heating value of approximately 160 Btu per scf. The cleanup system consists of a number of components designed to remove dust particulates, tar, oil, sulfur (to meet EPA requirements), and trace metals (for turbine gas requirements). The components are:

<u>COMPONENT</u>	<u>PURPOSE</u>
● Cyclone	Remove entrained coal dust particles
● Humidifier	Accomplish first stage cooling resulting in condensation and partial removal of high boiling tar/oil
● Tar separator	Remove high boiling tar/oil aerosols
● Venturi scrubber	Remove remaining dust and high boiling tar/oil
● Electrostatic precipitator	Particulate removal/alternate to venturi scrubber
● Direct cooler	Accomplish final gas cooling causing condensation of water vapor and light oils
● Indirect coolers	Heat exchangers which transfer heat from the gas to the plant cooling water system
● Stretford unit	Removes H ₂ S from the gas stream and converts it to water and elemental sulfur
● Alkali scrubber	Remove trace contaminants entrained in the gas stream

The Holmes-Stretford Desulfurization Unit - This is packaged unit from Peabody Process Systems that is used in the gasifier cleanup system. It removes sulfur compounds from the product gas stream and converts the sulfur to its elemental form. The unit has a large number of associated processes that are not directly involved with removing the sulfur from the gas stream. Therefore, it is set up as a separate ADACS user.

The Ceramic Corrosion/Erosion Project - The objective of this project is to create a technology base for ceramic materials to be used in stationary gas power turbines using high-temperature, coal-derived, low-Btu gas. Ceramic samples are tested in two ways - statically, using low velocities (to determine the effects of corrosion), and dynamically, using high velocities (to determine the combined effects of corrosion and erosion). The test units use gas from the METC gasifier for testing.

The Slurry Drying System - This project is currently inactive.

The Two 18-inch Fluidized Beds - These units are part of a project to provide the research and development support needed to encourage the utilization of fluidized-bed technology in the industrial and utility sectors. An extensive program of low-grade fuel studies (including anthracite refuse, high sulfur coals, lignites, oil shales, and discarded tires) is in progress to provide operational design data and demonstrations of low-grade fuel feasibility.

Features of the METC program include:

- Available FBCs include a 4-inch, 6-inch, and two 18-inch-diameter hot units, and a 6-foot x 6-foot cold model.
- The units are fully instrumented for collecting data on emissions and combustion characteristics.
- Additional study areas include in-bed gas-probe development, various bed-level detection devices, distributor design, two-stage combustion, and a cyclone inside the FBC.

THE B-15 SYSTEM

This is the second largest ADACS and it is utilized only by the METC Lockhopper Valve Testing and Development Projects.

The goal of the State-of-the-Art Lockhopper Valve Testing and Development Project is to survey the capabilities of existing valves and work with the valve manufacturers to upgrade current valve technology to meet the valve requirements for lockhopper systems on coal conversion and utilization pilot plants and early demonstration facilities.

The goal of the Prototype Lockhopper Valve Testing and Development Project is to develop new valve designs to extend the capabilities of lockhopper valves for coal conversion systems.

There are four test units associated with the valve testing projects. In each of these, the internal leakage, external leakage, actuation time, and operating force are monitored. The test units are described below.

The Valve Static Test Unit - This unit is used for initial checkout of a valve using clean, inert gas at a range of pressures to 1600 psig. A range of operating temperatures to 600°F is attained using induction heaters.

The Valve Dynamic Test Unit - This unit tests the valve's ability to operate with a solids flow. A range of pressures to 1600 psig and temperatures to 600°F are used. The higher temperatures are attained using induction heaters. Limestone or aluminum oxide are used as the solids.

The Valve Hot Solids Test Unit - The VHSTU uses a fluidized-bed gas combustor to heat the solids which are then cycled through the valves. Again, limestone or aluminum oxide is used, and a variety of pressures and temperatures is used.

The Valve Slurry Test Unit - This unit is used to test valves determined to have potential for slurry service.

THE B-1 SYSTEM

The B-1 ADACS serves a small research laboratory, and it provides only high speed data acquisition.

THE ADVANCED GASIFICATION LABORATORY SYSTEM

This ADACS is not yet operational. The system will serve a number of advanced coal gasification projects, including a small fluidized bed gasifier and a small entrained bed gasifier.

APPENDIX C
Thermocouple Conversion Tables

TABLE A
COPPER-CONSTANTAN

Temperature Range	MV Range	Slope	Intercept
-80°F to 110°F	-6.4 to -1.15	45.59	153.24
110°F to 155°F	-1.15 to 0.17	41.85	148.25
155°F to 257°F	0.17 to 3.24	38.49	149.40
257°F to 356°F	3.24 to 6.27	35.32	159.33
356°F to 404°F	6.27 to 7.75	33.60	169.55
404°F to 440°F	7.75 to 8.85	32.94	174.50
440°F to 506°F	8.85 to 10.91	32.11	181.81
506°F to 692°F	10.91 to 16.60	30.42	200.94
692°F to 1379°F	16.60 to 38.8	29.28	219.02

TABLE B
CHROMEL-ALUMEL

Temperature Range	MV Range	Slope	Intercept
-80°F to -16°F	-4.99 to -3.70	50.82	171.31
-16°F to 80°F	-3.70 to -1.60	46.19	154.28
80°F to 298°F	-1.60 to 3.40	43.51	149.59
298°F to 504°F	3.40 to 8.00	44.75	146.67
504°F to 742°F	8.00 to 13.50	43.20	159.06
742°F to 1237°F	13.50 to 25.20	42.27	171.54
1237°F to 1471°F	25.20 to 30.60	43.10	150.86
1471°F to 1717°F	30.60 to 36.10	44.47	109.07
1717°F to 1939°F	36.10 to 40.90	46.10	50.42
1939°F to 2300°F	40.90 to 48.32	47.65	-12.94

Note: Both Copper-Constantan and Chromel-Alumel functions take into effect the 150°F cold junction reference ovens used for all direct thermocouple inputs on ADACS.

APPENDIX D
Post Processing

The following discussion about post processing calculations is excerpted from METC Information Release No. 936.

Flows

All flow calculations are based on the same algorithm and, therefore, provide space for all parameters, although some parameters are not used in all flow calculations (exception: words 14-15 which can contain one of two different parameters, or neither, depending upon the type of flow calculation).

The first two words of the CPOST layout (0 and 1) contain the word "FLOW" to indicate a flow calculation is to be done. Words 2 and 3 contain the slot numbers for a converted temperature (DEGF) and a converted pressure (PSIG) to use in the calculation (neither used in all calculations). Words 4 through 9 are used in all calculations and contain in order: the slot number of the raw flow transmitter (MV), a number indicating type of flow calculation to be done (gas, liquid, etc.), full-scale flow at design conditions, the offset voltage of the transmitter in MV, and the voltage span of the transmitter in MV. Words 10 and 11 are set aside for the design temperature (DEGF) for those calculations which require it, while words 12 and 13 are for the design pressure (PSIG). Words 14 and 15 are normally left unused except as noted later. Word 16 is used to store the address (usually defined as octal in POSTX.YY) of a discrete bit which is used to enable the particular calculation (may not be used and set to -1). Word 17 contains the state of the discrete bit (0 or 1) which enables the calculation if a discrete is used. Words 18 and 19 contain a real offset value which is added to the flow calculation (if not used, must be set to zero).

Gas Flow

Gas flow calculations use both the converted temperature and converted pressure slots as well as the design temperatures and pressure.

CPOST Layout for Gas Flow (Flow corrected by pressure and temperature)

Words 0-1 "FLOW"
 Word 2 Slot number of converted temperature (DEGF)
 Word 3 Slot number of converted pressure (PSIG) (-1 if unused)
 Word 4 Slot number of raw flow transmitter output (MV)
 Word 5 Type = $\emptyset$
 Words 6-7 Full-scale flow at design conditions (real value)
 Word 8 MV offset voltage (i.e., 1000) may be changed at run time
 Word 9 MV range of transmitter (i.e., 4000)
 Words 10-11 Design temperature (DEGF) (real value)
 Words 12-13 Design pressure (PSIG) (real value)
 Words 14-15 Unused
 Word 16 Discrete bit address used to enable calculation (-1 if unused)
 Word 17 Discrete state to enable calculation (0 or 1)
 Words 18-19 Offset value to be added to the result of the flow calculation (real value)

The equation used for "GAS" flows is as follows:

let FFLOW = Value in words 6 and 7.

OFFSET = Value in words 18 and 19.

MVOFF = Value in word 8.

MVSPAN = Value in word 9.

FTRAN = Flow transmitter output (millivolts), word 4 contains slot number.

PRESS = Pressure correction (psig) -- Word 3 contains slot number.

TEMP = Temperature correction ($^{\circ}$ F) -- Word 2 contains slot number.

DT = Design temperature -- Words 10 and 11.

DP = Design pressure -- Words 12 and 13.

then

$$\text{FLOW} = \text{FFLOW} \sqrt{\frac{(\text{FTRAN} - \text{MVOFF})}{\text{MVSPAN}} \frac{\text{PRESS} + 14.4}{\text{DP} + 14.7} \frac{\text{DT} + 460.}{\text{TEMP} + 460.}} + \text{OFFSET}$$

Example:

FLOW	3077	FI-1B	;FLUID AIR FLOW (LOW)	
2	2	490	;TEMPERATURE SLOT	NOMINAL TEMP = 60 DEGF
2	3	483	;PRESSURE SLOT	NOMINAL PRESS = 100 PSIG
2	4	485	;RAW FLOW TRANSMITTER SLOT	HEAD = 10", Orifice = 0.873", LINE = 2" (2.067" I.D.)
0	5	0	;GAS FLOW	
1	3	5344.	;FLOW MULTIPLIER	NOMINAL FULL-SCALE FLOW = 5344 SCFH
0	8	1000	;OFFSET VOLTAGE IN MV	
0	9	4000	;SPAN IN MV	

1	5	60.	;DESIGN TEMPERATURE
1	6	100.	;DESIGN PRESSURE
0	16	-1	;DISCRETE ADDRESS NOT USED
1	9	0.0	;FLOW OFFSET CORRECTION
-1	-1	-1	
FLOW	3206	FV-126	;LEG 1 -- 2ND ORIFICE AIR FLOW
2	2	300	;CONVERTED TEMPERATURE SLOT
2	3	319	;CONVERTED PRESSURE SLOT
2	4	316	;RAW FLOW TRANSMITTER
0	5	0	;GAS FLOW
1	3	5.0	;5.0 SCFM FULL-SCALE FLOW
0	8	1000	;MV OFFSET VOLTAGE
0	9	4000	;MV RANGE
1	5	60.	;60. DEGF DESIGN TEMPERATURE
1	6	100.	;100. PSIG DESIGN PRESSURE
3	16	0	
014062K			;DISCRETE ADDRESS TO ENABLE CALCULATION
0	17	0	;ENABLE STATE
1	9	0.0	;NO OFFSET
-1	-1	-1	

Liquid Flow (Flow with no pressure or temperature corrections)

For liquid flows, neither the converted temperature and pressure slot numbers nor the design temperature and pressure are used. Note that this option could be used for gas flows that do not have pressure and temperature measurements available.

CPOST Layout for Liquid Flow

Words 0-1	"FLOW"
Words 2-3	Unused
Word 4	Slot number of raw flow transmitter output (MV)
Word 5	Type = 1
Words 6-7	Full-scale flow at design conditions (real value)
Word 8	MV offset voltage (i.e., 1000) may be changed at run time
Word 9	MV range of transmitter (i.e., 4000)
Words 10-15	Dummy
Word 16	Discrete bit address used to enable calculation (-1 if unused)
Word 17	Discrete state to enable calculation (0 or 1)
Words 18-19	Offset value to be added to the result of the flow calculation (real value)

The equation used for "LIQUID" flows is as follows:

let FFLOW = Value in Words 6 and 7.

OFFSET = Value in Words 18 and 19.

MVOFF = Value in Word 8.

MVSPAN = Value in Word 9.

FTRAN = Flow transmitter output (millivolts) (Word 4 contains slot number).

then

$$\text{FLOW} = \text{FFLOW} \sqrt{(\text{FTRAN} - \text{MVOFF})/\text{MVSPAN}} + \text{OFFSET}$$

Example:

FLOW	3223	FI-104	; COOL H ₂ O FROM SHELL TO HEXCH	HEAD = 171.7"
2	4	88	; FLOW TRANSMITTER SLOT	ORIFICE = 2.0"
0	5	1	; LIQUID FLOW	LINE = 3" SCH40 (3.068" I.D.)
1	3	200.	; FULL-SCALE FLOW (GPM)	
0	8	1000	; OFFSET MILLIVOLTS	
0	9	4000	; SPAN MILLIVOLTS	
0	16	-1	; DISCRETE ADDRESS NOT USED	
1	9	0.0	; FLOW OFFSET CORRECTION	
-1	-1	-1		

Saturated Steam with Pressure Correction

The pressure corrected saturated steam looks for a converted pressure in word 3.

CPOST Layout for Pressure Corrected Steam

Words 0-1	"FLOW"
Word 2	Unused
Word 3	Slot number of converted pressure (PSIG) (-1 if unused)
Word 4	Slot number of raw flow transmitter output (MV)
Word 5	Type = 2
Words 6-7	Full-scale flow at design conditions (real value)
Word 8	MV offset voltage of transmitter (i.e., 1000) may be changed at run time
Word 9	MV range of transmitter (i.e., 4000)
Words 10-11	Unused
Words 12-13	Design pressure (psig)
Words 14-15	Unused
Word 16	Discrete bit address used to enable calculation (-1 if unused)
Word 17	Discrete state to enable calculation (0 or 1)
Words 18-19	Offset value to be added to the result of the flow calculation (real value)

The equation used for saturated steam with pressure correction is as follows:

let FFLOW, OFFSET, MVOFF, MVSPAN, FTRAN, PRESS, and DP be the same as for "GAS" flow discussed earlier

then

$$\text{FLOW} = \text{FFLOW} \sqrt{\frac{(\text{FTRAN} - \text{MVOFF})}{\text{MVSPAN}} \frac{\text{PRESS} + 14.4}{\text{DP} + 14.7}} + \text{OFFSET}$$

Example:

FLOW	3391	FI-38	;STEAM TO SUPERHEATER	SATURATED STEAM
2	3	307	;PRESSURE SLOT	NOMINAL PRESS = 350 PSIG
2	4	308	;RAW FLOW TRANSMITTER SLOT	HEAD = 173.57", ORIFICE = 1.317", LINE = 2" SCH80 (1.939" I.D.)
0	5	2	;SATURATED STEAM	
1	3	5000.	;FLOW MULTIPLIER	NOMINAL FULL-SCALE FLOW = 5000 LBS/HR
0	8	1000	;OFFSET MV	
0	9	4000	;SPAN MV	
1	6	350.	;DESIGN PRESSURE	
0	16	-1	;DISCRETE ADDRESS NOT USED	
1	9	0.0	;FLOW OFFSET CORRECTION	
-1	-1	-1		

Saturated Steam With Temperature Correction

Calculations for temperature corrected steam need a converted temperature slot in word 2 and a density in words 14 and 15.

CPOST Layout for Saturated Steam with Temperature Correction

Words 0-1	"FLOW"
Word 2	Slot number of converted temperature (DEGF)
Word 3	Unused
Word 4	Slot number of raw (MV) flow transmitter output (MV)
Word 5	Type = 3
Words 6-7	Full-scale flow at design conditions (real value)
Word 8	MV OFFSET voltage of transmitter (i.e., 1000) may be changed at run time
Word 9	MV range of transmitter (i.e., 4000)
Words 10-13	Unused
Words 14-15	Density of steam at nominal temperature (real value)
Word 16	Discrete bit address used to enable calculation (-1 if unused)
Word 17	Discrete state to enable calculation (0 or 1)
Words 18-19	Offset value to be added to the result of the flow calculation (real value)

The equation used for saturated steam with temperature correction is as follows:

let FFLOW, OFFSET, MVOFF, MVSPAN, FTRAN, and TEMP be the same as for "GAS" flow discussed earlier

and DENS = Density of steam at the design temperature (value in Words 14 and 15).

then

$$FLOW = FFLOW \sqrt{\frac{(FTRAN - MVOFF)}{MVSPAN} * \frac{CDENS}{DENS}} + OFFSET$$

where "CDENS" is the steam density at the current measured temperature "TEMP". See Attachment 1, section 1 for a discussion of "CDENS".

Example:

FLOW	3081	FI-8	;LOWER H.E. STEAM FLOW
2	2	600	;TEMPERATURE SLOT, NOMINAL TEMP = 350 DEGF
2	4	489	;RAW FLOW TRANSMITTER, HEAD = 367",
0	5	3	;STEAM CORRECTED WITH TEMPERATURE CALCULATION
1	3	1000.	;FULL-SCALE FLOW (LBS/HR)
0	8	1000	;OFFSET MV
0	9	4000	;SPAN MV
1	7	0.299218	;NOMINAL STEAM DENSITY
0	16	-1	;DISCRETE ADDRESS NOT USED
1	9	0.0	;FLOW OFFSET CORRECTION
-1	-1	-1	

Rotameter Flow Calculation (Linear flow signal requiring pressure and temperature correction)

- Notes:
1. The flow transmitter value is not under the square root for rotameters.
 2. The "rotameter" function can also be used for orifice flow transmitters that have integral square root extractors and thus yield a signal that is linear with flow.

