PNL-6317
UC-95

18J

PCDAS Version 2.2

Remote Network Control and Data Acquisition

M. J. Fishbaugher

September 1987

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

Battelle

PNL-6317

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

PNL-6317
UC-95

PCDAS VERSION 2.2

REMOTE NETWORK CONTROL AND DATA ACQUISITION

M. J. Fishbaugher

September 1987

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

CONTENTS

FIGURES

TABLES

1.0 INTRODUCTION

1.1 MANUAL PURPOSE

This manual is intended for both technical and non-technical people who want to use the PCDAS remote network control and data acquisition software. If you are unfamiliar with remote data collection hardware systems designed at Pacific Northwest Laboratory (PNL), this introduction should answer your basic questions.

Even if you have some experience with the PNL-designed Field Data Acquisition Systems (FDAS), it would be wise to review this material before attempting to set up a network. This manual was written based on the assumption that you have a rudimentary understanding of personal computer (PC) operations using Disk Operating System (DOS) version 2.0 or greater (IBM 1984). You should know how to create subdirectories and get around the subdirectory tree.

1.2 READING HINTS

If you are interested in the day-to-day operations of PCDAS, read Sections 2 through 5. These sections cover the information you will need in order to set up a remote network and collect data. Section 5 also includes information on conversion of data to ASCII files with engineering units. Section 6 is a technical description of each of the functional areas of PCDAS.

A listing of the message data base and associated explanations are included in Appendix A. Appendices B through G are included to help you set up and manage a remote collection network.

Version 2.2 of the PCDAS software is designed to work with all versions of the FDAS firmware. See Appendix D for a summary of the FDAS single-character commands for PCD_ processor types.

1.3 WHAT IS REMOTE DATA COLLECTION ANO NETWORK CONTROL?

Recent advances in microcomputer technology have precipitated many advances in automated data collection. At PNL, a remote data logging microcomputer, called a Field Data Acquisition System (FDAS), has been designed. This unit provides relatively inexpensive remote data collection in numerous applications. The PCDAS software package was designed to aid in the automatic collection of data from a network of FDAS units. PCDAS also provides utilities for network control and data management.

Most remote collection networks can be divided into simple functional areas. A block diagram of a typical network is shown in Figure 1.

The FDAS conditions the signals from the sensors and records the data in temporary storage. Each field unit has a dedicated or shared phone line for communications. Note that the information flow arrows in Figure 1 point in both directions, indicating both the data collection and FDAS control paths.

The data collection computer initiates phone calls and collects data from all of the remote units in the network. If PCDAS is used on a very large network of FDAS units (e.g., greater than 2000 measurement points), the data can be converted and transmitted to a larger mainframe or mini-computer for archiving and analysis.

The two areas enclosed in dashed lines are explained in greater detail in the next two subsections.

1.4 WHAT IS THE FIELD DATA ACQUISITION SYSTEM (FDAS)?

The FDAS (or data logger) is a micro-computer that collects data from a variety of sensors. This unit was designed at PNL to be used in

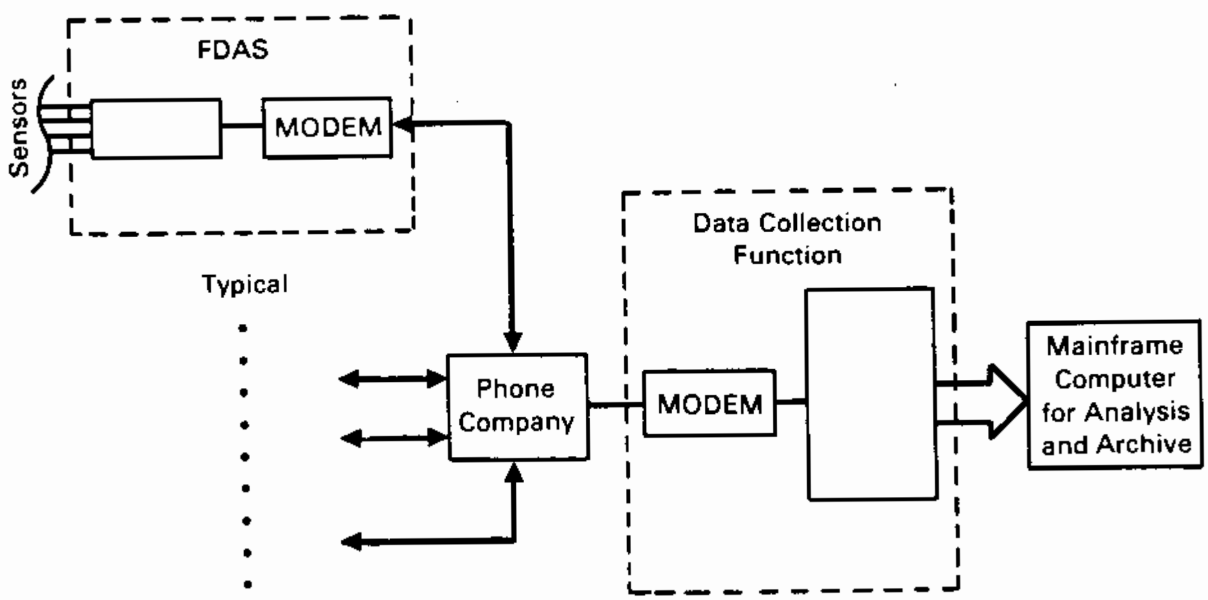


FIGURE 1. Typical Remote Data Collection Network

large electrical end-use studies. The FDAS can be configured to measure signals from many types of sensors. Figure 2 is a simple block diagram of this system.

What follows is a description of the FDAS operations. It is important for you to understand how the FDAS works if you want to use the data in any study. The FDAS operations are largely transparent to the PCDAS users, so the remainder of this section is optional reading.

The FDAS has both analog and digital inputs. Signals from analog sensors are conditioned into 0- to 5-volt dc signals and fed to the analog-to-digital converters. These devices change the signal (0 to 5 vdc) to a binary number (0 to 255 counts). The processor then scans these numbers and stores them in a temporary location. At the end of a particular time interval called the integration period, the processor

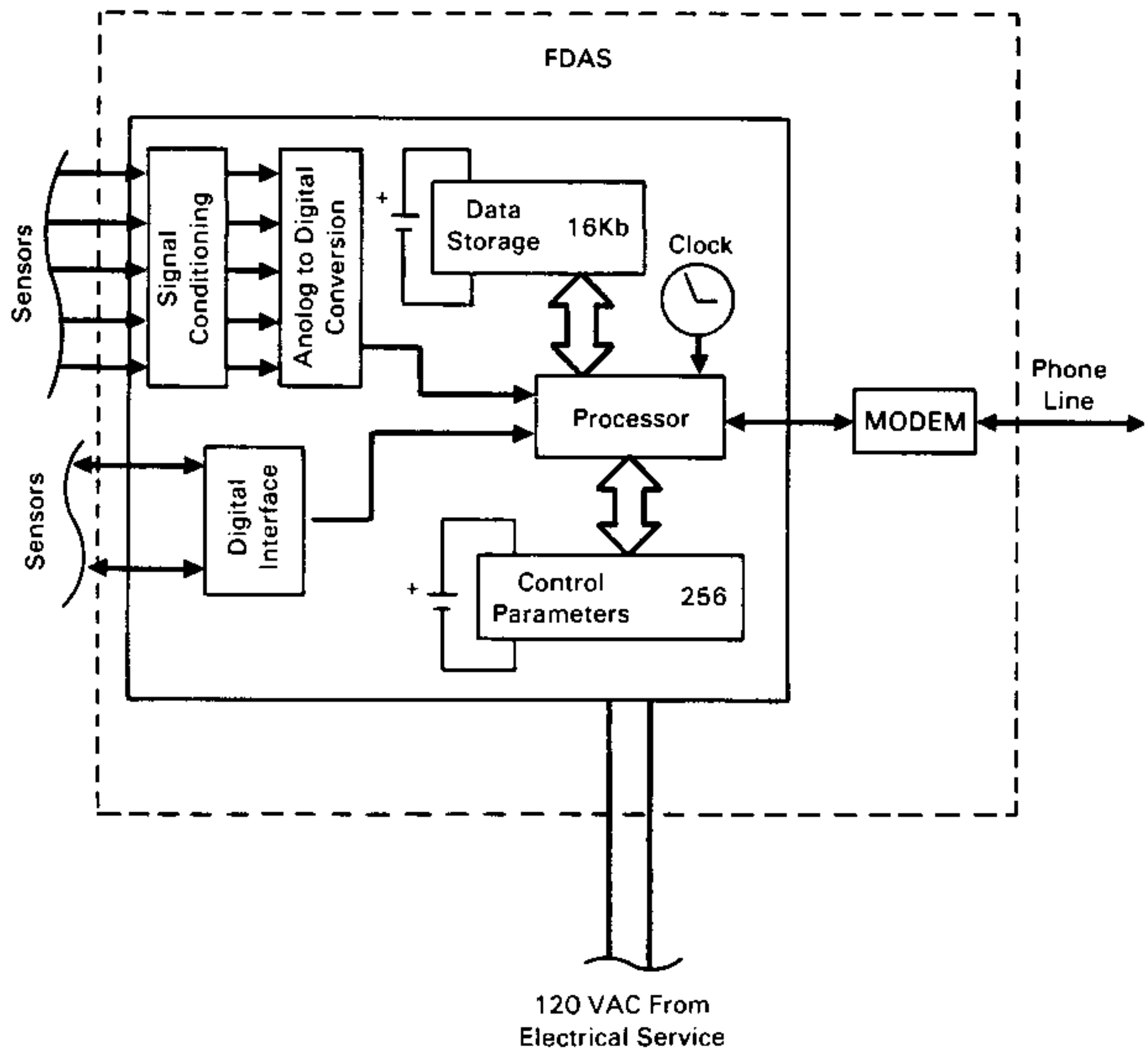


FIGURE 2. Field Data Acquisition System

writes a number to permanent memory for each active channel. This number represents the sum of all of the previous scans during that interval. The data acquisition system collects this number from each of the active channels in the remote FDAS.