CPOST Layout for Flow Rotameter

Words 0-1	"FLOW"
Word 2	Slot number of converted temperature (DEGF)
Word 3	Slot number of converted pressure (PSIG) (-1 if unused)
Word 4	Slot number of raw flow transmitter output (MV)
Word 5	Type = 4
Words 6-7	Full-scale flow at design conditions (real value)
Word 8	MV offset of transmitter (i.e., 1000)
Word 9	MV span of transmitter (i.e., 4000)
Words 10-11	Design temperature (DEGF) (real value)
Words 12-13	Design pressure (PSIG) (real value)
Words 14-15	Unused
Word 16	Discrete bit address used to enable calculation (-1 if unused)
Word 17	Discrete state to enable calculation (0 or 1)
Words 18-19	Offset value to be added to the result of the flow calculation (real value)

The equation for rotameter flows is as follows:

let FFLOW, OFFSET, MVOFF, MVSPAN, FTRAN, TEMP, PRESS, DP, and DT be the same as for "GAS" flow discussed earlier

then

$$\text{FLOW} = \text{FFLOW} \left(\frac{(\text{FTRAN} - \text{MVOFF})}{\text{MVSPAN}} \right) \sqrt{\frac{\text{PRESS} + 14.4}{\text{DP} + 14.7} \frac{\text{DT} + 460.}{\text{TEMP} + 460.}} + \text{OFFSET}$$

Example:

FLOW	3665	FI-151	;PRODUCER GAS FLOW (DYNAMIC)
2	2	376	;TEMPERATURE SLOT
2	3	3661	;PRESSURE SLOT
2	4	358	;RAW FLOW SLOT, HEAD = 15" H ₂ O, ORIFICE = 2.8792", LINE = 6" SCH40 (6.065" ID)
0	5	4	;GAS FLOW, DENSITY = 0.1256 LBS/CUFT, W = 2546.1 LB/HR
1	3	41000.	;DESIGN FULL-SCALE FLOW = 41,000 SCFH
0	8	1000	
0	9	4000	
1	5	70.	;DESIGN TEMPERATURE = 70 DEGF
1	6	15.	;DESIGN PRESSURE = 15 PSIG
0	16	-1.	;DISCRETE ADDRESS NOT USED
1	9	0.0	;FLOW OFFSET CORRECTION
-1	-1	-1	

Superheated Steam with Temperature and Pressure

Superheated steam calculations need temperature and pressure slots, a design pressure, and a correction factor in words 14 and 15.

CPOST Layout for Superheated Steam

Words 0-1	"FLOW"
Word 2	Slot number of converted temperature (DEGF)
Word 3	Slot number of converted pressure (PSIG)
Word 4	Slot number of raw flow transmitter output (MV)
Word 5	Type = 5
Words 6-7	Full-scale flow at superheat design conditions (real value)
Word 8	MV offset of transmitter (i.e., 1000)
Word 9	MV range of transmitter (i.e., 4000)
Words 10-11	Unused
Words 12-13	Design pressure (PSIG)
Words 14-15	Superheat correction factor (real value)

$$SC = \frac{\text{Head at superheated design conditions}}{\text{Head at saturated steam design conditions}}$$

Word 16	Discrete bit address used to enable calculation (-1 if unused)
Word 17	Discrete state to enable calculation (0 or 1)
Words 18-19	Offset value to be added to result of the flow calculation (real value)

The equation for superheated steam flow is as follows:

let FFLOW, OFFSET, MVOFF, MVSPAN, FTRAN, TEMP, PRESS, and DP be the same as for "GAS" flow discussed earlier

and DCORR = superheated steam/saturated steam density design ratio -- Words 14 and 15

then

$$\text{FLOW} = \frac{\text{FFLOW}}{\text{SCORR}} \sqrt{\frac{(\text{FTRAN} - \text{MVOFF})}{\text{MVSPAN}}} \frac{\text{PRESS} + 14.4}{\text{DP} + 14.7} + \text{OFFSET}$$

where "SCORR" is the superheat correction factor discussed in Attachment 1 Section 2.

Example:

FLOW	3390	FI-33	;CYCLONE TO SCRUBBER	;SATURATED/SUPERHEATED STEAM
2	2	223	;TEMPERATURE SLOT	;NOMINAL TEMP (SATURATED) =
				620 DEGF
2	3	305	;PRESSURE SLOT	;NOMINAL PRESSURE = 250 PSIG
2	4	304	;RAW FLOW TRANSMITTER	;HEAD = 18.2-24", ORIFICE =
				2.505", LINE = 4" SCH80
				(3.826" I.D.)
0	5	5	;SUPERHEATED STEAM CALCULATION	
1	3	5000.	;FLOW MULTIPLIER	;NOMINAL FULL-SCALE SAT. FLOW =
				5000 LBS/HR
0	8	1000	;OFFSET MV	
0	9	4000	;SPAN MV	
1	6	250.	;DESIGN PRESSURE	
1	7	1.31868	;SUPER/SAT HEAD RATIO	
0	16	-1	;DISCRETE ADDRESS NOT USED	
1	9	0.0	;FLOW OFFSET CORRECTION	
-1	-1	-1		

Note: In designing an orifice to measure superheated steam flow, first, size the orifice for saturated steam at the design pressure. Then use the tables to find the superheat correction factor at the superheated design temperature and pressure.

Notes on flow calculations:

1. All pressure corrections performed on flow calculations assume that the barometric pressure is 14.4 PSIA. The barometric pressure is taken on several ADACS systems, but for simplicity and reliability, a constant barometric pressure is assumed.
2. Standard conditions used by ADACS are 14.7 PSIA and 60°F. All flows are converted to these conditions.
3. Minor corrections such as orifice expansion due to temperature are ignored. Orifice expansion can be corrected by the following equation if desired:

$$\text{CORR} = 2.002 \times 10^{-5} * T + 0.9983, \text{ where } T = ^\circ\text{F}$$

The standard flow value computed by ADACS is multiplied by "CORR."

4. The steam flow calculations present some interesting computational challenges. In particular, saturated steam corrected with temperature requires an algorithm to derive steam density from temperature and superheated steam requires an algorithm to derive the "superheat correction factor" from pressure and temperature. Appendix C shows these algorithms and their accuracy.

5. ADACS will refuse to compute flow if it suspects that any of the input variables are questionable. Any of the following conditions will result in a zero flow result:
 - a. Delta-P negative.
 - b. Delta-P less than 0.25 percent full scale. (Flow less than 5 percent full scale.)
 - c. Pressure correction is negative.
 - d. Temperature less than 0°F or greater than 2200°F.
 - e. For steam -- temperature less than 220°F or greater than 620°F.

Polynomial Calculations

If a calculation can be expressed in the form:

$$\text{Result} = K_1X^3 + K_2X^2 + K_3X + K_4$$

Where K's are constants

X is the slot to be transformed

then the calculation can be accomplished through the post-processed polynomial calculation.

In the polynomial definition of CPOST, words 0 and 1 contain the word "POLY." Word 2 contains the slot number where the data value is found. Word 3 is unused and words 4 through 11 contain the real values for the constants.

CPOST Layout for Polynomial Calculations

Words 0-1	"POLY"
Word 2	Slot number to be transformed
Word 3	Unused
Words 4-5	Constant Term (K_4) (real value)
Words 6-7	First order multiplier (K_3) (real value)
Words 8-9	Second order multiplier (K_2) (real value)
Words 10-11	Third order multiplier (K_1) (real value)
Words 12-19	Unused

Example:

POLY	3073	A.CO2.1	;CO ₂ ANALYZER	INPUT VARIABLE IS PRENORMALIZED
				TO 0 - 100 PERCENT
2	2	477	;ANALYZER SLOT NO.	OUTPUT IS 0 - 20 PERCENT
1	2	8.231E-2	;CONSTANT TERM	
1	3	1.298E-1	;FIRST ORDER TERM	
1	4	6.849E-4	;SECOND ORDER TERM	
1	5	0.0	;THIRD ORDER TERM	
-1	-1	-1		

MELOY SO2 Calculation

This post processing function is used to linearize data from MELOY total sulfur analyzer. The function that is applied is:

$$\text{Result} = 10 [A * \text{LOG}_{10} (10^{-9} * 10^{\text{mV}/20} - \text{IBASE}) + B]$$

where A = Slope
 B = Intercept
 mV = Meloy Analyzer mV
 IBASE = Base Current of Analyzer (AMPS)

is needed, CPOST words 0 and 1 contain the word "SO2," and word 2 is the slot number of the raw MELOY output. Words 4 and 5 contain the real slope value for the calculation. Words 6 and 7 contain the intercept and words 10 and 11 contain the base current (of the transmitter). All other words of CPOST are unused. Note: A program named "MELOY" is used to derive the slope and intercept for a given analyzer.

CPOST Layout for MELOY SO2 Calculation

Words 0-1	"SO2"
Word 2	Slot number of the raw output of MELOY (0-100 MV)
Word 3	Unused
Words 4-5	Slope (real value)
Words 6-7	Intercept (real value)
Words 8-9	Unused
Words 10-11	Base Current (AMPS) (real value)
Words 12-19	Unused

Example:

SO2	3074	A.S02.2	;MELOY NO. 2 (CLEAN ROOM)
2	2	478	;MELOY SLOT NO.
1	2	.5883	;SLOPE
1	3	6.0878	;INTERCEPT
1	5	1.E-8	;BASE CURRENT
-1	-1	-1	

General Math Calculation

This calculation allows a general calculation, combining slot values, of the type:

$$\text{Result} = K_1 * (X_1 + X_2) * (X_3 + K_2) * (X_6 + K_4) / (X_4 + K_3) + X_5$$

where K's are constants
X's are slot values

In CPOST words 0 and 1 contain the word "MATH," words 2 through 7 and 14 and 15 contain the constant values for the calculation, and the words 8 through 13 contain the slot numbers to be used.

CPOST Layout for General Calculation

Words 0-1 "MATH"
Words 2-3 Constant K_1 (real value)
Words 4-5 Constant K_2 (real value)
Words 6-7 Constant K_3 (real value)
Word 8 Slot "X₁" (-1 if not used)
Word 9 Slot "X₂" (-1 if not used)
Note that if both "X₁" and "X₂" are undefined, ($X_1 + X_2$) will be set to 1.0
Word 10 Slot "X₃" (-1 if not used)
Word 11 Slot "X₄" (-1 if not used)
Word 12 Slot "X₅" (-1 if not used)
Word 13 Slot "X₆" (-1 if not used)
Words 14-15 Constant K_4 (real value)
Words 16-19 Unused

Example:

MATH	3083	SV-1	;SUPERFICIAL VELOCITY
1	1	3.02286E-7	;CONSTANT K_1
1	2	460.	;CONSTANT K_2
1	3	1.0	;CONSTANT K_3
1	7	1.0	;CONSTANT K_4
2	8	3076	;FLUID AIR FLOW SLOT
2	9	3080	;COAL INJECTION AIR FLOW SLOT
2	10	570	;TEMPERATURE SLOT
-1	-1	-1	

Special Math Calculation

This calculation allows a second general calculation to be done on a number of slots of the type

$$\text{Result} = [K_1 * (X_1 - X_2) + K_2 * (X_3 - X_4)] / (X_5 + K_3 * X_6)$$

where K's are constants
X's are slot values

If desired, a naperian logarithm may be taken of the result.

The CPOST layout is almost the same as that for the "MATH" calculation. Words 0 and 1 contain "MTHX." Words 2 through 7 contain the constant values, and words 5 through 13 contain the slot numbers. Word 15 contains a flag telling whether or not the naperian log is to be taken.

CPOST Layout for Special Math Calculation

Words 0-1 "MTHX"
Words 2-3 Constant K_1 (real value)
Words 4-5 Constant K_2 (real value)
Words 6-7 Constant K_3 (real value)
Word 8 Slot "X₁" (-1 if unused)
Word 9 Slot "X₂" (-1 if unused)

Word 10 Slot "X₃" (-1 if unused)
 Word 11 Slot "X₄" (-1 if unused)
 Word 12 Slot "X₅" (-1 if unused)
 Word 13 Slot "X₆" (-1 if unused)
 Word 14 Unused
 Word 15 Calculation flag
 0 → No transformation
 1 → Napierian log

Words 16-19 Unused

Example:

MTHX	3407	DUMMY2	;INTERMEDIATE VALUE FOR EFFICIENCY
1	1	1.0	;K ₁
1	2	0.0	;K ₂
1	3	-1.0	;K ₃
2	8	206	; "A"
2	9	208	; "B"
0	10	-1	;NO "C"
0	11	-1	;NO "D"
2	12	235	; "E"
2	13	207	; "F"
0	15	1	;LOG TRANSFORMATION
-1	-1	-1	

BCD Digit Assembly

This algorithm allows the synthesis of a 6-digit numeric value from the values in three slots. A scale factor and the use of a fourth slot to provide a sign for the value are also provided for.

CPOST Layout for BCD Digit Assembly

Words 0-1 "BCD"
 Word 2 Units and tens digits slot
 Word 3 100 and 1,000 digits slot
 Word 4 10,000 and 100,000 digits slot
 Word 5 Unused
 Words 6-7 Scale factor (used to place decimal point in final result) (real value)
 Word 8 Sign flag (0 → not used, 1 → slot provides sign)
 Word 9 Slot containing sign
 Words 10-19 Unused

Example:

BCD	3202	QI-110	;TV10 CYCLE COUNTER
2	2	327	;SLOT WITH 1ST AND 2ND DIGITS
2	3	328	;SLOT WITH 3RD AND 4TH DIGITS
2	4	329	;SLOT WITH 5TH AND 6TH DIGITS
1	3	1.0	;SCALING FACTOR
0	8	0	;NO SIGN BIT
-1	-1	-1	

Integrator Function

This function allows the integration of a slot value or the product of two slot values. The integrator function is simulated using the trapezoidal rule. This function should be limited to slowly changing inputs. Parameters required in the definition for post-processing include initial data value in words 2 and 3, initial integral value in words 4 and 5, and initial time value in words 6 and 7. Also required is the word "INTG" in words 0 and 1. Word 8 contains the pointer to the first slot number, which must be defined. Word 9 contains the second slot number, which is optional, while words 10 and 11 contain the scaling factor. The expression is

$$\text{Result} = K * \int X_1 * X_2$$

where K is the scaling factor
X's are slot values

CPOST Layout for Integrator Function

Words 0-1	"INTG"	
Words 2-3	Last/initial data value (real value)	Normally initialized to Zero
Words 4-5	Last/initial integral value (real value)	Normally initialized to Zero
Words 6-7	Last/initial time (seconds) (real value)	Normally initialized to Zero
Word 8	"X ₁ " slot (must be defined)	
Word 9	"X ₂ " slot (-1 if not used)	
Words 10-11	Scaling factor K (real value)	
Words 12-19	Unused	

Example:

INTG	3393	TOT-5	; INTEGRAL OF SLURRY FLOW TO H.P. PUMP
1	1	0.0	; INITIAL VALUE OF DATA
1	2	0.0	; INITIAL VALUE OF INTEGRAL
1	3	0.0	; INITIAL VALUE OF TIME
2	8	3389	; SLOT NO. 1 (FI-5 FLOW)
2	9	290	; SLOT NO. 2 (DI-3 SLURRY DENSITY)
1	5	0.13895	; SCALING FACTOR (8.337/60.)
-1	-1	-1	

Rate Calculation

The rate calculation permits the determination of the rate of use of a material (i.e., coal, limestone, etc.). The use of this calculation requires initial values for data value (words 2 and 3), rate prediction (4 and 5), and calculation time (6 and 7). Word 8 contains the pointer to the slot number upon which the calculation is to take place. Also needed for the calculation are a time constant (words 10 and 11), a minimum calculation interval (word 12) in seconds indicating how often this calculation should be done, a maximum instantaneous rate for the calculation (word 13), and a minimum input delta (words 14 and 15).

Notes on 'RATE' calculation:

1. To minimize the effect of noise, a minimum calculation interval is defined in word 12. This value might typically be 30 seconds.

2. A second noise reduction requirement is on the input delta defined in words 14 and 15. For instance if the input signal is 0-1000 pounds a minimum input delta might be 1 pound.
3. A third noise reduction requirement is on the instantaneous slow rate. Assuming a 0-1000 pound transmitter, a typical maximum slow rate might be 500 pounds/hour. This value is defined in word 13.
4. A fourth measure that is taken to reduce noise is to apply a first order filter to the instantaneous rate values. The time constant is specified in words 10 and 11.

The calculation is as follows:

```

let PTIME = current time
    LTIME = last calculation time (words 6 and 7)
    RI = instantaneous rate
    VALUE = current data value (slot specified in word 8)
    LVALUE = last data value (words 2 and 3)
    TC = time constant (words 10 and 11)
    RATE = new rate prediction
    LRATE = last rate prediction (words 4 and 5)

```

then

```
RI = -3600. (VALUE-LVALUE)/(PTIME-LTIME)
```

and

```
RATE = LRATE + (PTIME-LTIME) * (RI-LRATE)/TC
```

When all parameters are properly defined, this algorithm performs very well, especially on slow changing inputs.

CPOST Layout for Rate Calculation

Words 0-1	"RATE"	
Words 2-3	Last/initial data value (real value)	Normally initialized to Zero
Words 4-5	Last/initial rate prediction (real value)	Normally initialized to Zero
Words 6-7	Last/initial time (seconds) (real value)	Normally initialized to Zero
Word 8	Slot Number of input signal	
Word 9	Unused	
Words 10-11	Time constant (seconds) (real value)	
Word 12	Minimum calculation interval (seconds)	
Word 13	Maximum instantaneous rate	
Words 14-15	Minimum input delta (real value)	
Words 16-19	Unused	

Example:

RATE	3084	R.COAL	;COAL FEED RATE
1	1	0.0	;INITIAL DATA VALUE
1	2	0.0	;INITIAL RATE PREDICTION
1	3	0.0	;INITIAL TIME
2	8	474	;COAL WEIGHT SLOT
0	12	25	;MINIMUM CALCULATION INTERVAL (SECONDS)
0	13	500	;MAXIMUM INSTANTANEOUS RATE
1	7	0.5	;MINIMUM INPUT DELTA
1	5	1000.	;TIME CONSTANT (SECONDS)
-1	-1	-1	

Sums Calculation

This algorithm allows the simple addition of up to 5 slot values.

$$\text{Result} = (K_1 * X_1) + (K_2 * X_2) + (K_3 * X_3) + (K_4 * X_4) + (K_5 * X_5) + (K_6 * X_6)$$

where K's are constants
X's are slot values

The slot numbers are stored in CPOST words 2 through 7, and their associated constants are in words 8 through 19.

CPOST Layout for Sums Calculation

Words 0-1	"SUMS"	
Word 2	Slot "X ₁ " (-1 if not used)	
Word 3	Slot "X ₂ " (-1 if not used)	
Word 4	Slot "X ₃ " (-1 if not used)	
Word 5	Slot "X ₄ " (-1 if not used)	
Word 6	Slot "X ₅ " (-1 if not used)	
Word 7	Slot "X ₆ " (-1 if not used)	
Words 8-9	Constant K ₁	(Constants used only if associated
Words 10-11	Constant K ₂	slot used, then must be defined.)
Words 12-13	Constant K ₃	
Words 14-15	Constant K ₄	
Words 16-17	Constant K ₅	
Words 18-19	Constant K ₆	

Example:

SUMS	3508	FTOT-1	;LEG 1 TOTAL AIR FLOW
2	2	3500	;1ST SLOT
2	3	3501	;2ND SLOT
2	4	3502	;3RD SLOT
2	5	3503	;4TH SLOT
0	6	-1	;5TH SLOT NOT USED
0	7	-1	;6TH SLOT NOT USED
1	4	1.0	;1ST CONSTANT
1	5	1.0	;2ND CONSTANT
1	6	1.0	;3RD CONSTANT
1	7	1.0	;4TH CONSTANT
-1	-1	-1	

Enable Function

Usage -- This function allows a raw slot that contains time multiplexed data to be decoded and directed to any number of post processed slots. The decoding would be normally enabled when a discrete input is in some specified state or a slot (usually with a switch threshold) is greater than or less than some specified threshold. Up to three simultaneous conditions may be specified to enable decoding. If conditions are not met, post processed output will hold the last decoded value.

"CPOST" Layout

Words 0-1	"ENBL"
Word 2	Raw slot number (contains time multiplexed data)
Word 3	1st flag, 0->unused, 1->slot enable, 2->discrete address enable
Word 4	Slot number or discrete address
Word 5	Function for slots (0->less than, 1->greater than), state for discretes
Words 6-7	Threshold for slots, unused for discrete
Word 8	2nd flag
Word 9	Slot number or discrete address
Word 10	Function or state
Words 11-12	Threshold for slots, unused for discrete
Word 13	3rd flag
Word 14	Slot number or discrete address
Word 15	Function or state
Words 16-17	Threshold for slots, unused for discrete
Words 18-19	Holds last decoded value

Example:

ENBL	3900	AI-304	;ANALYZER 4 ON UNIT 3
2	3	403	;RAW SLOT (CONTAINS TIME MULTIPLEXED DATA)
0	3	1	;SLOT ENABLE
2	4	410	;SLOT USED TO ENABLE (SWITCH THRESHOLD)
0	5	1	;TEST IS SLOT 410 > 0.0
1	3	0.0	;VALUE TO TEST SLOT AGAINST
0	8	2	;DISCRETE ADDRESS ENABLE
4	9	0	
010063K			;DISCRETE TO INDICATE ANALYZER ON
0	10	0	;ANALYZER IS ON IF DISCRETE = 0
0	13	0	
-1	-1	-1	

STEAM FLOW CORRECTIONS

1. Saturated steam density from temperature:

$$\text{DENS} = e^{((K_1 * T + K_2) * T + K_3) * T + K_4)}$$

where T = temperature of steam (°F)

DENS = density of the steam (lbs/ft³)

$$K_1 = 2.515 \times 10^{-5}$$

$$K_2 = -4.185 \times 10^{-5}$$

$$K_3 = 0.03245$$

$$K_4 = -8.516$$

This equation yields the density to within 0.3 percent over a temperature range of 240 to 600°F. Typical results are:

<u>Temperature</u>	<u>Density (lbs/ft³)</u>
240°F	0.06136
280°F	0.1154
320°F	0.2032
360°F	0.3381
400°F	0.5366
440°F	0.8206
480°F	1.2204
520°F	1.7826
560°F	2.5821
600°F	3.7449

2. Superheated steam correction factor:

Superheated steam flow is first computed as saturated steam flow and then the result is corrected by the "superheated correction factor." The algorithm used can be used for superheated steam ranging from 300°F to 1000°F and 0 to 500 PSIG. The equation is:

$$\text{CORR} = K_1 + (K_2 * T + K_3) * T + (K_4 * PZ + K_5 * T + K_6) * PZ$$

where

T = temperature of the steam (°F)

P = pressure of the steam

$$PZ = \sqrt{P + 14.4}$$

$$K_1 = 0.8514$$

$$K_2 = -1.51 \times 10^{-7}$$

$$K_3 = 7.774 \times 10^{-4}$$

$$K_4 = 1.01 \times 10^{-4}$$

$$K_5 = 3.998 \times 10^{-6}$$

$$K_6 = -1.136 \times 10^{-2}$$

CORR = "superheat correction factor"

If the equation yields a value less than 1.0, then the steam is not superheated and no correction is applied, otherwise the saturated steam flow value is divided by "CORR."

Typical values of "CORR" over its full range of input values is shown below:

	300°F	400°F	500°F	600°F	700°F	800°F	900°F	1000°F
0 PSIG	1.034	1.103	1.168	1.231	1.291	1.347	1.401	1.451
100 PSIG	.974	1.045	1.114	1.179	1.242	1.301	1.357	1.411
200 PSIG	.944	1.017	1.087	1.154	1.218	1.279	1.337	1.392
300 PSIG	.923	.997	1.068	1.136	1.202	1.264	1.323	1.379
400 PSIG	.906	.981	1.054	1.123	1.189	1.252	1.313	1.370
500 PSIG	.893	.969	1.042	1.112	1.179	1.244	1.305	1.363

APPENDIX E
Command Summary

MORGANTOWN ENERGY TECHNOLOGY CENTER
MORGANTOWN, WEST VIRGINIA

DATE:

2/15/80

IR No. 661

ORIGINATOR: *DRG*

David R. Goff

RECIPIENT:

Distribution

PROJECT:

Instrumentation Branch (MQ)

TITLE:

ADACS Command Line Manual

INFORMATION RELEASE/REQUEST:

Introduction

The METC Automatic Data Acquisition and Control System (ADACS) is an evolutionary computerized data acquisition system that serves most major METC research projects. It is a system capable of scanning up to eight independent projects simultaneously and supporting a host of graphic display terminals.

One of the more unique features of ADACS is its ability to analyze English-like commands from any terminal and act on the request. This facility is accomplished through an extensive communications interpreter.

The communications interpreter is designed to cope with a real world environment. It accepts free format input and will tolerate a great deal of misspelling and abbreviation. However, as with any language, some structure is required. This document describes in detail the commands currently accepted by ADACS along with the consequences of each command.

Discussion

This document describes in some detail the various commands available on the ADACS (Automatic Data Acquisition & Control System). The usage, format, and consequence of each command is given. The command interpreter built into the ADACS system has a great deal of capability. It allows an operator to monitor the system, modify data flow to output peripherals (line printer, CRTS, disk, and magnetic tape), modify the rate at which data is transferred to the peripherals, and also change the type of analysis applied to the data. It allows up to eight projects to be monitored simultaneously and independently. Commands can be accepted from either the master CRT (located in the ADACS Control Center) or from one of the project CRTS (located in the project control rooms). There are restrictions on the commands from the project CRTS so that each operator can only enter commands pertaining to his project.

DISTRIBUTION:

As shown on attached list.

Definitions

In the descriptions that follow, the following conventions will be used:

1. Braces [] around an item indicate that a variable input is to be entered. In most cases, the nature of the variable will be described immediately afterwards. However, some of the more common variables will only be described once.
2. Parenthesis () around a character or group of characters indicate that the enclosed characters or variables are optional and may be omitted.
3. An asterisk * before a command description indicates that the command may only be executed from the master CRT.
4. Double quotes "" will be used around the actual user input.
5. The carrot sign ^ will be used to indicate that the control key should be depressed at the same time the following character is entered.
6. A pound sign # will be used to represent the return key.
7. An ellipsis ... indicates the preceding structure may be repeated.
8. A slash / separating two words indicates that either word may be used.

ADACS Command Syntax

I) Overall ADACS System Control.

*A. Start the ADACS Program (Command entered from RDOS).

This command starts the program running. The program will ask several questions which must be answered before any further commands can be entered. The ADACS program can only be executed from the DPO directory.

*1. B4 Eclipse System	"ADACS4#"
*2. B15 Eclipse System	"ADACS15#"
*3. B1 Nova 3 System	"ADACS1#"

*B. Stop the ADACS Program. "SHUT DOWN SYSTEM/ADACS#"

This command terminates the ADACS program. It will terminate all of the projects gracefully, put an end-of-file mark on the magnetic tape and return control to RDOS. To remove the data from the tape unit, first type RELEASE MT0 or RELEASE MT1, then rewind the tape. This command will abort if any sequencers are running. If the 'SHUT DOWN' command can not terminate all of the projects and output tasks, it will type back a message indicating that the shut down was unsuccessful. The operator should then use the status commands at this point. (See IX.) Keyboard interrupts will be enabled if the shut down fails.

The ^A or ^C interrupts can be used as a last resort to terminate the program. See command 2B ("STOP [PROJECT]#") for a discussion of the users acquisition control files.

II) Project Control.

A. Start Scanning Data from a Given Project.

"START [PROJECT] SCANNING ([ACF])#"

[PROJECT] may take on a number of possibilities.

First it may be of the form: "USER [IUSER]" where [IUSER] is the user number ranging from 0 to 7. A typical command would be; "START USER 0 SCANNING#"

The [PROJECT] variable can also be the project name such as: FBC1, FBC2, Producer, VSTU, VSLTU, etc. As other projects are added to ADACS, new names corresponding to the projects will be added to the list. Note: If this command is entered from one of the project CRT's, the project name may be omitted. The command would simply be: "START SCANNING#". However, when the short form of the start command is used, the optional [ACF] can not be specified.

The [ACF] variable is optional. It is used to supply an optional acquisition control file name. Normally the default name of 'ACFX.AQ' is used. The ability to specify an optional name increases versatility by allowing several operating configurations to be predefined for a given project.

B. Stop Scanning Data from a Given Project.

"STOP [PROJECT] SCANNING#"

This will terminate scanning data from a given project. It will not terminate the ADACS program or affect other projects in any way. When the project is terminated, the current acquisition control file being used by the project is output to a disk file named ACFX.NF, where X is the project number. (0 through 7.) This file will contain changes made while the project was active. The following command actions will be preserved.

1. "SCHEDULE" (COMMAND III.)
2. "ADD" AND "REMOVE" COMMANDS (COMMAND IV)
3. "UPDATE" COMMANDS (COMMAND VI.B)
4. "MODIFY" COMMANDS (COMMAND XIII)

If you wish to use the modified acquisition control file, you must either rename the file from RDOS or start the project with the file name, ACFX.NF.

i.e. "START [PROJECT] SCANNING ACF1.NF#" or
"RESTART [PROJECT] SCANNING#"

If the project is restarted using any name other than ACFX.NF, the information in the ACFX.NF file will be destroyed. Note: If this command is entered from one of the project CRT's, the project name be omitted. The command would simply be: "STOP SCANNING#"

C. To Restart a Project after a Short Shutdown Interval.

"RESTART [PROJECT] SCANNING#"

This is used to restart a project after it has been shut down for a short period of time. This command will restart the project using the ACFX.NF control file. It will also attempt to save as much of the recall file as is possible. If the ACFX.NF file does not exist and the recall file can not be reused, this command reverts to a "START [PROJECT] SCANNING#" command.

III) Data Flow Rates to Peripherals.

- Notes: 1. This group of command requires that the project is being scanned (see 2A).
2. In setting the output intervals to peripherals, the burden on the system can be minimized if scheduling times are identical for several peripherals or exact multiples of each other. For example:

PLOT=30 SEC, LPT=300 SEC, TAPE=30 SEC.
is more efficient than PLOT=25 SEC, LPT=300 SEC, TAPE=35 SEC.

likewise, PLOT=30 SEC, LPT=300 SEC, TAPE=90 SEC.
is more efficient than PLOT=35 SEC, LPT=350 SEC, TAPE=95 SEC.

In fact, the second case would require 64 percent more computer time.

3. When a "SCHEDULE" command is issued, an output will be sent to the peripheral immediately and then will be sent when the scan counter equals a multiple of the scheduling interval. For example, the tape scheduling interval was 30 seconds. A "SCHEDULE" command was typed into the system when the users scan counter was 48 seconds. It requested the scheduling interval to be changed to 70 seconds. The tape would receive output at the following times:

TIME=0, 30, 48, 70, 140, 210, 280, ...

4. When these commands are entered from the project CRT's, the project name may be omitted. A typical command would be: "SCHEDULE CHANGE PLOT 5#"

A. Line Printer.

"SCH(EDULE) [PROJECT] LPT [INT]#"

The variable [INT] is the update interval at which a data report is to be output to the line printer for the specified project. The interval must be greater than 300 seconds and less than 32,000 seconds.

B. CRT Flowchart Output Mode.

"SCH(EDULE) [PROJECT] FLOWCHART [INT]#"

The interval must be greater than or equal to 10 seconds.

C. CRT Real Time Plot Output Mode.

"SCH(EDULE) [PROJECT] PLOT [INT] ([TSCALE])#"

The update interval [INT] can be set as low as 1 second, however, a setting of 1 second may overburden the system if a large number of users are running simultaneously. This command sets the output interval for the real time plot mode.

The optional [TSCALE] variable is used to specify the time scale for the horizontal axis of the real time plot. It may be:

SECOND(S)
MINUTE(S)
HOUR(S)

If [TSCALE] is omitted, it is assumed to be seconds.

D. Magnetic Tape.

"SCH(EDULE) [PROJECT] TAPE [INT]#"

The update interval may range from 1 second to 32,000 seconds. If output to magnetic tape is not desired, [INT] may be set to -1.

F. Data Limit Checking Interval.

"SCH(EDULE) [PROJECT] LIMIT [INT]#"

The variable [INT] specifies how often a validity check is to be made on the incoming data. It should normally be set at some value greater than 20 seconds.

G. Disk File Output Interval.

"SCH(EDULE) [PROJECT] DISK [INT]#"

The user disk file contains a number previous scans taken at the rate specified by this command. Disk file capacity varies from site to site because of hardware differences. The B4 system has the largest capacity. Disk scheduling cannot be changed while the disk file is active (see XVI).

H. Scheduling Status.

"SCH(EDULE) [PROJECT] STATUS#"

This command simply displays the current scheduling intervals for a given project.

IV) Changing the Data Values sent to Peripherals.

Notes: 1. This group of commands requires that the project is being scanned (see 2A).

2. When these commands are entered from the project CRT's, the project name may be omitted. A typical command would be: "ADD PLOT SLOTS 8 9 140#"

A. CRT Real Time Plot Output Mode.

"ADD [PROJECT] PLOT SLOT(S) [ISLOT] ([ISLOT]) ...#"

This command will first remove all previous slots from the plot output list and then add the specified slots. Any time a slot is added or removed from the CRT plot output list, the CRT will be cleared and a new plot will be started. Up to eight variables may be plotted simultaneously.

B. Removal from Real Time Plot.

"REMOVE [PROJECT] PLOT SLOT(S) [ISLOT]#"

C. Adding a Slot to a Flowchart Display.

"ADD [PROJECT] FLOWCHART SLOT [ISLOT] X=[XVAL] Y=[YVAL]#"

This command allows a slot to be added temporarily to a flowchart display. It is necessary to specify an X and Y coordinate for the displayed slot. X coordinates range from 0 to 1023 and Y ranges from 0 to 767.

D. Removing a Slot from a Flowchart Display.

"REMOVE [PROJECT] FLOWCHART SLOT [ISLOT]#"

V) Forcing Output to a Peripheral (Overrides Normal Scheduling).

Notes: 1. This group of commands requires that the project is being scanned (see 2A).

2. When these commands are entered from project CRT's, the project name may be omitted. A typical command would be:

A. Line Printer.

"OUTPUT [PROJECT] LPT#"

This will produce a users line printer report without affecting normal scheduling.

B. Magnetic Tape Data Record.

"OUTPUT [PROJECT] TAPE#"

This forces a data record to be written on tape.

VI) User CRT Output Control.

Notes: 1. All 'CRTMODE' commands refer to terminals, not projects. Therefore, the project name is never used.

A. CRT Mode Control.

The user CRTS have a number of operating modes. These modes can be set by the following commands.

1. Place CRT in Real Time Plot Mode.

"CRTMODE PLOT#"

Upon receiving this command, ADACS will clear the user CRT and draw a graph on the screen. It will then plot the slots on its output list at the update interval previously set (see 3C, 6B).

2. Place CRT in the Flowchart Mode.

"CRTMODE FLOWCHART#"

ADACS will place the flowchart display on the CRT. The locations for the displayed slots must be predefined.

3. To Load a Flowchart Definition File.

"LOAD [PROJECT] FLOWCHART [IFIL]#"

This command allows a user to load any one of a number of predefined flowchart layouts. The layouts must be generated and compiled before ADACS is started. The [IFIL] variable is the overlay number to be loaded. It may range from 0 to 25. The corresponding overlay file must have been previously defined and compiled.

4. Produce a Hard Copy of Plots.

"CRTMODE AUTOCOPY#"

This mode will issue a copy command when a plot is completed. It will not produce hard copy of flowchart displays. Be certain the hard copy unit is properly connected and turned on.

5. Terminate Hard Copy Output.

"CRTMODE NOCOPY#"

Complement of Command 4A.4.

6. To Gain Control of a User CRT.

"^C#"

This puts the user CRT in the control mode. It will then be possible to enter commands on the user CRT pertaining to that project. This command will work regardless of whether the project in question is being scanned or not. This command will only work on user CRT's.