The control parameters include the integration period and the list of active channels. The version 2.0 (16-kbyte FDAS) has 112 possible channels: 64 are analog inputs and 48 are digital input/output channels. In most installations, many of the channels will not be used. The control parameters are used to tell the processor to skip unused channels during a scan.

Digital channels are scanned like the analog channels. In this case, 5-vdc pulses during the integration period cause the number in temporary storage to be incremented by 1. At the end of the integration period, a number representing the total number of pulses during that period is written to data storage. Appendix C contains a more detailed discussion of FDAS data reduction and storage.

Data in the FDAS is stored in records. Each record contains the analog and digital readings taken during an interval. In addition, the records contain the time stamp of the end of the interval. Approximately 16 thousand bytes of data can be stored in the FDAS before the oldest data is overwritten. Both the integration period and the number of bytes per record (record length) determine the interval at which data must be collected from the FDAS. Appendix B contains sample polling interval and network data flow calculations.

It is important for you to understand the basic FDAS operations in order to set up your own remote collection network. This manual does not attempt to provide the information necessary for the installation of a remote FDAS. Rather, it explains the steps necessary to set up the data collection system after the remote units are installed and the installation documents have been written.

Additional information on the FDAS unit hardware and firmware is available (Tomich and Schuster 1985). Information on FDAS metering applications and installation, is available through the Energy Sciences Department of the Applied Physics Center at the Pacific Northwest Laboratories.

1.5 HOW DOES THE DATA ACQUISITION SYSTEM (PCDAS) WORK?

This section provides an overview of the major functional areas of PCDAS. Details will be covered in later sections of this users manual.

Figure 3 is a block diagram that shows the details of the data collection function left blank in Figure 1. Data is represented by six-sided figures, and programs are in rectangles.

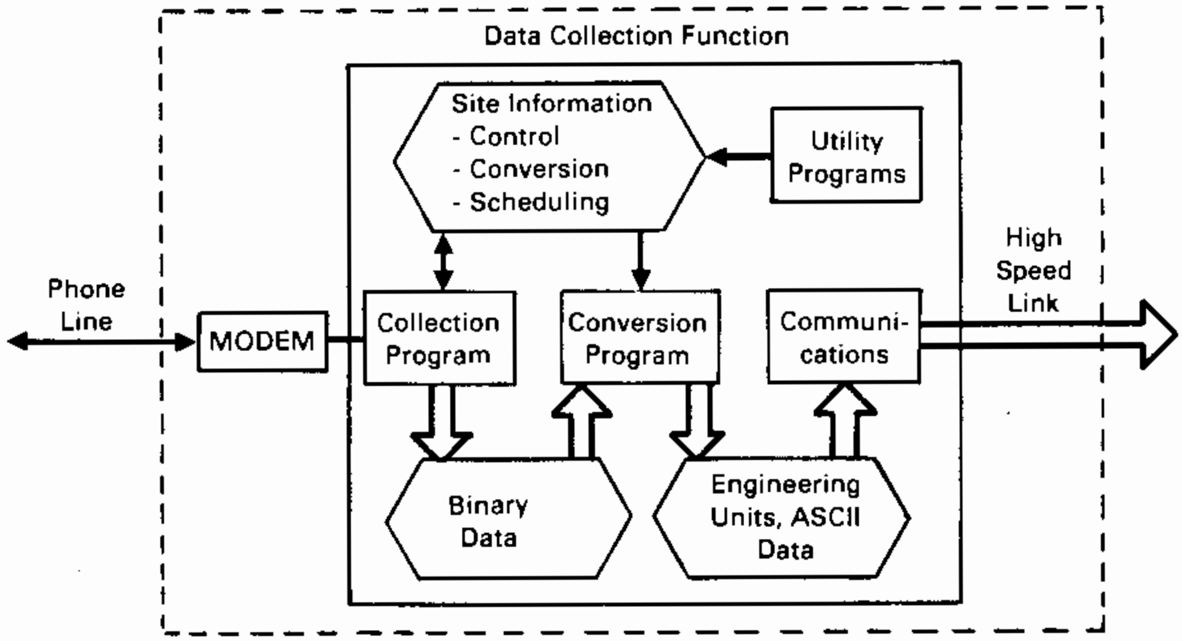


FIGURE 3. PCDAS Data Collection Function

The collection system works as follows:

- First, a set of utility programs is used to encode the measurement plans for a particular remote site. This encoding includes entering information used to control data collection and conversion of the data to engineering units.

- Next, the collection routine automatically schedules collection events, calls individual remote units, and archives the data in a binary file structure on the PC.

- If the data is to reside on some other computer, the conversion routine is used. This routine accesses the conversion information (entered earlier) and the binary data files to create larger ASCII files with engineering units. These files can then be transferred to a different computer using a commercially available data transmission package. Because the engineering data files are much larger than their binary counterparts, it may be important to have high-speed communications with the mainframe or mini-computer. All of the functional areas shown in Figure 3 will be explained in much greater detail in later sections.

PCDAS has a number of features that are designed to help you operate remote data collection networks. A few of the more important features are:

1. All FDAS clocks are synchronized to within 60 seconds of the data collection computer's clock. Data collection is synchronized on the end of each interval.

2. Extensive FDAS diagnostics are performed during each polling event to ensure constant operating parameters.

3. Scheduling phone calls, communications, and data archiving are performed automatically.

4. PCDAS has both manual and automatic operating modes.

5. The binary data archive format is compact.

6. The utilities for converting and using the data are simple.

The PCDAS system has many automatic features that help you manage a collection network. Regardless of the data collection automation, you still control the essential network parameters. You must verify the accuracy of the site-specific data base, and must also back up all critical operating files. Section 4 contains more information on data collection operations.

1.6 USER SUPPORT

Software enhancements and user support are available through the Building Sciences Section of the Applied Physics Center at the Pacific Northwest Laboratory. Support can be provided on a time-available basis for simple changes and on a contract basis for larger modifications. Please direct your comments and inquiries to:

Building Sciences Section
Applied Physics Center
Pacific Northwest Laboratory
P.O. Box 999
Richland, Washington 99352

2.0 GETTING STARTED

2.1 BACKING UP THE SYSTEM DISK - MEDIA PROTECTION

The PCDAS master disk is write-protected, but the programs are not copy protected. It is a good idea to make a backup copy of the master system disk. Feel free to make as many copies as you need. The standard installation procedures will automatically make one copy of the master disk to be used in operating the system. If you are going to use the standard PCDAS configurations, you can skip to Section 2.2.

The nonstandard installation procedure does not create a working copy of the system. On a floppy drive system you may copy the master disk by placing it in drive A, inserting a blank, formatted disk into drive B, and typing the command

```
COPY A:*.*  B:*.*
```

To copy the master disk on a hard drive system, you first need to create a subdirectory on your hard disk (drive C, for example). Place the PCDAS disk in drive A; then execute the command

```
COPY A:*.*  C:\subdirectory path\*.*
```

Store the original PCDAS disk in a safe place in case your working copy gets damaged.

For directions on how to format a new diskette or create a subdirectory, please refer to the DOS user's manual (IBM 1984).

2.2 HARDWARE AND SOFTWARE REQUIREMENTS

PCDAS is designed to run on IBM PC/XT/AT or compatible computers. Disk Operating System (DOS) version 2.0 or higher is required to support the subdirectory data structures of PCDAS.

This software package is disk access intensive. It relies heavily on disk space for filing messages to the operator, storing site-specific information, and for archiving data from the remote loggers. For this reason, it is important to have disk drives in good working order.
If the PCDAS software is to be run on a floppy disk drive system, it is necessary to have two disk drives: one for storing program files, operator message files, and site-specific data files; and the other

for archiving the data from the loggers. In a hard disk system, a directory structure provides separate locations for the different types of data and programs.