7. To Relinquish Control of a User CRT.

"^P#"

This returns the CRT to its normal mode of operation. The CRT plot or CRT flowchart mode will resume operating. This command will only work on user CRT's.

8. To Inhibit Updates to a User CRT.

"^S#"

This is a command that is entered on the user CRT whenever it is desired to preserve a CRT display for some time.

9. To Resume Updates to a User CRT.

"^Q#"

Complement of command 9A.8.

10. To Identify Displayed Slots.

"^I#"

When the CRT is in the flowchart or analyze mode, the display only shows the current data values for the various slots. At times, the operator may want to obtain the slot number and names of the slots. This command will temporarily place the

CRT in and "identify" mode for one update. The slot numbers and slot names will appear on the screen in place of the data values. After one update interval, the display will revert to its normal mode.

B. Modification of Real Time Plot Axes and Labels.

Note: These commands are not associated with project, they are associated with terminals. Each operator can only change the parameters associated with his terminal.

1. To Change the Y-Axis Lower Limit.

"UPDATE YLL [VAL]#"

The variable [VAL] is the new integer or real value that is to be used as the new lower limit for the real time plot. It should be less than the upper limit (see 6B.2).

2. To Change the Y-Axis Upper Limit.

"UPDATE YUL [VAL]#"

Use an upper limit value greater than the lower limit (see 6B.1).

3. To Change the Y-Axis Label.

"UPDATE LABEL [SYM]#"

The variable [SYM] is up to eight characters to be used to label the Y-axis of the plot. If no label is desired, enter several spaces after the label.

4. Note: Several of the commands can be stacked into a single command if desired. For example:

"UPDATE YLL [VAL] YUL [VAL] LABEL [SYM]#"

The order and number of variables referenced is completely free format.

5. To Change the X-Axis Lower Limit.

"UPDATE XLL [VAL]#"

This is used for the manual plotting mode only.

6. To Change the X-Axis Upper Limit. (Section XVIII).

"UPDATE XUL [VAL]#"

- C. To Place a CRT in the Multiple Y-Axis Plot Mode.

"CRTMODE MULTIPLE (PLOT)#"

This command allows signals of widely differing ranges to be plotted together. The Y-axis will show the data values on a 0-100 percent scale.

- D. To Return a CRT to the Normal Y-Axis Mode.

"CRTMODE NORMAL (PLOT)#"

This command cancels the 'multiple plot' mode.

- *E. To Associate a CRT with a Project.

ATTACH QTY/CRT [IQTY] USER [IUSER]#"

This command allows any CRT to be associated with any project. The command will only be accepted from the master terminal. The [IQTY] variable specifies which terminal is being referred to. If it is a number from 0 to 7, then QTY 0 to 7 is being referred to. If it is -1 then the foreground or background terminal is being referred to. The [IUSER] variable specifies which user is being referred to. If it ranges from 0 to 7, then users 0 to 7 are being referred to. If it is -1, then the master CRT is being defined. Any number of CRT's can be attached to a project and any CRT can be the master CRT.

- *F. To Remove a QTY/CRT from ADACS.

"DETACH QTY/CRT [IQTY]#"

This command removes a QTY from the ADACS program.

- G. To Find the Status of System CRT's.

"STATUS CRT/QTY/TERMINAL#"

This command shows which CRT's are attached to which projects.

VII) Online Definition of Shortened ADACS Commands (MACROS).

- A. To Define a MACRO/X.

"DEFINE MACRO/X [IMAC] [COMMAND]#"

This command allows a shortened form of any ADACS command to be defined online. The word 'MACRO' can be replaced with 'X' to save typing. The [IMAC] variable is a number ranging from 0 to 99. MACRO's are associated with terminals, not projects. If no command is specified, the macro [IMAC] will be deleted if it exists.

- B. To Execute a MACRO.

"MACRO/X [IMAC]#"

This command calls up a previously defined MACRO and executes it.

- C. To List All Defined MACRO's.

"LIST/DUMP MACRO/X (LPT)#" .

This command lists all defined MACRO's on the CRT along with their creation date.

VIII) Request to Generate Printed Reports.

- *A. Print of the System Control Files.

"PRINT/TYPE SYSTEM/ADACS FILES#" .

This may be requested from ADACS at any time regardless of the state of the system.

- *B. Print a Copy of the User Control File.

"PRINT/TYPE [PROJECT] CONTROL FILES#" .

This command requires that data is being scanned from the project in question.

- *C. To Release the Line Printer.

"MOVE LPT#" .

This command causes ADACS to relinquish control of the line printer. ADACS then stores all subsequent line printer output in a file named "DPO:LPT".

- *D. To Print any RDOS Source File.

"PRINT/TYPE RDOS FILE [FILE NAME]#" .

This allows any source file to be printed without disturbing data acquisition. The [FILE NAME] must have the proper directory extensions prefixing it.

IX) Status Commands.

- A. To Find Out Which Projects are Running.

"STATUS USER(S)#" .

This command will print a list of the users that are active. It will also give the status of the user CRT, and give the run time

for each user. It will also give the extended memory pointers for each user.

B. Magnetic Tape Status.

"STATUS TAPE#"

This command will indicate which tape units are in service and which are in the standby modes. It will also give the serial numbers of tapes being used and indicate if automatic tape switching is enabled. The command also prints an estimate of how much tape has been used.

C. System Task Status.

"STATUS TASK(S)#"

This is primarily of interest to system programmers. It should be used any time ADACS is suspected of operating improperly. Be certain to make a record of the output.

D. Examine Peripheral Interlocks.

"STATUS LOCK(S)#"

Same comment as 9C.

E. Examine Dynamic Buffer Usage.

"STATUS BUFFER#"

Same comment as 9C.

F. Examine System Performance Counters.

"DUMP SYSCNT#"

Same comment as 9C.

G. Dump Current Alarm Conditions.

"DUMP ALARMS#"

This command will produce a list of all slots that exceed the limit or alarm boundaries. If this command is entered from the master CRT, all project alarms will be listed. If it is entered from a user CRT, only the alarms pertaining to that user will be listed. It will also list mag tape errors.

H. To Find Out Which Sequencers Are Active.

"STATUS SEQ(UENCER)#"

This command lists which sequencers are currently loaded and active. It also shows the sequencer control words, the 16 sequencer parameters and the 16 accumulators.

- I. To Examine The State of a Project.

"A(NALYZE) [PROJECT] ([OVERLAY])#"

This command displays a graphic representation of the project. It can draw vessels, display numeric values, and graphically show the state of valves. It requires a previously defined file to operate. It will draw the graphic for the project corresponding to the CRT. The [OVERLAY] variable is a number from 0 to 25.

- J. To Examine the ADACS Clock.

"STATUS TIME#"

This command prints out the current time and date of the ADACS Clock.

- K. To Examine all Gas Chromatograph Outputs.

"DUMP CHROM#"

This command lists the current status and gas analysis of all serial output gas chromatographs.

- L. Examine Producer Level Detector Outputs.

"DUMP LEVDET#"

This command lists the log of the normalized level detector outputs.

- M. To Examine the Dynamic Buffer Allocation Pointers.

This command is only of the interest to system programmers.

"DUMP ALLOC#"

- N. To Examine the Use Command Array.

It is only of interest to system programmers.

"DUMP COMMAND"

- O. 1. To Produce a Printout of the Producer Ultrasonic Detectors.

"ULTRA(SONIC)#"

2. To Examine the Calibration Factors for the Ultrasonic Detectors.

"ULTRA(SONIC) STATUS#"

3. To Change an Offset Value for an Ultrasonic Channel.

"ULTRA(SONIC) UPDATE [ICHAN] [OFFFSET]#"

- P. To Examine the Capacity of the ADACS System.

"STATUS SYSTEM SIZE#"

- Q. To Examine 'LOG' & 'DATA' Statement Usage.

"STATUS LOG#"

- X) Maintenance Aids.

- *A. To Set the Calibration Supply Voltage.

"SETCAL [G]#"

The variable [G] is the gain code (0 through 11). The resulting voltage equals 2** (13-G) millivolts. Thus G=0 Yields 8192 millivolts
G=1 Yields 4096 millivolts
and so on.

- B. Display the Value of a Slot.

"D(ISPLAY) S(LOT[S]) [ISLOT] ([NVALU])#"

The variable [ISLOT] is the first slot to be displayed. The optional variable [NVALU] gives the number of consecutive slots to be displayed. It can range from 1 to 20. If [NVALU] is omitted, only the slot specified is displayed. The calibration correction factors are applied to all displayed values. It is not necessary that any user be scanning the slots in question, but it is necessary that the MUX boxes containing the requested slots are being scanned by the system. If the slot is being scanned by a user, the slot will be displayed in engineering units. Otherwise, the slot will be displayed in millivolts. If the slot is a post-processed slot, only one slot may be displayed.

- C. Display a Slot of Group of Slots With a Function Applied.

"D(ISPLAY) S(LOT(S)) [ISLOT] ([NVALU]) FUNCTION [IFUNC]#"

In this case, [NVALU] is also optional. The variable [IFUNC] is the function pointer. It can range from -5 to 50. A value of 0 indicates no function. The negative values indicate nonlinear functions while positive values indicate linear functions.

- D. Display the Raw Uncalibrated Signal on a Slot.

"D(ISPLAY) RAW [ISLOT] ([NVALU])#"

This command yields the raw MUXADC count received from a given slot or slots and also gives the raw uncalibrated voltage corresponding to the MUX counts. This is useful when it is suspected that some of the Datum equipment may not be functioning properly.

E. Display a Slot in Metric Units.

"D(ISPLAY) METRIC [ISLOT] ([NVALU])#" .

This command will convert a slot to metric units if such a conversion is possible. If the units cannot be changed, this command will produce the same results as a 'DISPLAY SLOT' command.

F. List the Slots Being Output to a Users Peripherals.

1. Line Printer.

"DUMP SLOTS [PROJECT] LPT#"

2. Flowchart.

"DUMP SLOTS [PROJECT] FLOWCHART#"

3. CRT Plot.

"DUMP SLOTS [PROJECT] PLOT#"

4. Magnetic Tape.

"DUMP SLOTS [PROJECT] TAPE#"

5. Disk File.

"DUMP SLOTS [PROJECT] DISK#"

G. 1. Perform a Millivolt Calibration.

"LCAL#"

2. To Examine the Status of the LCAL Feature.

"STATUS LCAL#"

*3. To Change the Scheduling of the Automatic LCAL.

"SCH(EDULE) LCAL [LCINT]#" .

The [LCINT] variable determines the interval at which LCALS occur. It is normally set for 1,200 seconds. The minimum allowable interval is 600 seconds and the maximum is 27,000 seconds.

This will cause a calibration of all analog channels to be performed. The normal 20 minute scheduling of LCALS will be overridden and the next scheduled LCAL will take place 20 minutes after the manually requested LCAL.

- *H. Print a Report of the Millivolt Calibration Values.

"ACTIVATE LCAL PRINT#"

This will produce a report on the line printer giving the calibration values for all multiplexor boxes and gain settings. The report will follow each LCAL.

- *I. To Turn off the LCAL Report.

"SUSPEND LCAL PRINT#"

Complement of Command X.H.

- J. To Identify a Slot.

"IDENTIFY SLOT [ISLOT]#"

This command will print all of the information available for the specified slot. It will tell which project scans the slot, the name of the slot, the function being applied to the slot, the limits being used, and so on. If the slot is not being scanned by any of the users, the command will fail.

- K. To Identify a Slot Name.

"IDENTIFY NAME [INAME]#"

This will provide the same information as Command X.J except the search will be keyed on the name of the slot. The [INAME] variable is the ASCII name of the slot. It must be eight characters or less. The name must match exactly, i.e., "IDENTIFY NAME TIR-21#".

- L. To Examine the Current DAC Outputs.

"DUMP DAC#"

This will display the values being output to the active DACS at the instant the request is issued. It will also give the voltage that the outputs should correspond to. Note that the output of the DAC is not actually scanned. The voltage given is a computed value. The actual DAC outputs can be measured with a voltmeter.

- M. To Examine the Current Discrete Output Values.

"DUMP DISCRETE OUTPUT#"

This will display the current state of all discrete output registers. Note that the output is in OCTAL.

- N. To Examine the State of all Discrete Inputs.

"DUMP DISCRETE INPUT#"

All discrete input registers will be displayed in OCTAL.

- O. To Modify a DAC Output Value.

"OUTPUT DAC [IDAC]=[VAL]#"

This is used to force a DAC output to a specified voltage. The [IDAC] variable is the DAC number. It can range from 0 to 31. The [VAL] variable is the voltage to be forced on the DAC output. The voltage can range from 0.0 to 10.24 volts. Several things must be considered when using this command. First, consider what is connected to the DAC output. It is possible that considerable damage could be done to a valve or some other type of transducer if voltages are indiscriminately sent to DAC outputs; second, it should be noted that the value you send to a DAC may be overridden by a value sent elsewhere in the program. Use caution and ask before using if there is any doubt.

- P. To Modify a Discrete Output Bit.

"OUTPUT DISCRETE [ICARD], [IREG], [IBIT]=[IVAL]"

This command is used to change the state of a discrete output bit. The [ICARD] variable is the device code of the discrete output card. The value must be followed by a 'K' to indicate an OCTAL value. The [IREG] variable is the register number to be accessed on the card. It may only assume a value of 0, 1, or 2. The [IBIT] variable is the bit number to be accessed. It must range from 0 to 15. The same cautions and considerations listed for command X.0 also apply to this command.

- Q. Examine Part of the Extended Memory Buffer.

"OUTPUT BUFFER [ADDR] [FORMAT]#"

This allows a portion of the extended memory buffer to be dumped to the CRT. The output format is determined by the [FORMAT] word.

[FORMAT]=INTEGER	:DUMP 16 INTEGER WORDS
[FORMAT]=OCTAL	:DUMP 16 OCTAL WORDS
[FORMAT]=REAL	:DUMP 8 REAL WORDS
[FORMAT]=ASCII	:DUMP 32 ASCII CHARACTERS

The [ADDR] variable specifies the integer address of the first word to be accessed.

- R. Modify Part of the Extended Memory Buffer.

Only to be used by direct order of Dave Goff

"MODIFY BUFFER [ADDR] [FORMAT] OLD [OLD VALUE] NEW [NEW VALUE]#"

This allows part of the extended memory buffer to be modified during acquisition. It is an exceptionally dangerous command if misused. The [ADDR] variable gives the address of the first word to be modified. The [FORMAT] variable indicates the type of data being changed. The [VALUE] variable is the new integer or real value to be loaded into the buffer.

S. To Monitor a User CRT.

"DUMP MONITOR ([ILINE]) (LPT)#"

This command shows the commands entered by users recently. By default, the command will show the last 30 commands processed. If more commands are to be displayed, enter the value in the [ILINE] variable. If there are not sufficient entries in the file, only what exists will be shown. The file stores a maximum of 1,000 commands. Output may be directed to the line printer.

T. 1. Calibrate the Offset of a Flow Transmitter.

"CALIBRATE [PROJECT] SLOT [ISLOT] [VAL]#"

This command is used to compensate for offset drift in flow transmitters. The [ISLOT] variable is the slot number of the post processed flow calculation. It must be greater than or equal to 3072. The [VAL] variable is the value of the offset voltage to be used as zero flow. It must be in millivolts. A perfectly calibrated transmitter would have a value of 1,000 millivolts.

2. To Reset an Integrator.

"RESET SLOT(S) [ISLOT] ([ISLOT])...#"

This command is used to reset integrator post processed slots.

U. Examine Values in the Sysfil Buffer.

"OUTPUT SYSFIL [ADDR] [FLAG]#"

This allows a portion of the sysfil buffer to be dumped to the CRT. See Section X.Q for an explanation of the variables.

V. To Modify the Sysfil Buffer.

"MODIFY SYSFIL [ADDR] [FLAG] OLD [OLD VALUE] NEW [NEW VALUE]#"

This command allows the sysfil buffer to be modified. See command X.R for an explanation of the variables.

W. To Advance the Pneumatic Scanner a Specified Number of Steps.

"OUTPUT STEP [ISTEP]#"

This command sends "ISTEP" pulses to the Pneumatic Scanner.

X. To Home The Pneumatic Scanner.

"OUTPUT HOME#"

This command sends the pneumatic scanner to the home position.

*Y. To Change the ADACS Clock.

"UPDATE DATE [HH:MM:SS]#"

This allows the operator to correct the ADACS clock.

*Z. To Change the ADACS Calendar.

"UPDATE DATE [MM/DD/YYYY]#"

This allows the operator to correct the ADACS date.

*ZA. To reset a 'HUNG' CRT.

"RESET CRT [ICRT]#"

This command resets the software flags for the specified CRT. It should only be used as a last resort if the CRT cannot be made to respond and if the 'STATUS LOCKS' command indicates the software flags are locked up. This command may crash ADACS!!!

*ZB. To set plotter character size

"SET PLOTTER SIZE [ITEK]#"

This command is used to select the size of the characters on a Tektronix 4662 Interactive Plotter that is attached to the terminal entering the command. [ITEK] may be 4006 or 4014. The 4014 value sets the smallest character size.

XI) Control of Major Features Within ADACS.

*A. Suspension of Major Features.

*1. Limit Checking on a Given Project.

"SUSPEND LIMIT CHECKING [PROJECT] ALL#"

This eliminates the validity checking of the incoming data. The project does not have to be running to enter this command.

***2. Suspend a User.**

"SUSPEND [PROJECT] [PASSWORD]#"

This command should be used with extreme caution!!! It suspends the task responsible for scanning the project. Misuse could cause the system to crash. The [PASSWORD] variable is given out on a need to know basis only. It will be changed from time to time to insure security.

***3. Suspend Output To DAC's.**

"SUSPEND DACIO [PASSWORD]#"

Inhibits output to the DAC's.