The computer that you are using must also be equipped with a Hayes command compatible modem on a serial port (COM1 or COM2). The PCDAS programs use the DTR line on your modem. This line should NOT be permanently enabled with a jumper or switch. Refer to your modem users manual for information on how to configure the DTR line. If you are using a Hayes 1200-baud smartmodem, the switch configuration should be

Switch	1	2	3	4	5	6	7	8
Config.	UP	UP	DN	UP	DN	UP	UP	DN

In addition to the AT command set, your modem must also implement the following control register:

S7 Seconds to wait for carrier.

Your computer should also have a minimum of 256 Kb of memory and a line printer. Currently, four printers are supported:

- Epson
- HP thinkjet
- Prism
- Talaris T810 laser.

File redirection of all output is available if you do not have one of the printers listed. The files created by the redirection are all 80-column standard ASCII files.

In addition to the hardware described above, you will need a copy of the BASRUN20.EXE library of subroutines that comes with the IBM BASICA compiler Version 2.0 (IBM 1985). This compiler package contains all of the necessary tools for you to modify the PCDAS software, and is required for running the programs.

A copy of the BASRUN20.EXE file should be copied from your compiler disk to the PCDAS system directory. This directory will be \PCDAS for hard disk-based PC systems and A: for floppy-based systems.

If you want to do data analysis on the PC, or if you want to transfer the data to a mainframe, you will need other commercially available software.

2.3 STANDARD PCDAS INSTALLATION

The "generic method" is the easiest manner in which to install PCDAS on your system. To install PCDAS on either a floppy or hard disk computer system, you have to know which communications port the modem is on and the type of printer that you are using. Procedures for a nonstandard installation are given in Sections 2.4 and 2.5.

2.3.1 Floppy Disk Drive System

The following steps outline the procedure for installing PCDAS on a floppy drive system:

1. Format a new floppy disk by placing the DOS disk in drive A and a new disk in drive B. Type the command

   FORMAT B:/S

   This command will format the disk in the B drive and copy the operating system onto it.

2. Remove the DOS disk from drive A and insert the PCDAS master disk.

3. To begin installation, type the command

   FLOPINST

   This will bring up a screen of information; the system will wait for verification before proceeding with the installation. To proceed, you need only hit a key. To stop the installation from proceeding, you may type a **CNTL-C**.

   The FLOPINST procedure will create all the necessary subdirectories on drive B and copy all of the needed files from the master disk.

4. When you are prompted for the communications port, type in the appropriate number.

5. Select the appropriate printer type when prompted.

6. To copy BASRUN20.EXE onto the working copy disk, place the IBM BASICA Version 2.0 compiler disk in drive A and the PCDAS working disk in drive B. Then execute the command

   COPY A:BASRUN20.EXE B:*.*

   You will not be able to operate the system without the BASRUN20.EXE file on the system disk. Once the BASRUN20.EXE file is copied to your working PCDAS disk, you will be ready to operate the system. Read Section 2.4.1 for information on the directory structure that FLOPINST created. You can ignore the "make directory" and "copy" instructions in Section 2.4.1. Section 3 covers FDAS control and communications.

2.3.2 Hard Disk Drive System

The following steps outline the procedure for installing PCDAS on a hard drive system:

1. Insert the PCDAS master disk in drive A.

2. To begin installation, type the command

   HARDINST

   This will bring up a screen of information; the system will wait for verification before proceeding with the installation. To proceed, you need only hit a key. To stop the installation from proceeding, you may type a **CNTL-C**.

   The HARDINST procedure will create all the necessary subdirectories on your hard disk and copy all of the needed files from the master disk.

3. When you are prompted for the communications port, type in the appropriate number.

4. Select the appropriate printer type when prompted.

5. There are two ways of copying the BASRUN20.EXE file into the PCDAS subdirectory, depending on where the file is located:

   - BASICA compiler residing on floppy disk

     Place the IBM BASICA Version 2.0 compiler disk in drive A and type the command

     COPY A:BASRUN20.EXE C:\PCDAS*.*

   - BASICA compiler residing on hard disk

     Given the appropriate pathname to the IBM BASICA Version 2.0 compiler, type the command

     COPY C:\pathname\BASRUN20.EXE   C:\PCDAS*.*

   The PCDAS system will not operate without the BASRUN20.EXE file. Once the BASRUN20.EXE file is copied to your working PCDAS directory on the hard disk, you will be ready to operate the system. Read Section 2.4.2 for information on the directory structure that HARDINST created. You can ignore the "make directory" and "copy" instructions in Section 2.4.2. Section 3 covers FDAS control and communications.

2.4 ALTERNATE PCDAS INSTALLATION

If you have more than one hard disk or perhaps a cartridge drive system, you may want to specify different devices for programs and data. In this case you will want to use the installation procedure outlined in this section. The INSTALL program is used to configure the data collection system after you have created subdirectories and copied the correct files from your system disk.

Before setting up PCDAS on a floppy disk drive system, it would be wise to consider the data flow anticipated in your network. In general, if your network includes more than 10 sites, it may be necessary to upgrade your PC to handle the volume of data. If data for a large network is going to be written to floppy disks with 360 Kb of capacity, you may find yourself with a large pile of disks. Appendix B contains sample data flow calculations that may help you anticipate how much data you will collect for any study.

If you are using a floppy disk-based system, particular attention should be paid to the available space on the two disk drives. Depending on the size of the network, data can fill up the disk in the B drive in a short time. Because a message file is created on the system disk drive for every collection event, the system disk can also fill up quickly. When the data disk fills up, you have to format and use another disk.

When the system disk gets full of message files, you will have to clear some of the files. The procedure for doing this is covered in Section 4.5.

2.4.1 Floppy Disk Drive System

If you have a relatively small network, running PCDAS from a floppy disk may be the most cost effective way to collect the data. The PCDAS system on floppy disks will comprise three disks. The first disk is called the PCDAS system disk and contains program files, the site-specific information, and driver files organized in subdirectories. This disk will normally be inserted into drive A when the programs are running.

The second disk is called the data disk. It will contain the daily binary data files from the FDAS units, organized by site and by month. When a data disk fills with data, another formatted disk is used. In this way a number of data disks are accumulated with data files organized chronologically across the disks. All data disks should be labeled consistently.

The third disk will be called the data conversion disk. This disk will contain the conversion program file and the appropriate driver files as well as the converted ASCII data files. The files that belong on the conversion disk are explained in a Section 5. This disk will replace the system disk in drive A when you are going to convert and use the data. The data disk containing the target time window of data will remain in drive B. PCDAS cannot convert data across different data disks. All of the target data must be on a single disk in drive B.

Figure 4 shows the typical directory structure for the PCDAS system. Drive A is the PCDAS system root directory, and the B drive is the data root directory. The "SITEDAT" subdirectory contains the site-specific information and is automatically created when you enter information.

The "SUMMARY" subdirectory shown in Figure 4 contains operator message files. Like the SITEDAT subdirectory, this subdirectory is created

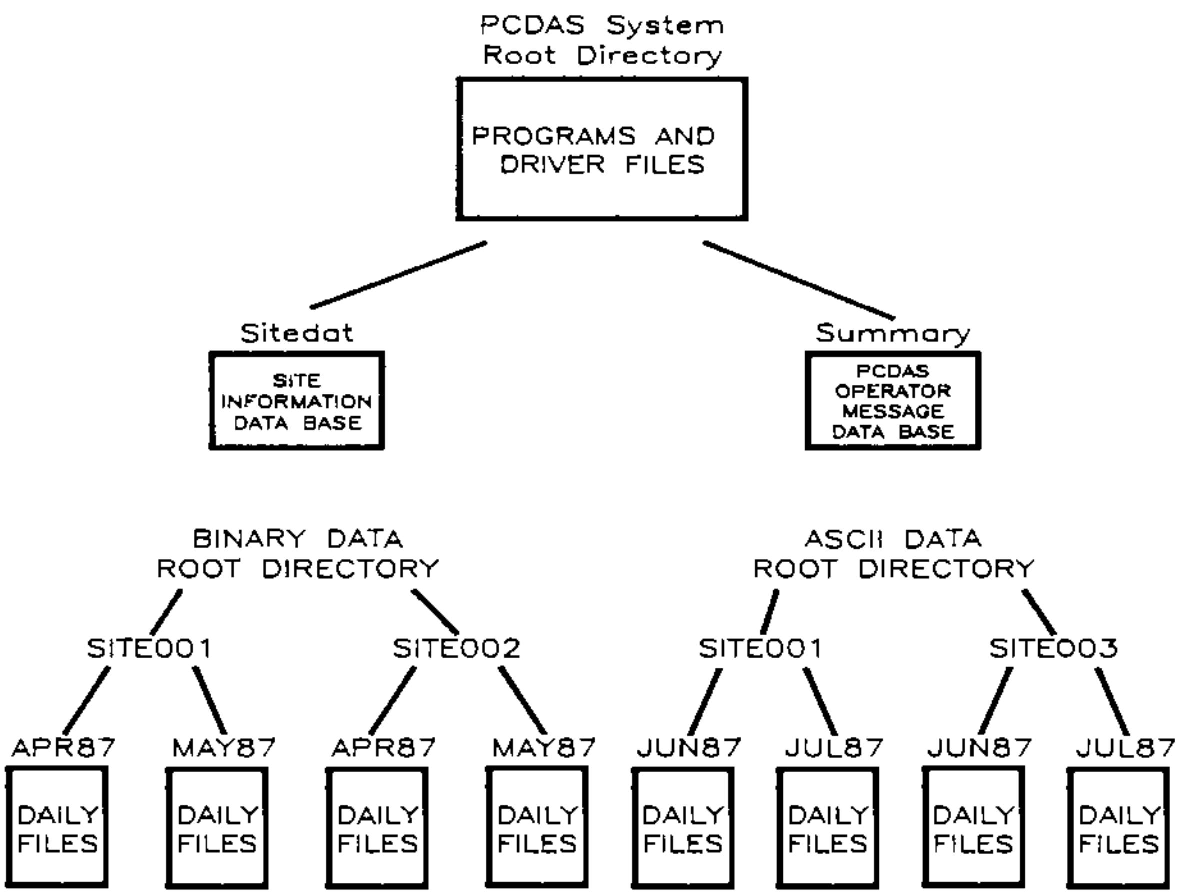


FIGURE 4. Typical PCDAS Directory Structure

automatically. The PCDAS system will use this subdirectory for writing messages to the operator. The binary coded operator message files will all have the filename extension .SUM.

In a floppy-based system, the B drive is the data destination for PCDAS. Subdirectories for each site are created automatically when data is acquired. These site-by-site subdirectories will stem from the data destination directory and will be named

SITEnnn

where "nnn" is the three-digit site number. You do not need to create any subdirectories on the data disk to install the system on your PC. All of the data collected from the sites is written to the data disk. When the disk gets full, you will have to format and label a new disk. It is important for you to keep track of the data on each disk, so you should adopt a disk numbering and labeling convention and stick with it.

The binary data files are organized by month under each SITEnnn subdirectory. This directory structure simplifies data access and backup procedures.

To have adequate disk space for converted files, you will need to create the separate data conversion disk. With a new formatted disk in B drive and your working system disk in A, type **COPY A:TRANS.EXE B:** and **COPY A:BASRUN20.EXE B:**. This will copy the conversion program to the data conversion disk. You will also have to keep an up-to-date copy of the SITE.DRV and USAGE.DAT files on this disk so the conversion program can correctly identify the sites in your network. The SITE.DRV file is updated each time data is collected from a site.

In addition to the site driver file, you will have to create a SITEDAT subdirectory on your conversion disk and copy the ".PAR" and ".DAT" files from the SITEDAT subdirectory on the PCDAS system disk into the this subdirectory on the conversion disk. The information in the ".DAT" and ".PAR" files is what the conversion program uses to convert the data into numbers with engineering units.

Once you have created the summary subdirectory on your system disk and copied the "MESSAGE.DAT", you are ready to run the INSTALL program.

2.4.2 Hard Disk Drive System

Once you have copied the program and data files onto your hard disk, you will need to make some other subdirectories stemming from your root "PCDAS" directory. The PCDAS system root directory for a hard disk system will be "C:\PCDAS". Two directories stem from this root:

- The "SUMMARY" subdirectory contains operator message files. This subdirectory is created automatically the first time you run the system.

- The "SITEDAT" subdirectory contains site-specific information and is automatically created whenever information is entered. Site-specific information includes channel descriptions, conversion factors, and other parameters that are used to control each site. The PARSET program is used to input and edit this information (Section 3).

The data destination directory contains all of the data from the remote loggers. You must create this directory with the DOS MD command. The site-by-site subdirectories shown in Figure 4 are automatically created when data is collected. Daily binary data files are organized by month under the individual SITEnnn subdirectories.

In a hard disk system, the conversion program (TRANS.EXE) resides in the PCDAS system root subdirectory. Data conversion can be performed with the TRANS program and ASCII files written to a separate subdirectory on the same hard disk where the binary data resides.

After you have created a summary file subdirectory, copied the "MESSAGE.DAT" file, and created the data destination subdirectory (or subdirectories), you are ready to run the INSTALL program.

2.5 RUNNING THE ALTERNATE INSTALLATION PROGRAM (INSTALL)

The INSTALL program tells PCDAS how you have organized your directory structure and how your hardware is configured. To run the program, set your default directory to the PCDAS system root and type

```
INSTALL
```

The program will prompt you for the data destination root directory path name. With a floppy-based system you could type "B:\" for the data destination path name and "A:\SUMMARY" for the summary file path name. For a hard disk system you could type "C:\DAS\DATA" for the binary data destination path name.

You will also have to answer a question about saving operator messages. Type a "1" if you want to save all messages and a "2" if you want to save only the most recent messages. Saving only the most recent messages will conserve disk space.

The INSTALL program will also prompt you for the modem communication port number, a yes or no response about media protection, and a modem command string to be appended to each dial-out sequence.

If you answer yes to the question about media protection, a message will be generated every time PCDAS has been run 200 times. The message will advise you to replace the disk media in order to guard against worn-out disks.

To be safe, you should back up all site-specific information (from the SITEDAT directory), as well as any system files ending with a ".DAT" or ".DRV" extension. Refer to Section 5 for more information on backing up critical PCDAS files.

The prompt for an additional modem command sequence can be used to append a dial-out string to each call. If you have to dial a "9" to get an outside line in your building, you could type **9** at the prompt. You can also use this string to specify a forced entry billing for all data collection calls.

The last prompt from the INSTALL program will be a printer selection menu. Select the printer that is installed as LPT1: on your PC. If your printer is not on the list, select the EPSON as default. All of the PCDAS programs have a file redirection option that lets you create an 80-column ASCII file rather than a printer listing.

The INSTALL program will write a small data file named "USAGE.DAT" after you select a printer. You will need to run the INSTALL again if you reconfigure your system. If you change from a floppy-based system to a hard disk system, you will need to copy all of your PCDAS operating files (SITE.DRV and SCHEDULE.DAT), data files, and site-specific files to the new file structure and then run INSTALL again.

3.0 FDAS PARAMETER CONTROL

The first task in setting up a remote data collection system is to determine what sensors are in the network. The people who install the FDAS systems communicate this information to you by the means of a measurement plan. This document will contain all of the information you will need to collect and archive data from a remote data logger. Figure 5 shows the first sheet of a blank measurement plan. There are seven sheets in a complete measurement plan, four for analog channels (channels 1 through 64) and three for digital channels (channels 65 through 112). The PARSET program has a routine for entering and editing site specific data files from these measurement plans. The digitized measurement plans are called parameter sets.

FDAS V2.2 Measurement Plan

Site#:____ Phone#:________ Date:______ Completed by:____________ Page__ of__

Analog Ch #	Description (11 char. Max)	Status on	Status off	CF1	CF2	Units	Offset Counts	Comments
1								
2								
3								
4								
5								
6								
7								
8	5VREFERENCE							
9								
10								
11								
12								
13								
14								
15								
16								

Additional Comments:

FIGURE 5. Measurement Plan Form (Channels 1 Through 16)

3.1 RUNNING THE PARAMETER SET PROGRAM (PARSET)

The PARSET program can be run by typing

PARSET

in the root directory of your PCDAS system. The name PARSET stands for PARameter SETting. As its name implies, this program lets you create the necessary site data files, communicate manually with the logger, and load the parameters to the logger. In this section we are mainly concerned with the first of these features. Other features of PARSET are discussed in Sections 3.3 and 3.4.

The main menu of PARSET is shown in Figure 6. With your measurement plan in hand, you now have to decide how to organize your network. Each site is assigned a number and a single-letter parameter set code. The site number is an integer from 1 to 300. The parameter set code is a single-letter code with 26 possible parameter sets for each site (A through Z). The site number and parameter set code is the way you will reference the site after it is entered into the PCDAS collection system, so it is important for you to understand the reasoning behind this organization.