***4. Suspend the Cleanup Task (Watchdog).**

"SUSPEND CLEANUP [PASSWORD]#"

This command must be used with extreme caution!!!

***5. To Suspend the Function Conversions Applied to the Data.**

"SUSPEND FUNCTION#"

This command allows the operator to examine the raw signals being scanned by the various projects. It can only be done if data is not being logged on tape (i.e., both tapes out of service).

***6. Suspend the Pneumatic Scanner Control Task.**

"SUSPEND PNSC [PASSWORD]#"

To suspend the task that controls pneumatic scanner.

***7. Suspend the Gas Chromatograph Control Task.**

"SUSPEND GC [PASSWORD]#"

Halts the chromatograph task.

8. To Suspend Limit Checking on a Given Slot.

"SUSPEND LIMIT CHECKING SLOT [ISLOT]#"

This causes limit checking to be removed on the specified slots. Once removed, it cannot be restored until the user restarts.

***B. Activation of Major Features.**

All of the features described in Section XI.A can be reactivated by replacing 'SUSPEND' with 'ACTIVATE'.

XII) Logging of Operator Comments.

A. To Enter Any Operator Comment onto Tape.

"LOG [TEXT]#"

The variable [TEXT] is up to 75 characters of operator comments. Longer comments can be entered by using a number of log commands. The operators comments will be printed on the line printer along with the time. It will also be logged onto the magnetic tape with the time and date. This command allows events to be noted on the same tape that contains the data. ***Note: If "LOG#" is entered with no text, the computer will print a line of asterisks across the screen showing the allowable input width ***.

B. To Enter Operator Measured Data.

"DATA [ID]=[VAL] ... #"

This command is used to enter data that ADACS can not measure. For instance the weight of dust removed from a cyclone or the amount of ash removed from the bed. The [ID] must be 2 characters, the first must be a letter, the second character may be a letter or number. It must be followed by an =, which is then followed by the data value. Several data values may be entered with one statement.

C. To Examine Log Entries.

"DUMP (PAGE [IPAGE]) LOG (LPT)#"

This command allows all logging entries and data entries to be examined. The 'PAGE [IPAGE]' is entered, if there is too much information to be shown on the CRT. The 'IPAGE' variable determines how many pages back to look. The 'LPT' option directs output to the line printer instead of CRT.

D. To Examine User Data Entries.

"DUMP (PAGE [IPAGE]) DATA (LPT)#"

E. To Examine Both Log Entries and Data Entries.

"DUMP (PAGE [IPAGE]) DATA (LPT)#"

XIII) Modify User Control Files.

Note: This group of commands requires that the project is being scanned. These commands would be used if the range on a transducer was changed during a run. These changes will be saved in the temporary acquisition control file named ACFX NF. See II.B for details.

- A. To Change the Function being Applied to a Slot.

"MODIFY [PROJECT] FUNCTION SLOT [ISLOT] OLD [OFUNC] NEW [NFUNC]#"

The variable [ISLOT] is the slot number on which you desire to specify a new function. The variable [OFUNC] is the number of the function that is currently being applied to the slot. It ranges from -5 to 50. The old function is required as input to serve as a double check that the correct slot is being referenced and also to verify that the operator is aware of the function currently applied to the data. The old function must be properly specified before any change will be accepted. The variable [NFUNC] is the new function number that is to be applied to the slot. It ranges from -5 to 50. Negative values specify nonlinear functions, positive values specify linear functions, and a value of zero specifies no function is to be applied to the slot.

- B. Modify the Engineering Units Associated with a Slot.

"MODIFY [PROJECT] UNIT(S) SLOT [ISLOT] OLD [OUNIT] NEW [NUNIT]#"

The variable [OUNIT] is the engineering units pointer currently in use. It must be correctly specified before any change will be accepted. It ranges from 0 to 31. A value of zero specifies millivolts. The variable [NUNIT] specifies the new engineering units pointer that is to be used. It also ranges from 0 to 31.

- C. Modify the Data Limits Associated with a Slot.

"MODIFY [PROJECT] LIMIT SLOT [ISLOT] OLD [OLIM] NEW [NLIM]#"

The variable [OLIM] is the limits pointer currently in use. It ranges from 0 to 63. A value of zero specifies no limit checking. The variable [NLIM] is the new limits pointer. It also ranges from 0 to 63.

- D. To Reassign a Slot Number.

"MODIFY [PROJECT] SLOT [ISLOT] NEW [NEWSLOT]#" .

This command is used to assign a new slot number to a slot that is currently being scanned. It could be used if an ADACS slot is found to be bad and it is desired to move the transmitter output to another physical slot. It can be used for a real slot to real slot move or a pneumatic slot to pneumatic slot move.

XIV) Sequencer Control and Monitoring.

This group of commands allows the operator to load, start, stop, and monitor the sequencer tasks. Up to eight sequencers may be activated at any given time. These commands currently only pertain to the B15 ADACS.

A. Loading the Sequencer Control File.

```
"LOAD SEQ(UENCER) [ISEQ] [SEQFILE]#"
```

This command is used to load the sequencer control file from disk. It must be executed before a sequencer may be started. The [ISEQ] variable is the sequencer number ranging from 0 to 7. The [SEQFILE] variable is the name of the disk file containing the sequencer control information. The file name always has a SQ appended to it. The second form of the command allows the sequencer to be loaded using the project name associated with the sequencer. For the valve test project, valid names are: VSTU, VDTU, VSLTU, VHSTU.

B. Starting a Sequencer.

```
"START SEQ(UENCER) [ISEQ]#"
```

This causes the sequencer task to be activated and a previously loaded sequencer control file to be executed.

C. Stopping a Sequencer.

```
"STOP SEQ(UENCER) [ISEQ]#"
```

This command stops the sequencer specified by [ISEQ]. The sequencer can be restarted by using the 'START' command.

D. Trace Mechanism.

1. Turn on the Real Time Trace Feature.

```
"ACTIVATE TRACE SEQ(UENCER) [ISEQ]#"
```

This causes a trace of the sequencer to be dumped to the line printer as it occurs.

2. Turn off the Real Time Trace Feature.

```
"SUSPEND TRACE SEQ(UENCER) [ISEQ]#"
```

E. To Examine the State of a Sequencer.

```
"ANALYZE [ISEQ] [OVERLAY]#"
```

This command is very useful in displaying the state of a sequencer driven project. It will produce a graphic representation of the project on the requesting CRT and actually show the state of all valves on the project. This mode is separate from the normal flow-chart mode and can only be produced on request. It cannot be made to repetitively produce output.

F. Controlling the Actions of a Sequencer.

1. Pausing a Sequencer.

"CONTROL SEQ(UENCER) [ISEQ] PAUSE#"

This command will cause the sequencer to pause the every time it executes a 'PAUSE' instruction. If the sequencer does not contain a pause instruction, no action will be taken.

2. Releasing a Paused Sequencer.

"CONTROL SEQ(UENCER) [ISEQ] CONTINUE#"

This command releases a pause previously imposed. If the sequencer is not currently paused, no action will be taken.

3. Forcing a Sequencer to a Specified Order.

"CONTROL SEQ(UENCER) [ISEQ] ORDER [IORD]#"

This essentially forces the sequencer to execute an unconditional branch to the specified order #. If the order # is invalid, no action is taken. The [IORD] variable specifies the new order #. This command should be used cautiously.

4. Loading a Sequencer Parameter.

"CONTROL SEQ(UENCER) [ISEQ] PARAMETER [IPARM] [VAL]#"

This command is use to load one of the 10 run-time settable parameters in the sequencer. The [IPARM] variable ranges from 0 to 9 and specifies the parameter to be loaded. The [VAL] parameter is the value to be loaded. The value may be real or integer.

5. To Release a Sequencer from the Pause Mode Altogether.

"CONTROL SEQ(UENCER) [ISEQ] RELEASE#"

This command cancels the 'CONTROL SEQUENCER PAUSE' command. Pauses in the sequencer will be treated as no-ops.

G. To Examine Sequencer Messages.

"DUMP SEQ(UENCER) ([ILINE]) (LPT)#"

This command allows sequencer messages to be recalled. By default the last 30 messages are shown. As many as 1,000 messages may be recalled as specified by the [ILINE] variable. Output may be directed to the line printer.

XV) Discrete Output Control Commands.

This group of commands allows the operator to change the state of valves, relays, and other devices driven by discrete outputs.

A1. Asserting a Condition.

"OPEN [PROJECT] [NAME]#"

This command is primarily intended for valves. The [NAME] variable is the name of the valve to be opened (i.e., CV1 or TV1). The [NAME] must be defined in the 'VTEXT' file.

A2. "TRUE [PROJECT] [NAME]#"

This command is basically the same as the 'OPEN COMMAND' except it makes more sense for devices other than valves.

B1. Non-Asserting a Condition.

"CLOSE [PROJECT] [NAME]#"

The opposite of the 'OPEN' command.

B2. "FALSE [PROJECT] [NAME]#"

The opposite of the 'TRUE' command.

XVI) User Data Disk File Control.

A. To Find Out the Status of all User Disk Files.

"STATUS DISK#"

B. To start a disk file for a user. "DISKFILE [PROJECT] START#"

This command erases any previous disk data for the specified user and starts a new file.

C. To Terminate a User Disk File.

"DISKFILE [PROJECT] STOP#"

This command causes output to the user disk file to stop. It does not, however, erase the file. The file will remain on disk until it is deleted or until the "DISKFILE START" is issued again.

D. To Recall a Tabular List of Data Values.

"R(ECALL) [PROJECT] L(IST) (SLOTS) [ISLOT] ([ISLOT]) ...#"

This command lets the operator examine a subset of tabular values from the disk file. The number of values dumped is determined by the 'RECALL INTERVAL'. The starting opint for the dump is set in the 'SET RECALL TIME' command. In addition to the data, ADACS will also display the minimum, average, maximum, and standard deviation of the data values listed.

E. To Plot a Slot from the Disk File Data.

"R(ECALL) [PROJECT] P(LOT) (SLOTS) [ISLOT] ([ISLOT]) ...#"

This command plots data values stored on disk for the specified slots. The plot will be a high resolution type and will be labeled indicating time in the past. The limits for the Y axis are selected using the 'UPDATE' command and the X axis time interval is selected using the 'SET RECALL INTERVAL' command.

F. To Specify the Starting Time for Recall.

"SET RECALL TIME [HRS:MIN] [MO/DAY]#"

This command allows the starting time for a recall. It requires the hours and minutes, and month and day to be entered. If only the most recent data is required, enter a time after the current time; for instance, if the 'RECALL TIME' is set to 23:59 12/31, the most recent data will always be displayed. This command is associated with a CRT and not a project.

G. To Specify the Recall Interval.

"SET RECALL INTERVAL [INT] [UNITS]#"

This command allows the recall interval to be set. The [INT] variable give the time and the [UNITS] variable is 'HOURS' or 'MINUTES'. This command is associated with a CRT and not a project.

H. To Recall Statistics Over Some Interval.

"R(ECALL) [PROJECT] S(TAT) (SLOT(S)) [ISLOT] ([ISLOT]) ...#"

This command produces summary statistics over the specified interval.

I. To Set the Number of Points to Plot/List.

"SET RECALL RESOLUTION [IRES]#"

This command can modify the number of points allowed on a recall plot. [IRES] is a number ranging from 1 to 10. It sets the number of points to be (720/IRES). The normal value for [IRES] is 3.

XVII) Online Calculator Feature.

"? [CALCULATION]##"

This feature allows any mathematically correct calculation to be performed online. Valid operators are:

- + Addition.
- Subtraction.
- * Multiply.
- / Divide.
- ** Exponentiation.
- () Up to 10 levels of parentheses.

The calculation may contain any parenthesis nesting as long as it is mathematically possible. Numbers may be entered with or without decimal points. If a number is negative, it should be placed in parentheses with a leading '-'.

EXAMPLES: ? 23*56 - 13*27
 Result = 937.0000

 ? 2*(23/13+(-23)/17)
 Result = 0.832579

 ? (34*34+57*57)**0.5
 Result = 66.37017

Slot numbers may participate in the calculator feature. Slot numbers are indicated by surrounding the number with < >. The only restriction is that the slot must be scanned by a user.

EXAMPLE: ? 10*<50>**0.5
 <50> is replaced with the current value of slot 50.

It is possible to chain the result of one calculation to the next. The result of the previous calculation can be referred to with a \$ sign. Use the \$ sign like any number.

EXAMPLES: ? 5*7+3
 Result = 38.0

 ? \$/2-5
 Result = 14.0

It should be noted that very powerful online calculations can be pre-defined by using the "MACRO" feature described earlier. Calculations could be defined that involve slot numbers and could be chained together if the calculation was sufficiently complex.

XVIII) Manual Data Plotting.

This feature allows a set of data values to be entered and plotted or analyzed. The data is entered into terminal unique files and remains intact until deleted.

A. To Open a File.

```
"F(ILE) O(PEN) [INUM]#"
```

To open a file, file numbers may range from 0 to 9.

B. To Close a File.

```
"F(ILE) C(LOSE)#"
```

To close the currently open file, contents are not destroyed.

C. To List a File on the Terminal.

```
"F(ILE) L(IST)#" .
```

This lists the contents of the currently open file on the terminal.

D. To Enter a Data Value.

```
"F(ILE) E(NTER) [IENTRY] [X] [Y]#"
```

This command allows a data value to be added, deleted, or modified. The [IENTRY] variable is the key to where a data value will be inserted. The [IENTRY] values will always be in ascending order. If an entry is to be delete, omit the [X] and [Y] fields. If an entry is to be modified, use the same [IENTRY] value as the entry to be replaced. If a new entry is being made, choose a new unique [IENTRY] value.

E. To Plot the Data Set.

```
"F(ILE) P(LOT)#" .
```

To plot all data values. The "UPDATE" command should be used to set up the plot limits.

F. To Compute Statistics.

```
"F(ILE) S(TAT)#" .
```

G. To Erase a File.

```
"F(ILE) ERASE#" .
```

To delete all entries from the open file.

Special Properties of the ADACS Command Line Interpreter

In addition to being able to understand the commands described above, the ADACS Command Line Interpreter has several additional features. First, it is fast with no command requiring more than 20 seconds to execute. Secondly, it can tolerate a large variety of misspellings. The two most common typing errors are letter transpositions and accidental letter repeats. The Command Line Interpreter will compensate for both of these errors. Lastly, the Command Line Interpreter is expandable with an almost unlimited command capacity by virtue of its overlay structure.

Conclusion

The ADACS Command Line Interpreter has proved to be highly successful in field use. It is routinely used by operating and engineering personnel at METC, most of whom have no computer background. As ADACS grows, new commands will be added and old ones will change. When the changes become significant, this document will be revised.

2:br:51lef

David R. Goff
Electrical Engineer
Instrumentation Branch

APPENDIX F
Typical ADACS Dialogue

ADACS4

ADACS-REV 4.10 (B 4) STARTED AT 14:24:18 ON 2/ 3/1982

***CAN'T USE ZEBRA DISK. SWITCHING TO THE DIABLO DISK
WHICH TAPE UNIT ACTIVE? : 0 OR 1 OR N(NEITHER)

N
BOX 2 GAIN 11 ERROR = .20% OFFSET = .2649 MV 14:24:47 2/ 3/1982

READY
START PRODUCER SCANNING

PROJECT 1 STARTED AT 14:25:13 ON 2/ 3/1982 ACF = ACF1.04

*** NO TAPE MOUNTED-DATA WILL NOT BE RECORDED ***

PROJECT 1 MAG BUFFER = 1 RECORDS 796 WORDS EACH
PROJECT WILL USE 26.88 FEET OF TAPE PER HOUR

DEVICE INTERVAL	
TAPE	30.
LPT	1800.
FLOWCHART	10.
LIMIT	30.
DISK	30.
PLOT	10. SEC

OK

READY

DISPLAY SLOT 48 10
 SLOT 48 • 991.749 MU (USER 1)
 SLOT 49 • 983.634 MU (USER 1)
 SLOT 50 • 1.960 PSIG (USER 1)
 SLOT 51 • 29.501 MU (USER 1)
 SLOT 52 • 1014.429 MU (USER 1)
 SLOT 53 • -37.331 PSIG (USER 1)
 SLOT 54 • 30.813 MU (USER 1)
 SLOT 55 • 969.902 MU (USER 1)
 SLOT 56 • 1.804 PSIG (USER 1)
 SLOT 57 • 8.049 MU (USER 1)
 OK

READY
 D S 4S 10
 SLOT 48 • 990.501 MU (USER 1)
 SLOT 49 • 976.144 MU (USER 1)
 SLOT 50 • 1.335 PSIG (USER 1)
 SLOT 51 • 29.520 MU (USER 1)
 SLOT 52 • 1014.429 MU (USER 1)
 SLOT 53 • -37.175 PSIG (USER 1)
 SLOT 54 • 30.930 MU (USER 1)
 SLOT 55 • 981.972 MU (USER 1)
 SLOT 56 • 1.960 PSIG (USER 1)
 SLOT 57 • 8.010 MU (USER 1)
 OK

READY

INTROD MULTIPLE PLOT
OK

READY
SET RECALL INTERVAL 15 MINUTES
OK

READY
RECALL USER 1 PLOT 48 49 50

READY

F-5

ECALL USER 1 LIST 48 49 50 51

TIME	SLOT 48	SLOT 49	SLOT 50	SLOT 51	SLOT
14:25:27	989.90	938.07	2.27	29.50	
14:25:56	978.02	933.07	1.80	29.91	
14:26:26	968.65	933.70	1.96	29.70	
14:26:56	973.65	934.95	2.12	29.38	
14:27:26	966.76	928.03	1.49	29.27	
14:27:56	980.51	978.02	1.65	29.21	
14:28:26	989.83	968.03	1.34	29.48	
14:28:56	986.13	963.66	1.65	29.85	
14:29:26	977.39	964.91	2.12	29.23	
14:29:56	981.14	966.16	1.18	29.64	
14:30:26	966.76	963.64	1.65	29.34	
14:30:56	986.76	963.66	1.34	29.25	
14:31:26	983.01	965.53	1.18	29.49	
14:31:56	981.14	967.42	2.12	29.46	
14:32:26	986.76	968.04	1.18	29.36	
14:32:56	984.88	968.04	1.02	29.54	
14:33:26	985.51	963.66	1.49	29.68	
MINIMUM	968.653	933.074	1.023	29.208	
AVERAGE	981.578	958.739	1.620	29.485	
MAXIMUM	989.877	988.628	2.272	29.910	
STANDEV	6.238	15.538	.391	.210	
OK					

READY

ANALYZE USER 1 2

READY

BACK-END

GRAPHICS 2 14:34 2/ 3/1982

GAS-SCRUBBERS

F-9

APPENDIX G
The ADACS Routines

Routine: ADACS 4
Priority: 0
Residency: Core
Size: 14012K

ADACS 4 is the root program for ADACS (the numeral at the end is the particular system number). The main task of the root program is to establish the parameters and limits on what that system will be able to do. It also sets up arrays, the size of which are determined by the parameters being set, for the storage of data values.