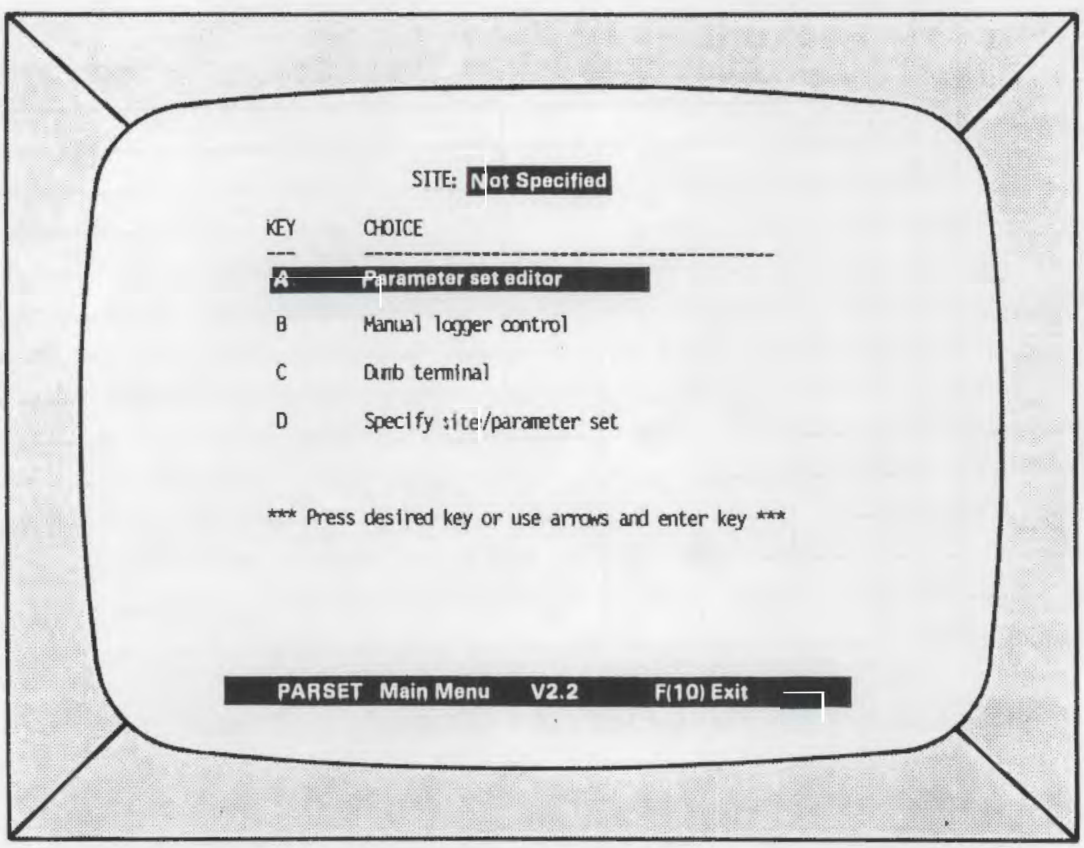


FIGURE 6. PARSET Main Menu

The site number is simply a label for each of the remote sites. Data can be collected from up to 300 sites with the PCDAS software. The maximum site number is 300.

The parameter set code is a single-letter label that uniquely identifies remote logger sensor configurations. These logger configurations will be called parameter sets.

Having a method for uniquely specifying parameter sets allows you to track the history of a site as it is reconfigured or as sensors are added and subtracted. Each parameter set for a site is represented in the PCDAS system by two files. These files reside in the SITEDAT subdirectory and have the file name convention

xSITEnnn.DAT and xSITEnnn.PAR

where "x" is the single-letter parameter set code and "nnn" is the site number with leading zeros. The ".DAT" files contain the channel descriptions and conversion data for a particular site/parameter set, and the ".PAR" files contain the binary data that is actually used to control the FDAS.

The SITEDAT subdirectory is an archive of all the ".DAT" and ".PAR" files that you have used to control your network. The site data files from old site configurations are required for the conversion and manipulation of data from that configuration. In general, you never throw away old site data files if you have collected any data under that parameter set code for a particular site.

The most obvious organization is to start with site number 1 and parameter set "A". So, at the main menu prompt, type **D**.

At the prompts, enter the site number and the single-letter parameter set code. Also, enter the baud rate for communications with the remote site (300 or 1200 baud, usually 1200, if not otherwise stated on the measurement plan). After entering the baud rate, you will return to the main menu again. Note that the site specifier at the top of the screen has changed from **"Not Specified"** to "ASITE001", indicating that you have selected a site. The first character in this site specifier indicates the parameter set code, and the last three characters indicate the site number.

You will notice that the site specifier will appear in most of the PARSET screens and in the file names of the site information data base. This program allows you to work on only one site/parameter set at a time.

To exit PARSET and return to DOS, simply press **F10** at the main menu. In general, typing **F10** in PCDAS will return you to higher levels in the system, eventually returning to DOS.

3.2 CREATING NEW SITE FILES/PARAMETER SETS

Whenever you add a site or update a parameter set, two new files are written to the "SITEDAT" subdirectory. If you are starting with site number 1 parameter set A, two files called

ASITE001.PAR
ASITE001.DAT

will be written. These files control channel activation, data reduction and storage, and data conversion to a usable form. It would be wise to always have these files backed up in a safe place. At the PARSET main menu, type **A**. If you have not specified a site and parameter set, the program will kindly ask you to do so and will return to the main menu. Use option **D** to specify the site and parameter set.

You are now ready to enter the measurement plan for site number 1 parameter set A. Option **A** from the main menu will bring up another menu called the Parameter Change Menu. Figure 7 shows the options in this menu. Note that the site specified is listed at the top of the screen. Option **A** of this menu lets you change the integration period

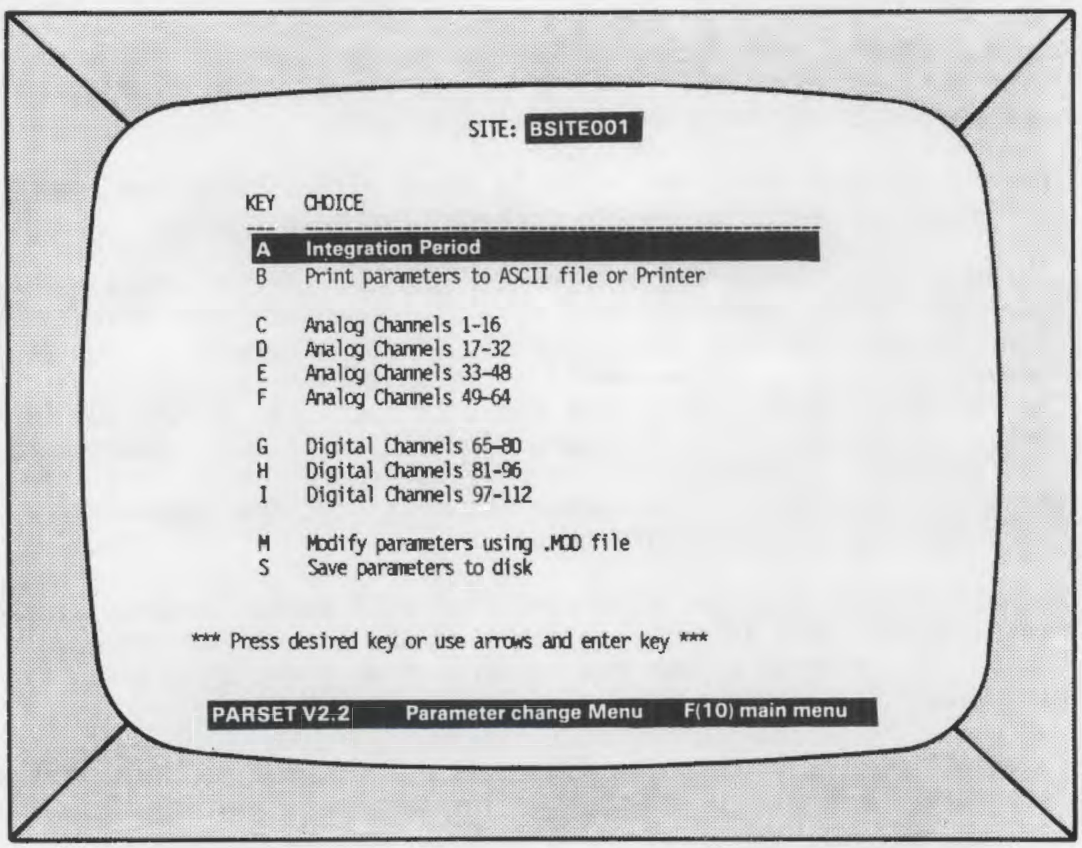
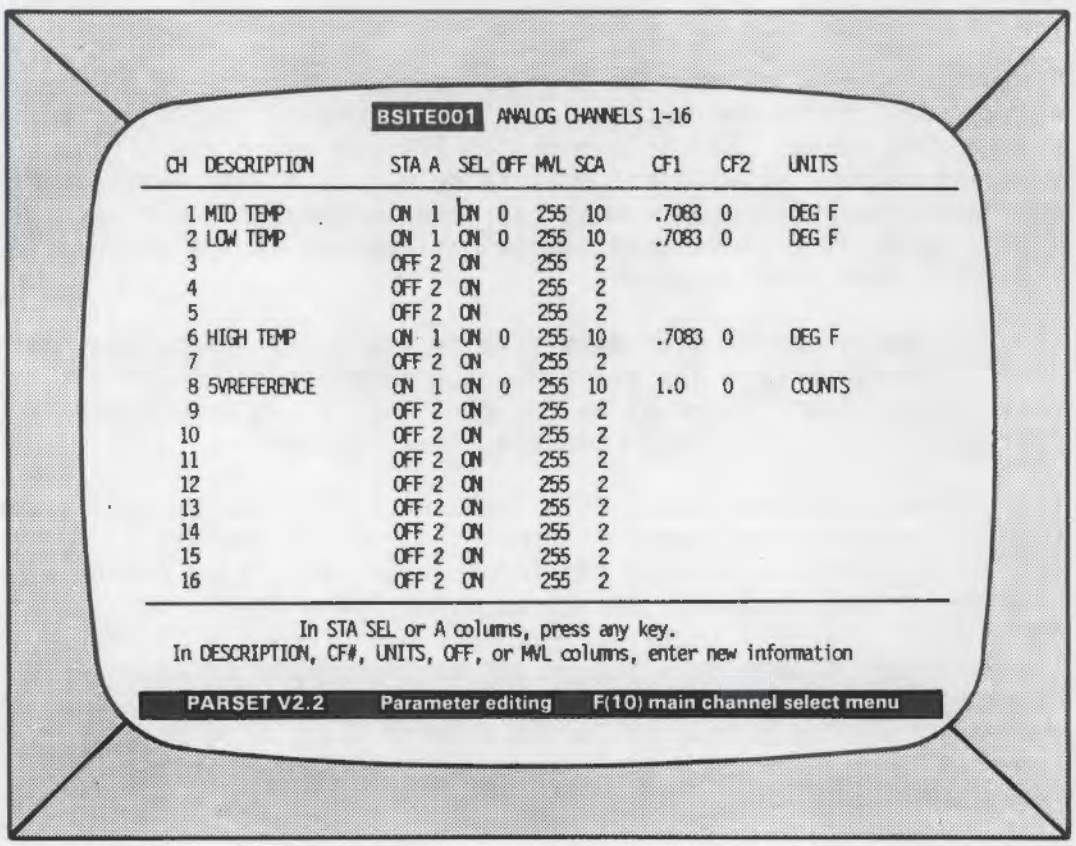


FIGURE 7. PARSET Parameter Set Editor

for the logger. The integration period is the time interval between records. This value can be any value from 1 minute to 18 hours.

Options **C** through **I** let you enter measurement plan information with a screen editor. Typing **C** will bring up a screen similar to that shown in Figure 8. You can make entries with this screen editor just as you would use a spreadsheet program. Position the cursor to the correct position and type the information. When you are done, you can type a carriage return or one of the arrow keys to make the entry. Some of the entries in the measurement plan are simple choices that you can toggle by typing any key.

Note the similarities between the blank measurement plan and the screen editor. The first entry is the logger's channel number (1 through 112), which is determined by the channel group you choose. Next is an 11 character descriptor to accompany that particular measurement. Care should be taken to make this descriptor as informative as possible. Next are three toggled values: the status (STA), the accuracy (A), and the select (SEL). The status column is a set of flags that activate data collection and must be toggled to **ON** for active channels.


BSITE001 ANALOG CHANNELS 1-16

CH	DESCRIPTION	STA	A	SEL	OFF	MVL	SCA	CF1	CF2	UNITS
1	MID TEMP	ON	1	ON	0	255	10	.7083	0	DEG F
2	LOW TEMP	ON	1	ON	0	255	10	.7083	0	DEG F
3		OFF	2	ON		255	2			
4		OFF	2	ON		255	2			
5		OFF	2	ON		255	2			
6	HIGH TEMP	ON	1	ON	0	255	10	.7083	0	DEG F
7		OFF	2	ON		255	2			
8	5VREFERENCE	ON	1	ON	0	255	10	1.0	0	COUNTS
9		OFF	2	ON		255	2			
10		OFF	2	ON		255	2			
11		OFF	2	ON		255	2			
12		OFF	2	ON		255	2			
13		OFF	2	ON		255	2			
14		OFF	2	ON		255	2			
15		OFF	2	ON		255	2			
16		OFF	2	ON		255	2			

In STA SEL or A columns, press any key.
In DESCRIPTION, CF#, UNITS, OFF, or MVL columns, enter new information

PARSET V2.2 Parameter editing F(10) main channel select menu

FIGURE 8. PARSET Sample Parameter Set Edit Screen

The accuracy can be either 1 or 2. This is the number of bytes saved by the processor in each record for that channel. This should always be toggled to 1, except in special circumstances (Appendix C).

The select column allows you to collect a subset of the sensor readings from a logger. This allows several different users to collect a sub-set of the channel data from a network of FDAS units. This column defaults to ON.

The **OFF** column stands for offset counts. This lets you set the number of counts offset for each channel. This counts offset is normally reserved for energy measurements using PNL-designed watt metering circuits or other instruments with a built-in offset. The **MVL** column stands for maximum value expected. The units of this number are counts for analog channels (1 through 64) and Hz for digital channels (65 through 112). In special cases where low values are expected from the instruments, this value may be set lower to adjust the scaling of the data.

The next column is the scaling factor. This quantity is calculated automatically from the accuracy, maximum value, and integration period. Appendix C includes a discussion of the logger's data reduction method and its use of the scaling factors.

The next two entries are the data conversion factors. All conversions are linear between counts recorded by the logger and the engineering value. **CF1** is the multiplicative value and **CF2** is the additive value. We will not go into detail on how to derive these constants, but will assume that they are already on the measurement plan. Appendix C shows some sample calibration factor calculations and conversion considerations.

The last entry in the site data file is the units of the engineering data. Six characters are allowed. The comments column in the measurement plan (Figure 5) is for field notes and does not have a corresponding entry in the computer parameter set.

The accuracy (A) and status (STA) columns in the screen editor can be toggled by typing any key while the cursor is on the value. Other entries require you to type the information, then type either a carriage return or one of the arrow keys. When the information for a group of 16 channels looks correct, you can press **F10** to return to the Parameter Change Menu and select another group of 16 channels. After you have made all of your entries, be sure to scan all of the channel groups to make sure you have typed everything correctly. You can scan your entries by pressing **C, F10, D, F10,** at the main Parameter Change menu.

When you think all of the entries are correct, you can type S to save the parameter set. Again, you will be prompted for the site number and parameter set code.

Option B in the main parameter change menu lets you print a copy of the parameter set to a printer or a file. Keeping a paper copy of the parameter sets for all site may enhance the organization of your network.

Typing **F10** at the Parameter Change Menu will cause you to return to the PARSET main menu. Upon exit, the computer will ask if you have saved your work, just in case you forgot to save the parameter set to disk. Simply answer **N** to the first question if you have already saved the parameter set.

3.3 UPDATING OR EDITING SITE FILES/PARAMETER SETS

If you need to edit a particular parameter set, use option **D** in the PARSET main menu to specify the site/parameter set and then use option **A**. The data will be retrieved from the disk, and you can edit with the screen editor described in the preceding section. When you use the **S** option to save your changes, be sure to type the same site number and parameter set code if you are EDITING, or the same site number and the next parameter set code if you are UPDATING a logger's measurement plan.

Here are the rules about parameter set updates and changes. The changing of parameter set can be broken into two categories: those that affect data collection in the remote logger (parameter update), and those that do not affect data collection (parameter edit). Refer to the list below to see the category in which parameter set changes fall:

- changes that affect data collection and **REQUIRE** a new parameter set code (updating):
  - changing the integration period
  - changing the number of active channels (i.e., adding sensors)
  - changing the bytes of resolution for any channel
  - altering the scaling factor by selecting a new maximum value
- changes that do not affect data collection and **DO NOT REQUIRE** a change in the parameter set code (editing):
  - changing the channel descriptors
  - changing the CF1 values
  - changing the CF2 values
  - changing the counts offset for any channel
  - changing the units.

If changes are required that affect data collection, you will have to go through several steps to ensure a smooth transition to the new parameters in the logger. Refer to Section 3.5 for details on changing a logger's parameters.

3.4 LOADING THE PARAMETERS TO THE LOGGER

Once you have carefully entered the measurement plan with the PARSET program, it is time to load those parameters to the remote site so that data collection can begin. Remember that the parameters you have entered are used by the remote site to decide what channels are turned on and how to scale the data that is collected from those channels.

Use option **D** in PARSET to specify the site/parameter set that you want to load. Next, use option **C** in the main menu to enter PARSET's dumb terminal mode. You will use this option for all manual communications with the remote loggers. As its name implies, the **C** option simply puts you in direct communication with your computer's modem. You must give your modem the dialing commands to establish communications.