The last ten percent of the code calls in the ADACS startup task, MAST.

The types of parameters that are set deal with the size of the system, the number of users, the routines needed, the number and types of engineering units, the number and types of data conversions, etc.

The parameters are stored in the common areas SYSF0 and SYSACT as follows:

SYSF0

KBOX	-	number of analog input boxes in the system
KCHAN	-	number of analog input channels in the system
DLIM	-	number of limit values in the system
KALM	-	number of alarm values in the system
KFLF	-	number of linear functions in the system
KFNF	-	number of non-linear functions in the system
KENG	-	number of engineering units in the system
KUSR	-	number of users allowed in the system
KSEQ	-	number of sequencers allowable in the system
KDAC	-	number of DAC's allowable in the system
KRANGE	-	number of ranges allowed in the system

SYSACT - 20 word array XACT

- Word 1 - sequencers allowed? - 1=yes, 0=no
- Word 2 - DAC's allowed? 1=yes, 0=no
- Word 3 - number of 1/10 second periods between DAC updates
- Word 4 - discrete I/O allowed? 1=yes, 0=no
- Word 5 - temperature monitors allowed? 1=yes, 0=no
- Word 6 - clock output available? 1=yes, 0=no
- Word 7 - watchdog feature on system? 1=yes, 0=no
- Word 8 - digital output for watchdog output
- Word 9 - maximum number of CRT's allowed in the system
- Word 10 - recall resolution - number of points on a recall plot
- Word 11 - number of special slots allowed for dedicated calculating
- Word 12 - calibration channel on each analog input box
- Word 13 - number of tape units on the system
- Word 14 - pneumatic scanners on the system? 1=yes, 0=no
- Word 15 - ADACS system number
- Word 16 - Zebra disk on the system? 1=yes, 0=no
- Word 17 - Maximum number of macros per terminal
- Word 18 - Buffer size for buffering tape output
- Word 19 - Size of postprocessing control table
- Word 20 - Maximum number of commands retained by ADACS

Routine: B4DUM
Priority: N/A
Residency: Core
Size: 14K

B4DUM is a small routine that provides dummy entry points for all of the programs that are not used by this ADACS system. This is done so that the linking program will see all of the expected programs referenced whether they are used or not.

Routine: B4MES
Priority: N/A
Residency: Overlay area 0, 11
Size: 620K

When called by ULTRA this routine prints the values of each ultrasonic detector on the line printer.

When called by LEVDET it prints an output message concerning void detection.

Routine: BUFFER
Priority: N/A
Residency: Core
Size: 363K

BUFFER sets up a number of commons and arrays that have fixed sizes and are common to all systems.

Routine: CHROM
Priority: 50
Residency: Core
Size: 523K

CHROM is the chromatograph handling routine. It is an asynchronous task that is used only with the B-4 system.

CHROM scans the outputs of the chromatographs once a second and once it has a millivolt value for each of the gases associated with the analyzer it calls GCOUT. After GCOUT converts the millivolt readings into meaningful units CHROM generates a report to the line printer and an auxiliary teletype located in the chromatograph room, and writes records to tape and disk units.

Routine: CL1
Priority: N/A
Residency: Overlay area 0, 3
Size: 770K

This routine, which is called by CLEANUP, performs some of the tasks described in CLEANUP. For example, if the system appears to be getting deadlocked it will print the messages, clear the software flags, and reinitialize the hardware.

It also has messages concerning the temperature monitors, and it has the calculations used in the floating point hardware check.

Routine: CL2
Priority: N/A
Residency: Overlay area 0, 4
Size: 666K

This routine, which is called by CLEANUP, deciphers the single-bit memory errors that occur, and keeps a history and prints out messages on them. It also is called once an hour to write the status of each system user to disk so that in the event of a system crash they can be recovered to the point they were at the last hour.

Routine: CLEANUP
Priority: 55
Residency: Core
Size: 641K

CLEANUP is the system garbage collector. It performs numerous small tasks that there is no place else to do, and it runs a lot of period diagnostics and checks on the system to make sure everything is running properly.

First, CLEANUP logs all single bit errors that occur in the system so that a failing memory board can be detected early. The Eclipse computer operates with a sixteen-bit word with five additional parity bits that are used for error detection and correction. If only one bit is in error the system will correct the error and continue as if nothing had happened. If two or more errors occur the system will usually crash.

CLEANUP checks the internal computer temperature monitors to see if any of them are above the threshold temperature, 105°F. If any of the monitors exceeds the threshold a message is printed and the threshold temperature is raised ten degrees.

A system sanity check, called a watchdog check, is performed by the system. At least every ten seconds the data acquisition tasks must set a bit in a specific memory location. CLEANUP will check those bits every fifteen or twenty seconds to make sure all of the tasks are running properly. If the bits are set the routine clears them and assumes everything is running properly. If one of the bits is not set the routine sets a flag, clears the bits, and gives it one more chance. If it is still not set, the routine will call a series of routines to try to get the data acquisition systems running again.

Next, CLEANUP resolves deadlocks between tasks that are competing for the use of common resources, such as the line printer. If it finds a piece of equipment locked up for an inordinate amount of time, it steps in and releases the equipment and lets them battle for it again. It continues this until eventually one of the tasks wins.

Finally, once an hour CLEANUP calls in a routine which checks out the floating point hardware, which performs all of the floating point calculations for the computer. The routine goes through a number of calculations designed to test each function and then compares the answer it receives from the unit with the answer it should get. If they differ it will sound an alarm.

Routine:	COLOR
Priority:	N/A
Residency:	Overlay area 2, 1
Size:	1230K

This routine is used to update the data values on the color graphics on the color CRT. It is a data table driven routine that is called by REPORT whenever an update is needed, and it updates the data values and the color status of the graphics.

Routine: CRT
Priority: 100
Residency: Core
Size: 633K

CRT is activated by MAST at startup. It determines when output is needed at one of the user's CRTs on a periodic basis. When it senses that output is required, it sets a flag that activates REPORT, which prepares the data. When REPORT is finished it sets another flag and CRT picks up the data.

Once it finishes, it clears the flags and frees the data storage area for the next request.

This one routine services all eight users on a first-come, first-served basis.

Routine: DACDR
Priority: N/A
Residency: Core
Size: 21K

This is a small assembly language routine that interfaces with the DAC output hardware. It takes the output and sends it to the hardware.

Routine: DACIO
Priority: 30
Residency: Core
Size: 137K

This routine is activated at startup by MAST. It has the responsibility of outputting data values to DAC's. It picks up the data values from an array at the required time interval and outputs it to the DAC's.

If a routine needs to output something to a DAC, all it needs to do is change the value in memory. The next time DACIO runs the voltage will be output.

Routine: DISCR
Priority: N/A
Residency: Core
Size: 132K

This is a series of assembly language routines that are used to interface with the digital I/O capabilities of the Datum hardware.

There is one routine to allow ADACS to read a digital input disk or sixteen bits simultaneously.

Another routine allows ADACS to write a single bit or sixteen bits simultaneously.

Routine: DRAWIT
Priority: N/A
Residency: Overlay area 0, 22
Size: 567K

This routine is called to create the initial background graphic on the color CRT. Once the image is established, it will remain on the screen until it is overwritten.

Routine: ENDIT
Priority: N/A
Residency: Overlay Area 0, 6
Size: 775K

This routine is called by REPORT to shut down a user's data acquisition task. It writes out shutdown messages to the line printer and tape, and shuts down everything that the task was doing.

Routine: GCOUT
Priority: N/A
Residency: Overlay area 1, 53
Size: 1520K

GCOUT takes the chromatograph millivolt readings from CHROM and compares each one to the full scale reading for that gas, converting it to a percentage. It then uses these results to compute the average molecular weight of the gas, the gross Btu content, the net Btu content, the specific gravity, and the density of the gas in grams per liter. It then returns all of these results to CHROM.

Routine: GRAPH
Priority: 60
Residency: Overlay area 1, 52
Size: 1705K

GRAPH is called by TSK 60 to clear the CRT screen, draw the background grid, label the plot and return. Before a new plot is started the routine will automatically make a copy of the previous one before it clears the screen if so specified by the user.

Routine: GTCHR
Priority: N/A
Residency: Core
Size: 153K

GTCHR classifies an input character into one of several categories:

- Class 1 delimiters - space, parenthesis, equal sign, comma, colon, slash, or tab.
- Class 2 numeric characters - 0 through 9.
- Class 3 alphabetic characters - A through Z.
- Class 4 carriage return or line feed.
- Class 6 dash/minus sign.
- Class 7 period/decimal point.
- Class 5 none of the above.

Routine: INPUT
Priority: N/A
Residency: Overlay area 0, 2
Size: 774K

This routine is used to do the initial deciphering of the commands that have been typed in. It looks at the first word of the command and selects one of about forty overlays to be called in for further processing.

INPUT first classifies the first word into one of several different categories:

-1	unrecognizable
0	primary command
1	integer number
2	valid command but not primary
3	real number
4	octal number

If the command is a primary command, INPUT looks the command up in a table and gets the number of the proper overlay to load.

Routine: LCAL
Priority: N/A
Residency: Overlay area 1, Ø
Size: 1346K

This routine counts down a time and at the end of the time interval it calibrates the analog input boxes and determines the appropriate calibration factors to ensure an accuracy of at least $\pm 0.1\%$. The routine can determine a single bad channel out of hundreds on the analog input.

The inputs are checked by applying a precise voltage to each of the channels and determining the calibration factors. Since the gains of the input channels are known, the theoretical value can be compared to the actual value. If the difference is more than one percent, it is assumed that the channel is going bad. Usually, an input channel can be repaired before it goes completely bad.

Routine: LEVDET
Priority: 120
Residency: Core
Size: 413K

This routine is dedicated totally to the producer and performs a number of functions associated with that project.

LEVDET's first task is to determine if a rupture disk blows. There are currently four rupture disks that have thermocouples associated with them. If a rupture disk blows, the thermocouple temperature will rise due to the hot gas passing over it. The slots associated with these thermocouples are scanned by LEVDET and if any of them are above a threshold temperature, currently 100°F, that rupture disk is assumed to be blown and an alarm is sounded.

LEVDET's next job is to make sure the pilot flare is lit. The pilot has three thermocouples placed sixty degrees from each other in a horizontal plane. Theoretically, no matter what direction the wind blows at least one of the thermocouples should be in the flame and, therefore, heated. LEVDET examines the slots for these thermocouples and if at least one of them is above the threshold temperature, currently set at 130°F, the flare should be lit. If all three are cold the flare is assumed to be out and an alarm is sounded.

The humidifier nozzles are controlled by LEVDET. The outlet temperature of the humidifier is compared to a set point and the nozzle outputs are adjusted to correct any difference. The nozzles are turned on sequentially, nozzle one through nozzle four, such that no nozzle can run at below fifty percent output.

Finally, LEVDET takes the outputs of the five nuclear density detectors on the producer shell and computes an estimate of the total mass in the bed. This estimate is printed on the line printer and is sent to the producer control room so that the technicians will know how constant the bed mass is staying.

LEVDET also tries to determine if a void is developing within the bed from the detector signals. If a void is detected an alarm is sounded.

Routine: LIMCHK
Priority: N/A
Residency: Overlay area 2
Size:

LIMCHK is a routine that determines if the data values are within what the project considers normal limits. If a signal is found outside of the normal bounds it will write messages to the line printer and tape, and sets a flag on the CRT display indicating that a signal is out of limits. Currently replaced by "COLOR".

Routine: LINE
Priority: N/A
Residency: Core
Size:

Given two x and y coordinates this routine will draw a line between those coordinates on a Tektronix CRT screen.

Routine: LPOUT
Priority: 75
Residency: Core
Size: 633K

This routine, which is activated at startup by MAST, is the line printer output task. It is similar to the CRT task in that it serves all eight users determining which ones require output to the line printer. If output is needed, it acquires it from REPORT, and then outputs it to the line printer.

Routine: MAST
Priority: N/A
Residency: Overlay area 2
Size: 6744K

MAST is the ADACS master task. It is called by the root program ADACS and once its calculations, questions, and functions are performed it is killed and not used again until ADACS is restarted.

This routine opens all of the required FORTRAN channels and initializes all of the appropriate I/O devices. The FORTRAN channels used are:

Channel	Use
0	tape I/O
2	overlay file
3	writes users status hourly to disk for recovery
10	console output
11	console input
12	line printer
13	auxiliary teletype output
20-27	user 0-7 crt I/O
30-37	user 0-7 recall files
40	master task temporary (to read control files)
41	command file storage
42	sequencer message storage
43	alarm storage
50-57	temporary user 0-7 restart files
62	temporary communications
63	master task and temporary communications

It is responsible for reading in all of the system control files - functions, engineering units, etc. It initializes all of the asynchronous tasks that run in ADACS and it initializes all variables in the arrays.

MAST's first task is to allocate the extended memory, which is 32K words of storage set aside for data storage. Then it rewinds the tapes and opens the console channels to the master CRT. It opens channels to the line printer and all of the user's CRTs.

Next, it prints out a startup message and the time and date. It defines the digital inputs and outputs in terms of the number of them and their addresses. It determines if a Zebra disk is associated with the system and, if there is, deletes any existing recall files.

It then asks the operator which tape unit(s) are active, how many feet of tape are on the tape that is mounted and whether or not the other tape drive is available to be switched to. It also asks if the tapes are new or continuations or old ones.

Log files and operator-entered data files are set up. The tape units are initialized and header messages written on the tapes.

Finally, the rest of the control files are read into memory, and some additional files are set up to keep track of the last 1000 sequencer messages and 1000 alarm messages.

Routine: MICRO
Priority:
Residency: No currently in use
Size:

This is a collection of assembly language service routines.

STORE is used to access a single byte out of a sixteen-bit word, either left or right.

BLOCK is used to move a block of words from one place to another.

LOCATE locates the first bit that is set to a one in a sixteen-bit word.

LOCR locates and resets the first nonzero bit in a sixteen-bit word.

COUNT is used to count the number of bits set to one in a sixteen-bit word.

Routine: MTEH 1
Priority: N/A
Residency: Overlay area 0, 0
Size: 650K

This routine, called by RMTEH, sounds an alarm that there is a problem with the tape drive. It then tries to determine whether it is a hard or soft failure.

If it is a hard failure it tries to determine if another drive is available. If not, it writes an error message and returns to RMTEH.

If a tape drive is available RMTEH calls in MTEH 2.

Routine: MTEH 2
Priority: N/A
Residency: Overlay area 0, 1
Size: 436K

MTEH 2 closes the channel on the tape that had been used and rewinds the tape. It then tries to initialize the new tape, determines if everything is okay with it, and then writes a continuation record indicating that the tape was continued from a previous tape.

Routine: MTEH 3
Priority: N/A
Residency: Overlay area 0, 1
Size: 211K

MTEH 3 writes messages to the CRTs and line printer indicating that a successful tape switch had been completed and releases the tape.

Routine: MUXID
Priority:
Residency: Core
Size: 327

This is a series of small routines which are used to interface with the analog input hardware. They define the hardware to the system and establish an interrupt clock at ten times a second. There is a small routine that gets interrupted to update that clock.

There is a small loop that counts down the interval to the next analog calibration. There are some small routines which are used by LCAL to actually perform the calibration, to establish the calibration configuration, and to set up the voltages of the calibration sources.

Routine: MUXINIT
Priority:
Residency: Core
Size: 71K

This is a collection of assembly language routines.

The first routine has an entry point called MUXANIT which is used to initialize the analog input hardware. This is used by MAST to get things going at startup, and by CLEANUP when problems have developed.

Another entry point, ERCCON, gets interrupted anytime a memory error is detected.

There is one other routine which is used to intercept system panics. A system panic occurs whenever RDOS detects some inconsistent position within its control tables; instead of allowing RDOS to shut down it is intercepted and an attempt is made at keeping data acquisition going, even at a reduced level.

Routine: PLOT
Priority: N/A
Residency: Overlay area 2, 1
Size: 461K

PLOT is called by REPORT to update the real time plots on the CRTs. It takes the data, scales it to the grid, and writes a character to the correct position on the display.

Routine: POST 4
Priority: N/A
Residency: Overlay area 2, 1
Size: 2343K

This is the post processing control routine. It is site specific but all have the same basic structure. It is called by REPORT whenever a slot is determined to be post processed.