For Hayes-compatible modems with touchtone service, the dialing command is of the form

AT DT phone number

Refer to the manual for your modem for more information on dialing. The PARSET dumb terminal has an autodial feature that you can use for sites that are already in your network data collection queue.

Once you have connected to the logger, you can issue logger commands. To verify communications, you can type a **#** character (shift 3) to get an ASCII transmission of the parameter block in the logger memory. Figure 9 is a sample screen showing an ASCII dump of a logger's parameter block. You can stop transmission with a carriage return.

After you establish communications with the logger using the dumb terminal, you can exit to the main PARSET menu using the **F10** key. Note that this will return you to the main menu without disconnecting you from the remote site. You can now execute option **B**, which is the manual FDAS control routine. This option will read and display the date and time in the remote logger's clock and give you several choices. Figure 10 shows the manual logger control screen; the manual control options available from this screen are:

- P sends the parameters to the logger, checks and corrects the transmission if necessary, and then clears the logger's memory.

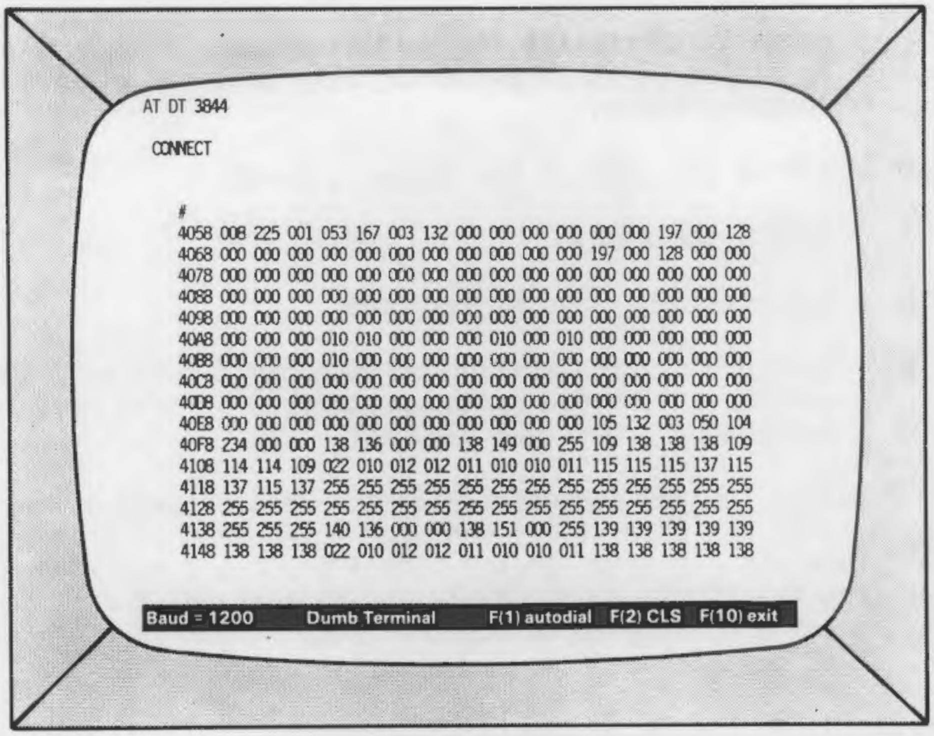


FIGURE 9. PARSET Dumb Terminal

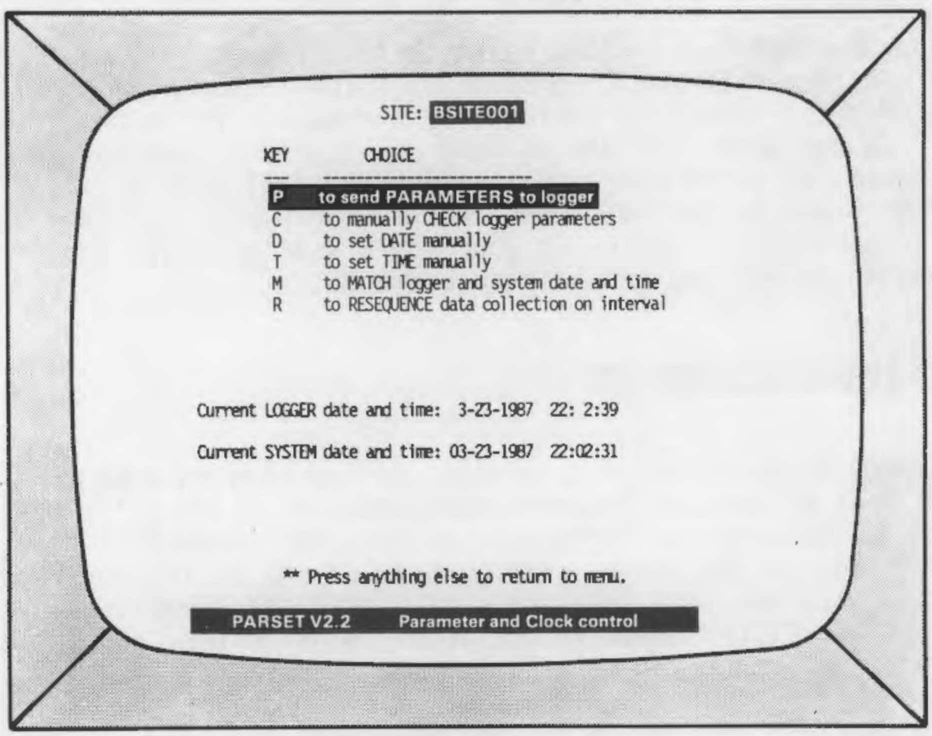


FIGURE 10. PARSET Manual Logger Control Screen

- **C** manually checks the logger's parameters against the specified disk parameters. This does not alter the logger's memory.

- **D** resets the date of the logger's clock.

- **T** resets the time.

- **M** matches logger and system time.

- **R** re-sequences the writing of records so that records are written at the end of the prescribed interval (for example, every hour or every 5 minutes).

Use the **P** option to load parameters first. Then use the **D**, **T**, and **R** options to reset the date, time, and sequencing in that order. You can substitute the M command (match times) for the D and T commands. Remember that the clock in your PC is the time standard for the entire network. After executing these commands, the logger will be initialized and will start collecting data.

An alternative method is to use the **P** option only. This will load the parameter to the logger and clear the logger's memory. Section 4.4 includes a discussion of how you can let the COLLECT program automatically handle the logger clock and record sequencing.

When you are finished loading parameters to the logger, exit the B option and return to the dumb terminal. The logger will still be connected. Disconnect by typing **+++**, and wait for the modem to respond "OK." You can then type **ATH** to hang up. Another easy way to hang up is to press F10 twice when you are in the PARSET dumb terminal. This will return you to the main menu, exit the program, and hang up the phone automatically. Refer to your modem user's manual for more information on dialing and hang-up commands.

3.5 CHANGING A LOGGER'S PARAMETERS

If a sensor is to be added to a site, or you plan to make some other changes that affect the logger's parameters (Section 3.3), you will need to follow a careful procedure to make the transition from one parameter set to the other. The purpose of the following procedure is to minimize the data loss associated with measurement plan changes. The steps are listed below in the order in which they should be executed.

- Use PARSET to update the parameter set for the site (Section 3.3).

- Use the COLLECT program to manually acquire all of the data for the site under the old parameter set (Section 4).

- Use the PARSET program to call the site and load the new parameters to the logger (Section 3.4).

- Use the QUE program to change the parameter set code in the SITE.DRV file (Section 4.2).

4.0 PCDAS OPERATIONS

We discussed the entering of measurement plans in some detail in Section 3. Section 4 covers the operation of the PCDAS data collection system. In general, this data collection system can be seen as a close interaction between programs and files. Each data or driver file plays an important role in the data collection process; an error in these files will cause errors in data collection. For this reason, it is important for you to understand how the entries in the various files control data collection and conversion.

The information in this section will be presented in general terms for those interested in running the programs. If you need more technical information on the programs or file structures, you can refer to Section 6.

4.1 PRELIMINARIES

Assume that you have the measurement plan(s) for some remote site(s). Also assume that you have encoded the measurement plan(s) with the PARSET program (Section 3). This means that you have numbered the sites and have assigned single-letter parameter set codes to identify a particular configuration (measurement plan) for each site. The question now is: how do I set up the automatic collection for a site? The answer to that question is that you simply make an entry in a driver file using the QUE program. But more on this later. Right now, you should be made aware of some preliminary operational concerns.

4.1.1 The Network Time Standard

The clock in your PC is the reference clock for the whole network. The data collection routine automatically adjusts the logger clocks to match the clock on your PC.

You are responsible for maintaining the accuracy of the network's time standard. If the collection routine is run in the automatic mode for an extended period, you still must examine and correct the time standard, if necessary. If the software is run in the manual mode, then you must examine the clock before each run. The clock on a PC can be examined and adjusted using the DOS "TIME" command. You should use a known standard time for correcting the computer's clock. The National Bureau of Standards Greenwich Mean Time is available (303-499-7111) just like local time.