POST 4 looks at a twenty-word control table which lists the type of processing required for a slot. The first two words of the twenty establish the type of processing that is required with a four-character ASCII mnemonic - FLOW, SO-2, etc. The other eighteen words are specific for each calculation, and they specify data for the calculation. Example:

CPOST Layout for Gas Flow (Flow corrected by pressure and temperature)

Words 0-1	"FLOW"
Word 2	Slot number of converted temperature (DEGF)
Word 3	Slot number of converted pressure (PSIG) (-1 if unused)
Word 4	Slot number of raw flow transmitter output (MV)
Word 5	Type - 0
Words 6-7	Full-scale flow at design conditions (real value)
Word 8	MV offset voltage (i.e., 1000) may be changed at run time
Word 9	MV range of transmitter (i.e., 4000)
Words 10-11	Design temperature (DEGF) (real value)
Words 12-13	Design pressure (PSIG) (real value)
Words 14-15	Unused
Word 16	Discrete bit address used to enable calculation (-1 if unused)
Word 17	Discrete state to enable calculation (0 or 1)
Words 18-19	Offset value to be added to the result of the flow calculation (real value)

The equation used for "GAS" flows is as follows:

let FFLOW = Value in words 6 and 7.
 OFFSET = Value in words 18 and 19.
 MVOFF = Value in word 8.
 MVSPAN = Value in word 9.
 FTRAN = Flow transmitter output (millivolts), word 4 contains slot number.
 PRESS = Pressure correction (PSIG) -- Word 3 contains slot number.
 TEMP = Temperature correction (°F) -- Word 2 contains slot number.
 DT = Design temperature -- Words 10 and 11.
 DP = Design pressure -- Words 12 and 13.

then

$$\text{FLOW} = \text{FFLOW} \sqrt{\frac{(\text{FTRAN} - \text{MVOFF})}{\text{MVSPAN}}} \frac{\text{PRESS} + 14.4}{\text{DP} + 14.7} \frac{\text{DT} + 460.}{\text{TEMP} + 460.} + \text{OFFSET}$$

Additional information on slot post processing is contained in METC Information Release No. 936 entitled "ADACS Slot Processing" by David R. Goff and Ralph Crane.

Routine: RCOMM
Priority: 70
Residency: Core
Size: 1745K

RCOMM, the core resident task for the communications interpreter, is created by MAST at startup. It polls the various USER tasks that are reading the CRT keyboards to see if they have a complete line of input.

If it has a complete line of input it puts it into a buffer and calls INPUT.

Routine: REDUC
Priority: N/A
Residency: Core
Size: 465

This routine converts voltage signals from the ADC's into the appropriate engineering units. There are three types of conversions handled by REDUC - linear, thermocouples (piecewise linear), and switch functions.

Routine: REPORT
Priority: 40 to 47, depending on the user ID
Residency: Overlay area 2, 1
Size: 3075K

This is the heart of ADACS. It can be activated up to eight times to perform acquisition on eight different users.

It writes startup information to the line printer and tape and then enters an infinite loop that checks a series of flag words to see where output is needed - i.e., the tape drive, the disk, the line printer, another task, etc.

If output is required it calls XFER to scan the slots for data. Then REDUC is called in to convert the data from a voltage to engineering units. Next, SPECIAL is called to perform the post processing calculations.

Next, REPORT goes through a series of checks to see where the output was required. There is a special flag to indicate if another routine needs the data, and REPORT will write it to a special area of memory, set another flag; then that routine can get the data.

Routine: RMTEH
Priority: 35
Residency: Core
Size: 74K

RMTEH is the core resident portion of the magnetic tape error handler. It, along with its support subroutines MTEH 1, MTEH 2, and MTEH 3, tries to determine what is wrong with the tape unit that is active. If it is considered a soft error, a bad data value, it will try to continue using that tape.

If it is a hard failure it will try to switch to the other tape drive.

Once it is through the task will be killed.

Routine: SCAL 1
Priority: N/A
Residency: Overlay area 1, 54
Size: 1675K

This routine performs the calculations for the special slots for USER 1 on the system, the producer. It then places the results of these calculations in an area where it can be later picked up by the actual scanning task.

There are currently 37 special calculations for the producer, some of which are extremely complex. Some of the computed quantities are humidifier efficiency, cyclone velocities, heat loss to cooling water, etc.

Routine: SCAL4
Priority: N/A
Residency: Overlay area 1, 55
Size: 643K

This routine is similar to SCAL1 except that it is for USER 4 on the system, erosion/corrosion.

Some of the computed quantities are the gas molecular weight, the mass flow of combustion air into the facility, and air-to-fuel ratios in the combustion cells.

Routine: SEQMES
Priority: N/A
Residency: Overlay area
Size:

SEQMES is called by a version of SEQUENCE. It decodes a control flag that is passed to it and performs the proper function, such as outputting a value to a DAC, or printing a message. It then returns to the proper version of SEQUENCE.

Routine: SEQMES 2
Priority: N/A
Residency: Overlay area
Size:

SEQMES is called by SEQUENCE and provides some additional functions that would not fit in the SEQMES routine. Most of the functions performed by SEQMES 2 are associated with writing data values to the tape and line printer.

Routine: SEQUENCE
Priority: 75
Residency: Core
Size:

This routine is used to implement the sequencer control feature of ADACS. It may be activated up to eight times simultaneously to control eight different projects. The only way to distinguish between the active versions of SEQUENCE is by the data tables.

The control tables are set up as four-word commands, with sixteen bits per word, by a compiler that takes an English version of the command, such as "OPEN VALVE TI-357" or "CLEAR ACCUMULATOR". There are forty locations within the routine that are transferred to for decoding of the specific instruction that is being executed at that instant. An example of a sequencer program is given in the Appendix.

The sequencers are limited to executing ten steps per second.

Additional information about the sequencer commands can be obtained from the METC Information Release entitled "ADACS Sequential Control and Testing (SCAT)" by David R. Goff.

The commands available are:

Accumulator operations

- 1 INCREMENT ACCUMULATOR
- 2 DECREMENT ACCUMULATOR
- 3 CLEAR ACCUMULATOR
- 4 TEST ACCUMULATOR
- 5 COMPARE ACCUMULATOR
- 6 LOAD ACCUMULATOR
- 7 ADD ACCUMULATOR
- 8 OUTPUT ACCUMULATOR
- 9 LOAD SLOT
- 10 MULTIPLY
- 11 LOAD PARAMETER
- 12 DIVIDE
- 13 STORE ACCUMULATOR

Accumulator - accumulator operations

- 1 ADD ACCUMULATOR ACCUMULATOR
- 2 MULTIPLY ACCUMULATOR ACCUMULATOR
- 3 MOVE ACCUMULATOR ACCUMULATOR
- 4 SUBTRACT ACCUMULATOR
- 5 DIVIDE ACCUMULATOR

Branching operations

- 1 BRANCH
- 2 GOTO SUBROUTINE
- 3 COMPUTED GOTO
- 4 RETURN

Miscellaneous operations

- 1 WAIT BIT
- 2 WAIT CARD
- 3 OUTPUT BIT
- 4 OUTPUT CARD
- 5 NO OPERATION
- 6 DELAY
- 7 SCAN USER
- 8 STOP
- 9 IF BIT
- 10 PAUSE
- 11 PRINT
- 12 LOAD TABLE
- 13 PULSE
- 14 EXECUTE
- 15 WRITE DATA

Routine: SMSG
Priority: N/A
Residency: Overlay area 0, 5
Size: 650K

This routine is called by REPORT to write the startup information about a data acquisition task to the line printer. The information includes the number of scans that will be in the data record on tape and what the scheduling intervals are for tape, line printer, flow charts, limit checking, disks and plotting.

Routine: SMSG2
Priority: N/A
Residency: Overlay area 0, 21
Size: 307K

This is a small routine called by MAST to alert all of the users' CRTs that ADACS is coming online. It rings a bell several times, writes out a message that ADACS has come online and gives the time and date.

Routine: SPECIAL
Priority: 130
Residency: Core
Size: 617K

This routine is associated with the special slots. It determines which slots need to be scanned for information and scans them, then calls in another routine to perform the calculations.

Routine: STACK 4
Priority: N/A
Residency: Core
Size: 7K

This is a small routine that is required for the operating system to define the size of the stack partitions for each of the tasks that will be running within ADACS.

Each task has to have a portion of memory, generally 300 to 500 words, associated with it that it can use to store intermediate results. STACK 4 creates fourteen 300-word partitions, three 400-word partitions, and one 500-word partition for the B-4 system. As tasks are activated, they will latch one of these partitions and keep it until the task is discontinued.

Routine: STFIL
Priority: N/A
Residency: Overlay area 0, 7
Size: 762K

STFIL is used to read in the acquisition control files, all the information that is needed to perform data acquisition, for a given user. The files contain a list of slots associated with the user, the type of processing to be performed, the engineering units, scheduling intervals for all of the various output devices, etc.

STFIL places the files in extended memory for REPORT to use, and on tape for the Computer Math Branch.

Routine: TWRITE
Priority: N/A
Residency: Core
Size: 215K

TWRITE is used to write records on to magnetic tape. Any routine that needs something written on tape tells TWRITE how many words to write and where the starting point of the information.

If any difficulties are encountered then an error handling routine will be called in which will try to resolve the problem with the tape unit, and, if that cannot be done, switch to the other tape unit.

TWRITE also keeps track of the number of records and words written to the tape so that the amount of tape left can be estimated.

Routine: TSK 13
Priority: 13
Residency: Core
Size: 40K

TSK 13 acquires all of the necessary resources for LCAL and then calls in LCAL. Once it is through the task is killed.

Routine: TSK60
Priority: 60
Residency: Core
Size: 25K

This is a small routine that draws a background grid on a CRT. This was placed in a task so that it wouldn't tie up any other task for the amount of time needed. After the grid is drawn the task is killed.

Routine: ULTRA
Priority: 130
Residency: Core
Size:

ULTRA takes the readings from the ultrasonic detectors that are mounted on the producer dust cyclone and converts them to milli-inches so that the cyclone wear can be determined. It only runs when the producer USER task is scanning.

Routine: USER
Priority: 79-87
Residency: Core
Size: 676K

USER may be started up to nine times, once for the master CRT and once for each user. It reads the inputs from the crt terminals. When a key is pressed at a terminal an interrupt is generated. The system picks up the interrupt, identifies which it came from, and sends the input to the appropriate ID version of USER, which then puts the input into a buffer.

Once a full line of input is available, it is passed to the communications interpreter, which will try to act on the command.

Routine: VECTOR
Priority: N/A
Residency: Overlay area 2, 1
Size: 76K

This routine takes a set of x and y coordinates and moves the cursor of a Tektronix terminal to that location on the screen.

APPENDIX H
COMMONS

<u>COMMONS</u>	<u>ARRAYS</u>	<u>TYPE</u>	<u>SIZE</u>	<u>REMARKS</u>	
WINDOW	IREF	Integer	1024	Map location	
	VMIN	Integer	1		
	VMAX	Integer	1		
DISCRETE	DIGIN	Integer	1	} Bit map 60K-77K	
	DIGOUT	Integer	1		
	DOUT	Integer	24	0:2, 0:7 60K-67K	
BUFFER	MAXBUF	Integer	1	Limit	
	NBUF	Integer	1	Current use	
	BUF	Integer	(2000)	Tape I/O	
AANDB	CALHDR	Integer	10	Tape header	} Group
AB1	A	Real	144	0:11, 0:NBOX slopes	
AB2	B	Real	144	0:11, 0:NBOX integer	
ADC1	INADC	Integer	768	NCHAN	
SYSFO	KBOX	Integer			
	KCHAN	Integer			
	KLIM	Integer			
	KALM	Integer			
	KFLF	Integer			
	KFNF	Integer			
	KENG	Integer			
	KUSER	Integer			
	KSEQ	Integer			
	KDAC	Integer			
	KRANGE	Integer			
SYSFIL	CFHEAD	Integer	10	Header	}
SYSF1	ENGU	Real	64	0:31	
SYSF2	FL	Real	200	(2,50) (slope, Integer)	
SYSF3	FN	Real	180	3, 15, NFNF	
SYSF4	LIMITS	Real	256	2, 64	
SYSF5	ALARMS	Real	256	2, 64	
SYSF6	RANGE	Real	64	2, 16	

<u>COMMONS</u>	<u>ARRAYS</u>	<u>TYPE</u>	<u>SIZE</u>	<u>REMARKS</u>
DACOU	JDAC	Integer	32	(0:31) -1 if unused = 0-2048 0-10 volts
PLOT	IPLOT	Integer	6	(-1:NUSER-1) controls plot mode
PLOT1	YLL	Real	12	(-1:NUSER-1)
PLOT2	YUL	Real	12	(-1:NUSER-1)
PLOT3	LABEL	Double	24	(-1:NUSER-1)
PLOT4	SCALE	Integer	6	(-1:NUSER-1) hrs,min,sec
PLOT5	XLL	Real	12	(-1:NUSER-1)
PLOT6	XUL	Real	12	
PLOT7	FILEX	Integer	6	Current open normal file
QTYIN	QBUF	Integer	420	(70, -1:NUSER-1)
QTY1	QFLAG	Integer	6	Terminal mode
QTY2	QNB	Integer	6	# bytes
QTY3	QLOCK	Integer	6	Interlock
DISKF	DFLAG	Integer	5	0:NUSER-1 Recall file status
DISK1	DCNT	Real	16	0:7 disk record counter
DISK2	DISKT	Integer	5	Recall file update interval
TEMPM	NTEMP	Integer	1	# temp
	TMIN	Integer	6	Slots
FLOVERLAY	FALM	Integer	5	# alarms
FLOV1	FNUM	Integer	5	Currently loaded overlay
FLOWC	IREQC	Integer	5	Request update flag
FLOW1	ICRT	Integer	5	CRT type 4006,4014
LPTC	IREQL	Integer	5	Request update flag
COMMAND	CMD	Integer	24	4, 0:NUSER-1 COMM → REPORT

<u>COMMONS</u>	<u>ARRAYS</u>	<u>TYPE</u>	<u>SIZE</u>	<u>REMARKS</u>
USEREF	ICOMM	Integer	1	
	USRPT	Integer	10	2, 0:NUSER-1 pointer, # words in buffer area
USRBUF	ALLOC	Integer	20	4, 0:NUSER-1 allocate memory to users
MUX	ITIM	Integer	3	hrs, min, sec
	NSCAN	Real	10	real scan counters (seconds)
LOGIT	ENTCNT	Integer	12	# log and data entries
RCALL	RNAME	Integer	8	base recall file name
	RMASK	Real	16	limit on disk size
	RTIME	Integer	24	hrs, min, mo, day, hy, interval
COMCNT	ICOMCNT	Integer	1	# commands
	IOK	Integer	1	command file status flag
SEQCNT	ISEQCNT	Integer	1	# commands
	SEQOK	Integer	1	command file status flag
SCRATCH	ISCRTH	Integer	30	comm. scratch
SYSACT	XACT	Integer	20	see ADACS4
PRNAME	PNAME	Double	20	project names (8 letters)
KKPART	JJPART	Integer	3	sets stack limits for tasks
GCDATA	POINT	Integer	4	current gas 1 8
	GASES	Integer	32	(4,8) magnitude of each gas
	ANALZ	Real	32	(4,4) mol wt, Btu gross, Btu net, ave. dens.
ALIVE	RUNNING	Integer	1	1 bit/user
	ICHECK	Integer	1	1 bit/user
INTER	ITDAT	Integer	1	pointer to intertask storage
LPBUF	LPLOCK	Integer	1	LPT lock
	LPPOINT	Integer	1	pointer to LPT buffer area
	BUFLOCK	Integer	1	buffer lock

<u>COMMONS</u>	<u>ARRAYS</u>	<u>TYPE</u>	<u>SIZE</u>	<u>REMARKS</u>
SYSCNT	ISYS	Integer	20	system performance counters
QTYZZ	QUSER	Integer	5	associates user and CRT
CONTROL	CFLAG	Integer	10	status/mode control
SINH	SQINH	Integer	0	
MTRECS	NFEET	Real	1	# feet available
	TRECS	Real	1	# records written
	TWORDS	Double	1	# words written
TAPE	TSW	Integer	1	hard status
	EC	Integer	1	error code
	MTF	Integer	1	soft status
	TSHEAD	Integer	10	header
	TSN	Integer	12	Tape ser. #'s
USRIN	THEAD	Integer	10	header
	IBUF	Integer	40	common storage
	IPT	Integer	1	
	IVAL	Integer	1	
	ITYPE	Integer	1	
	INTER	Integer	1	
	ICIIAR	Integer	1	character
	ICLASS	Integer	1	type of character
	NB	Integer	1	
	IFLD	Integer	1	
	IFIL	Integer	1	
	ITERM	Integer	1	
	IRETN	Integer	1	
	IUSER	Integer	1	
	VAL	Real	1	
EXMEM	LXMEM	Integer	1	extended mem interlock

<u>COMMONS</u>	<u>ARRAYS</u>	<u>TYPE</u>	<u>SIZE</u>	<u>REMARKS</u>
MEMERR	MEMFLAG	Integer	1	
	AREG	Integer	1	
	BREG	Integer	1	
CRTBUF	CRTLOCK	Integer	1	
	CRTPOINT	Integer	1	
KLCAL	ICAL	Integer	1	counter
	LCINT	Integer	1	interval
ALCNT	IALCNT	Integer	1	
	AOK	Integer	1	
CZOLD	CXOLD	Real	18	last calculation 1/terminal
VZLOCK	VXLOCK	Integer	8	
ABLOCK	MUXLOCK	Integer	1	MUX interlock
ADCDN	ADCLOCK	Integer	1	ADC interlock
LPSTAT	LPSTAT	Integer	1	
ISC	CDATA	Integer	24	color CRT controls

APPENDIX I
Magnetic Tape Format

MORGANTOWN ENERGY TECHNOLOGY CENTER

MORGANTOWN, WEST VIRGINIA

DATE:

4/25/80

IR No. 160

(Rev. 5)

ORIGINATOR:

David R. Goff *DRG*

RECIPIENT:

Distribution

PROJECT:

Instrumentation Branch (MQ)

TITLE:

Updated Magnetic Tape Format for the Automatic Data Acquisition and Control System (ADACS)

INFORMATION RELEASE/REQUEST:

Introduction

This document gives a detailed description of the format of the magnetic tapes produced by the Automatic Data Acquisition and Control Systems (ADACS). The format of the magnetic tapes has been recently modified to expand the capability of the ADACS. All of the features described in this document have been implemented.

Most of the changes in the ADACS tape format are due to new features and generalization of the ADACS program. The main new ADACS feature is the refinement of the automatic tape switching feature and additional data from Chromatographs.

The format of the ADACS tapes is designed to allow for much easier and less error prone analysis of the data tapes. The data on the tape is grouped into one of 16 possible record types. Some of the record types only convey status information while others contain various types of data. Each record is explicitly identified as to its type and content by means of a 10- or 14-word header.

The following describes each record type, its use and format:

I. "START ADACS SYSTEM"

This record type indicates the ADACS system has been brought on-line. It consists of a 10-word header plus a 12-word buffer that holds the tape serial numbers. This will be useful in determining which tape was used when the current tape ended. It should be the first record on the tape. The format is as follows:

Word	1	=	ascii "SS" or "RS"
Word	2	=	-1 (indicates it refers to all users)
Words	3-5	=	time (hours:minutes:seconds)
Words	6-8	=	date (month/day/year)

DISTRIBUTION:

AAPitrolo	GEFasching	MCerullo	MDemicco
WtWertman	JGCraig	JEDowney	JMercer
JENotestein	TWKeach	DRGoff	Library

Word 9 = 1 if tape is from B-1 System
 = 4 if tape is from B-4 System
 = 15 if tape is from B-15 System
 Word 10 = active tape unit (0 or 1)
 Words 11-13 = serial number for tape unit 0 (6-character ascii)
 Words 14-16 = ignore
 Words 17-19 = serial number for tape unit 1 (6-character ascii)
 Words 20-22 = ignore

If an old data tape is being used (i.e., a restart), the identifier will be "RS".

II. "TERMINATE ADACS SYSTEM"

This record type indicates the ADACS system has been shut down. It should be the last record on the tape followed by two end-of-file marks. It consists of a 10-word header and no additional data. Its format is:

Word 1 = ascii "TS"
 Word 2 = -1
 Words 3-5 = time system was shut down (hours:minutes:seconds)
 Words 6-8 = date (month/day/year)
 Word 9 = 0
 Word 10 = 0

III. "INITIATE ACQUISITION"

Indicates that data acquisition has begun on one of the eight projects (also called users) serviced by the ADACS. The user control information is written onto tape in up to nine records labelled "I1" through "I9". Each record consists of a 10-word header followed by control file information. The control files contain the "IN", "NF", "LIM", "IDEN", "MCRT", "CRTXY", and "CPOST" arrays. Each user (project) will have its own unique set of these arrays. The first three arrays are integer arrays that each contain (NVALU) elements, where NVALU is the number of slots the project is scanning. The "IDEN" array contains the 8-character ascii identifier for each slot and is thus (4*NVALU) words long. The "MCRT" and "CRTXY" arrays contain control data for the user CRT. Each contains (NVALU) elements. The "CPOST" array contains post processing control information.

The records are broken down as follows:

Record 1

Word 1 = ascii "I1"
 Word 2 = project number (0 thru 7)
 Words 3-5 = time acquisition started
 Words 6-8 = date
 Word 9 = NPOST
 Word 10 = NVALU
 Words (11) - (NVALU + 10) = "IN" array
 Words (NVALU + 11) - (2*NVALU + 10) = "NF" array

Record 2

Word 1 = ascii "I2"
Words 2-10 = same as Record 1
Words (11) - (NVALU + 10) = "LIM" array
Words (NVALU + 11) - (2*NVALU + 10) = First $\frac{1}{2}$ of "IDEN" array

Record 3

Word 1 = ascii "I3"
Words 2-10 = same as Record 1
Words (11) - (2*NVALU + 10) = Middle $\frac{1}{2}$ of "IDEN" array

Record 4

Word 1 = ascii "I4"
Words 2-10 = same as Record 1
Words (11) - (NVALU + 10) = Last $\frac{1}{2}$ of "IDEN" array
Words (NVALU + 11) - (2*NVALU + 10) = "MCRT" array

Record 5

Word 1 = ascii "I5"
Words 2-10 = same as Record 1
Words (11) - (NVALU + 10) = "CRTXY" array
Words (NVALU + 11) - (2*NVALU + 10) = First NVALU words of "CPOST"

Records 6 thru 9

Word 1 = ascii "I6" thru "I9" (all are optional)
Words 2-10 = same as Record 1
Words (11) - (2*NVALU + 10) = 2*NVALU words of "CPOST"

The arrays should be dimensioned as follows:

INTEGER IN(NVALU), NF(NVALU), LIM(NVALU), MCRT(NVALU), CRTXY(NVALU)

INTEGER CPOST(NPOST, 20)

DOUBLE PRECISION IDEN(NVALU)

The "IN" array contains the slot numbers and corresponds to the sequence of the data in the "DATA RECORD" described later. For the ADACS system, the slot number is simply 128*Box + channel.

The "NF" array contains several pointers and control bits. The left byte of "NF" (bits 0-7) contains the function pointer. It points to the "FL" array if positive or "FN" array if negative. The "FL" and "FN" arrays are described later. The least significant five bits of "NF" (bits 11-15) contain the engineering units pointer. It points to the "ENGU" array described later. The other bits are control bits that are probably of no interest to an analysis program.

The "LIM" array contains a pointer to the "LIMITS" array which contains floating point operating limits and the "ALARMS" array which contains two floating point limits that define a critical operating condition. Bits 8-15 is the limit pointer. Bits 4-7 is the range pointer which points to an entry in the "RANGE" array.

The "IDEN" array contains eight ascii characters per slot. The eight characters serve to uniquely identify each slot.

IV. "TERMINATE ACQUISITION"

This indicates that a project (user) has ceased acquiring data. It consists of a 10-word header and no additional data. Its format is:

Word 1 = ascii "TA"
Word 2 = project number (0 thru 7)
Words 3-5 = time acquisition was terminated
Words 6-8 = date
Word 9 = 0
Word 10 = 0

V.1. "USER INPUT LOGGING"

This record type is used to record operator logging inputs as they are received. It consists of a 10-word header plus a 40-word (80-character) buffer holding the ascii input line typed by the operator. Its format is:

Word 1 = ascii "LG"
Word 2 = project number (-1 thru 7); -1 indicates the input is from the master CRT
Words 3-5 = time the logging input was received
Words 6-8 = date
Word 9 = ascii "LO"
Word 10 = number of ascii characters in the input line
Words 11-50 = input text (up to 80 ascii characters) NOTE: Text will be padded with trailing spaces.

V.2. "USER DATA LOGGING"

This record type is used to record operator entered data values. Each data value consists of a two ascii character identifier and a floating point data value. The record is 50 words, with the first 10 words used as a header and the remaining 40 words holding up to 10 data values. Its format is:

Word 1 = ascii "LG"
Word 2 = project number (0 thru 7); the master CRT can not enter data.
Words 3-5 = time the data was input
Words 6-8 = date
Word 9 = ascii "DA"
Word 10 = number of data values (1 thru 10)
Word 11 = identifier for data value #1
Word 12 = 0
Words 13-14 = floating point data value #1

Word 15 = identifier #2
 Word 16 = 0
 Words 17-18 = data value #2

 Word 47 = identifier #10
 Word 48 = 0
 Words 49-50 = data value #10

VI. "DATA RECORD"

The record contains scanned data from one of the eight projects. It consists of a 14-word header followed by the data. Several data scans may be packed into a given data record. Each data scan will be preceded by a 14-word header. However, all of the data scans in a given data record will be from the same project. The data will always be floating point, so two 16-bit words are used for each slot. The data will be converted to engineering units wherever possible.

The format of the data record is:

Word 1 = ascii "DR"
 Word 2 = project number (0 thru 7)
 Words 3-5 = time the data was scanned
 Words 6-8 = date
 Word 9 = tape output interval (seconds)
 Word 10 = number of 16-bit data words following this header

Remember, there are two words per slot, so the number of slots = (word 10)/2.

Words 11-12 = real scan counter (seconds of operation)
 Word 13 = number of data scans in this data record
 Word 14 = disk scheduling interval (not required for data processing)
 Words 15... = data

In order to be able to correlate the data values with slot numbers, refer back to the "IN" array read in from the "INITIATE ACQUISITION" header. Likewise, the eight-character ascii identifier could be obtained by referring back to the "IDEN" array and so on. All of the seven control arrays correspond exactly to the order of the data in the data record. Note that each project has a different set of control arrays.

VII. "CALIBRATION RECORD"

This record contains the calibration data generated at 20-minute intervals. It consists of a 10-word header followed by the "A" values and "B" values for all multiplexor boxes and gain settings. The "A" values are the gain correction factors and the "B" values are the offset correction factors. This record is no longer routinely written on the tape. It will only appear when hardware problems are suspected.

Its format is:

Word 1 = ascii "CD"
Word 2 = -1 (indicates it refers to all projects)
Words 3-5 = time of the calibration
Words 6-8 = date
Word 9 = 0
Word 10 = NBOX (highest number mux box in the ADACS)
Words 11 - $(24*(NBOX + 1) + 10)$ = "A" values
Words $(24*(NBOX + 1) + 11) - (48*(NBOX + 1) + 10)$ = "B" values

The calibration data is placed on the tape for monitoring purposes. It can be used to detect an out-of-calibration condition or a hardware failure in the ADACS equipment.

The arrays are dimensioned:

```
REAL A(0:NBOX, 0:11), B(0:NBOX, 0:11)
```

The second dimension corresponds to the 12 possible gain settings.

VIII. "ERROR RECORD"

This record type is used to record error messages generated by the ADACS System. Only serious error conditions will be recorded. The record consists of a 10-word header followed by 10 words of data.

At present, there are two types of error records:

1. Limit check error

Word 1 = ascii "ER"
Word 2 = project number (0 thru 7)
Words 3-5 = time
Words 6-8 = date
Word 9 = number of slots in this record (1 thru 7)
Word 10 = 0
Words 11-13 = "LI", "MI", "T"
Word 14 = slot #1
Word 15 = slot #2
Word 16 = slot #3
Word 17 = slot #4
Word 18 = slot #5
Word 19 = slot #6
Word 20 = slot #7

2. Producer level detector status

Word 1 = ascii "ER"
Word 2 = 1 (producer project)
Words 3-5 = time
Words 6-8 = date
Word 9 = unused

Word 10 = 3; void detected in producer bed
 = 4; void cleared in producer bed
 Words 11-20 = "PR", "OD", "UC", "ER", " B", "ED", " S",
 "TA", "TU", "S "

IX. "SYSTEM SIZING RECORD"

The recent expansion of ADACS systems has made it necessary to add this record type. Up to this point, all ADACS systems have had identical system sizes and configurations. This can no longer be maintained. This record type should allow full versatility for all anticipated ADACS systems.

Word 1 = "SZ"
 Word 2 = -1 (all projects)
 Words 3-5 = time
 Words 6-8 = date
 Word 9 = unused
 Word 10 = unused
 Word 11 = NENG (number of engineering units for this system)
 Word 12 = NFLF (number of linear functions)
 Word 13 = NFNF (number of nonlinear functions)
 Word 14 = NLIM (number of limit values)
 Word 15 = NALM (number of alarm values)
 Word 16 = NUSER (maximum number of users on this system)
 Word 17 = NSEQ (maximum number of sequencers on this system)
 Word 18 = NRANGE (maximum number of ranges in this system)
 Words 19-50 = reserved for future use

X. "SYSTEM CONTROL FILES"

This record contains the main system control arrays. It should be found near the start of each data tape. It contains a 10-word header followed by the "ENGU", "FL", "FN", "LIMITS", and "ALARMS" arrays.

The "ENGU" array contains four-character engineering units identifiers, i.e., PSIG and DEGF. The "FL" array contains linear functions consisting of a slope and intercept. The "FN" array contains non-linear (actually piecewise linear) functions with up to 15 segments each.

The "LIMITS" and "ALARMS" array each contain pairs of floating point values that define the operating and alarm limits of the input data. Its format is:

Word 1 = ascii "CF"
 Word 2 = -1
 Words 3-5 = time the control files were written
 Words 6-8 = date
 Word 9 = 1 or 4 or 15 (ADACS System number)
 Word 10 = number of words in all of the following arrays
 Next 2 *NENG words = "ENGU" array
 Next 4 *NFLF words = "FL" array
 Next 90 *NFNF words = "FN" array
 Next 4 *NLIM words = "LIMITS" array
 Next 4 *NALM words = "ALARMS" array
 Next 4 *NRANGE words = "RANGE" array

The array should be dimensioned using the values read from the "SZ" record. If the "SZ" record is not found before the "CF" record, use the following default values:

NENG = 32
NLIM = 64
NALM = 64
NFLF = 50
NFNF = 5
NRANGE = 16

REAL ENGU(0:NENG-1), LIMITS(2,0:NLIM-1), ALARMS(2,0:NALM-1)

REAL FL(2, NFLF), FN(3, 15, NFNF), RANGE (2, 0: NRANGE-1)

XI. "CONTINUATION RECORD"

These record types indicate that data acquisition is continuous across two magnetic tapes. The first tape in the sequence will have either a "CX" or "EX" record at the end of the tape. This record will normally immediately precede the end-of-file marker. The second tape in the sequence will have a "CO" record at the start of the tape. Data acquisition records will then continue on the second tape.

First tape in sequence:

Word 1 = ascii "CX"
NOTE: If for some reason the tape switch can not be made,
word 1 will contain ascii "EX"
Word 2 = -1
Words 3-5 = time
Words 6-8 = date
Word 9 = 0
Word 10 = currently active tape unit (0 or 1)
Words 11-13 = serial number for tape unit 0
Words 14-16 = ignore
Words 17-19 = serial number for tape unit 1
Words 20-22 = ignore

Second tape in sequence:

Word 1 = ascii "CO"
Words 2-22 = same as "CX" record above

XII. "SLOT IDENTIFIER RECORD"

This record type contains a description of the transducer connected to each slot. Up to 32 characters of text is allowed for each slot.

This record contains a description for all slots in the system. The format is:

Word 1 = ascii "ID"
 Word 2 = -1
 Words 3-5 = time
 Words 6-8 = date
 Word 9 = 0
 Word 10 = NSLOTS (number of slots described in this "ID" record)
 Word 11 = slot #1
 Words 12-27 = text description #1
 Word 28 = slot #2
 Words 29-44 = description #2
 .
 .
 .
 .
 .
 Word 1694 = slot #100
 Words 1695-1710 = description #100

The arrays to receive the data are as follows:

INTEGER SLOTS (NSLOTS)
 DOUBLE PRECISION DESC (4, NSLOTS)

NOTE: No more than 100 slots will be contained in any given "ID" record. If the ADACS contains more than 100 slots, several "ID" records will be written. Slot numbers will be in ascending order. Only active slots will have a description.

XIII. "GAS CHROMATOGRAPH DATA RECORD"

This record type handles data from the Bendix and Greenbrier gas chromatographs. This record is written onto tape whenever a complete set of gasses have been received from one of the chromatographs. This record will be written regardless of which users may be running. The gas components will be converted to a percentage of the total gas (0-100 percent). The format of the record is:

Word 1 = ascii "GC"
 Word 2 = 1 (producer)
 Words 3-5 = time
 Words 6-8 = date
 Word 9 = chromatograph pointer
 1 → Bendix 6000 #2
 2 → Bendix 6000 #3
 3 → Bendix 7000 #4
 4 → Bendix 6000 #1
 Word 10 = 8 (number of gas components)
 Words 11-12 = H₂ component (real number) percentage
 Words 13-14 = CO₂ component
 Words 15-16 = C₂H₆ component
 Words 17-18 = H₂S component
 Words 19-20 = O₂ component
 Words 21-22 = N₂ component
 Words 23-24 = CH₄ component
 Words 25-26 = CO component
 Words 27-28 = sum of all gas components
 Words 29-30 = average molecular weight of the gas

Words 31-32 = average density of the gas in grams per liter
Words 33-34 = average specific gravity of the gas
Words 35-36 = gross Btu's per cubic foot for the gas
Words 37-38 = net Btu's per cubic foot for the gas

XIV. "SEQUENCER RECORD"

The ADACS System also has the capability of performing control functions. One immediate use of this capability will be the sequencing for the Valve Test Project. The ADACS can support up to eight simultaneous sequencer tasks. Each sequencer has a number of types of possible tape records. The format is:

Sequencer messages

Word 1 = ascii "SQ"
Word 2 = project number (0 thru 7)
Word 3-5 = time
Word 6-8 = date
Word 9 = sequencer number (0 thru 7)
Word 10 = unused
Words 11-50 = ascii message

XV. "PROJECT NAME RECORD"

This record is used to convey the name of each ADACS project on the system producing the data tape.

Word 1 = "NM"
Word 2 = -1 (refers to all projects)
Words 3-5 = time
Words 6-8 = date
Word 9 = highest user number allowed in this ADACS
Word 10 = unused
Words 11-14 = project 0 name (ascii, left justified, padded with spaces)
Words 15-18 = project 1 name
.
.
.
.
.
Word 39-42 = project 7 name

XVI. "SEQUENCER DATA"

This record allows the sequencer programs to record data onto tape. This allows data to be distilled down and sorted before it is written to tape. It is very similar to the "USER DATA LOGGING" record described in section V.

Word 1 = ascii "SD"
Word 2 = project number (0 thru 7)
Words 3-5 = time
Words 6-8 = date
Word 9 = ascii "DA"
Word 10 = number of data values (1 thru 10)
Word 11 = identifier for data value #1
Word 12 = 0

Words 13-14 = floating point data value #1

.
.
.

Word 47 = identifier #10

Word 48 = 0

Word 49-50 = data value #10

Conclusion

The magnetic tape format described in this document is being used to maximize convenience and minimize lost or misinterpreted data. Because of this, the format may be modified or added to in the future as experience is gained in analyzing ADACS tapes.

4:bl:604ef