If the write-protect switch is "ON" for all of the entries in the QUE program, then PCDAS software will not control the logger clocks and you need not worry about maintaining a time standard. In this case, it is assumed that some other collection computer is maintaining the network time standard.

If your PC doesn't have a battery-backed clock, you may want to consider buying one. It is a real convenience when using this system.

4.1.2 File Protection

The most critical files in the PCDAS collection system are the files with the extensions: ".DAT" and ".PAR". These files reside in the SITEDAT subdirectory. Files with a ".DRV" extension are also critical. All of the other files in the system (excluding binary data files) can be reconstructed easily or are reconstructed automatically if lost.

The files listed above represent all of the measurement plan encoding effort and the network setup effort. You should always keep a backup copy of these files, as well as backup copies of binary data.

4.1.3 Data on a Hard Disk

If the PCDAS system is run from a hard disk, you should back up the ".DAT" and ".PAR" files, the SITE.DRV, and the binary data base. The backup should be made to floppy disk or some other removable medium. This will protect against data loss due to failures of the hard disk.

4.1.4 General Binary Data Back Up

Backup of the binary data should be performed incrementally with only the most recent data files copied to backup. All data is stored in monthly subdirectories for easy directory backup. If you are using DOS Version 3.0 you can use the BACKUP command to do this.

4.1.5 Paper Trail

There should be a paper copy of the measurement plan as well as other installation documentation. Each site should have its own folder. A separate folder may be kept to file notes about data collection and to record error conditions in the network.

4.2 ADDING A SITE TO THE COLLECTION QUEUE

Before a site can enter the collection queue, it must meet some basic criteria. These criteria are simple, but they are absolutely necessary before data can be collected.

- The site must be installed. This means that the logger at the remote site and the logger's modem must be running. Also, a working phone line with a known phone number must be connected to the logger's modem.

- There must be a complete measurement plan. This document must accurately state what sensors are attached to the FDAS, the communications baud rate and phone number, and how to reduce the signals to data with engineering units (CF1 and CF2 values).

- The measurement plan must be entered with the PARSET program. You must ensure that the encoded measurement plan matches the paper copy. All scaling factors must be adjusted correctly, and the correct channels activated.

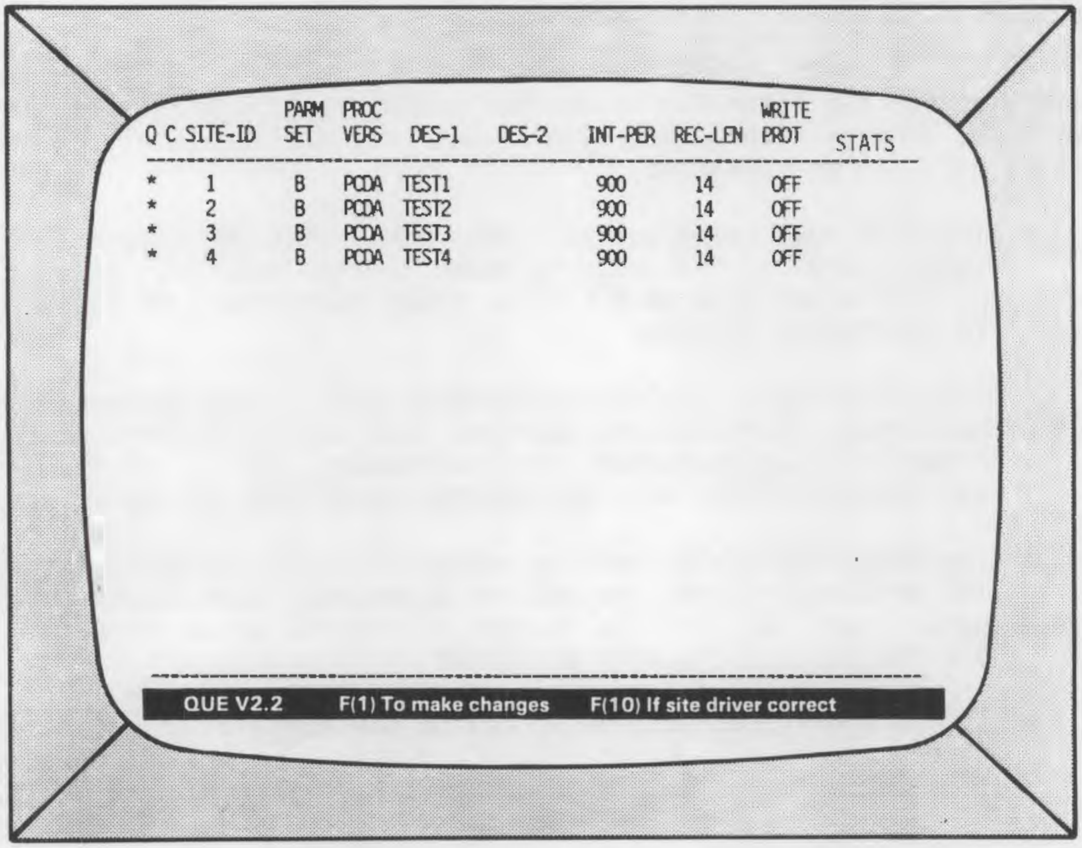
- The parameter set must be loaded to the logger. This can be done automatically by the COLLECT routine, or you can perform the loading manually with the PARSET routine (see Section 3).

If the site that you want to add satisfies all of the above criteria, then you are ready to enter the site into automatic polling. Do this by running the QUE program. Type **QUE** at the DOS prompt in the PCDAS subdirectory. If this is the first site to be put into automatic polling, then there will be no previous entries in the queue.

There will be a number of headings at the top of the screen and some function key definitions at the bottom. The **F1** key puts you into the screen editor mode to make changes. Note that the cursor starts in the upper left-hand corner. Also, the function key definitions will change after typing **F1**. Figure 11 shows what the first screen of QUE might look like if there were four sites in the collection queue.

If you are making the first entry in this file, you will need to know what each heading means. From left to right, the entries are:

- Q - a single-character entry that shows if a site meets the data collection criteria outlined above--This entry is toggled with any key. The "*" character indicates that the site meets the criteria.



Q	C	SITE-ID	PARM SET	PROC VERS	DES-1	DES-2	INT-PER	REC-LEN	WRITE PROT	STATS
*		1	B	PCDA	TEST1		900	14	OFF	
*		2	B	PCDA	TEST2		900	14	OFF	
*		3	B	PCDA	TEST3		900	14	OFF	
*		4	B	PCDA	TEST4		900	14	OFF	

QUE V2.2   F(1) To make changes   F(10) If site driver correct

FIGURE 11. QUE First Screen

- C - another single-character entry that can be toggled with any key--It indicates if data is to be collected on the next COLLECT run. This column is used in the manual mode only. If this column contains the "*" character, then data collection will be performed on the next COLLECT run. This entry also controls the calculation of the values in the REC- LEN and INT-PER columns.

- PARM SET - the single letter parameter set code that you used to save the ".DAT" and ".PAR" files

- PROC VERS - the four-letter processor version code of the microprocessor in the field--Version 2.2 of PCDAS can use PCDA, PCDB, PCDC, and SOPp processors.

- DES-1 - a seven-character field provided for site description

- DES-2 - a second seven-character field provided for site description

- INT-PER - the integration period in seconds--This a derived quantity

- REC-LEN - the number of bytes stored in the binary record from the logger--Because this is a derived quantity, the cursor in the QUE program will not point to this entry.

- WRITE PROTECT - the write-protect switch--The default is "OFF", meaning that your computer will control the network clocks and parameter blocks. This can be toggled to the "ON" position with any key.

- STATS - This entry controls the automatic generation of statistical information for all channels. Choices are daily, weekly, monthly, or any combination of the three. Statistics include running average, maximum, minimum, time of maximum, and time of minimum. Default is "OFF".

These are all of the entries in the first screen of the QUE program. To enter a site in the collection queue, you simply have to type in the site number, parameter set, processor version, and phone number. You will also have to toggle the Q column to the "*" character by typing any key while the cursor is in that column.

The second seven-character descriptor field (DES-2) can be used to alter the time stamps for CONVERTED data from a logger. The time stamps on the data in the BINARY archive files are not changed from the original time stamps written by the logger. The time change specification in DES-2 allows you to change the time stamps on the CONVERTED data from the TRANS or ACCESS programs. The format for the time change specification is:

{N}

where N is the time change in hours. This feature accounts for daylight savings time changes in April and October.

A typical application for use of the time change feature is if your network of FDAS units extended across time zones. In this case you could set your data collection computer to a standard time [say, Greenwich Mean Time (GMT)] and type in the difference between GMT and local standard time in the DES-2 column.

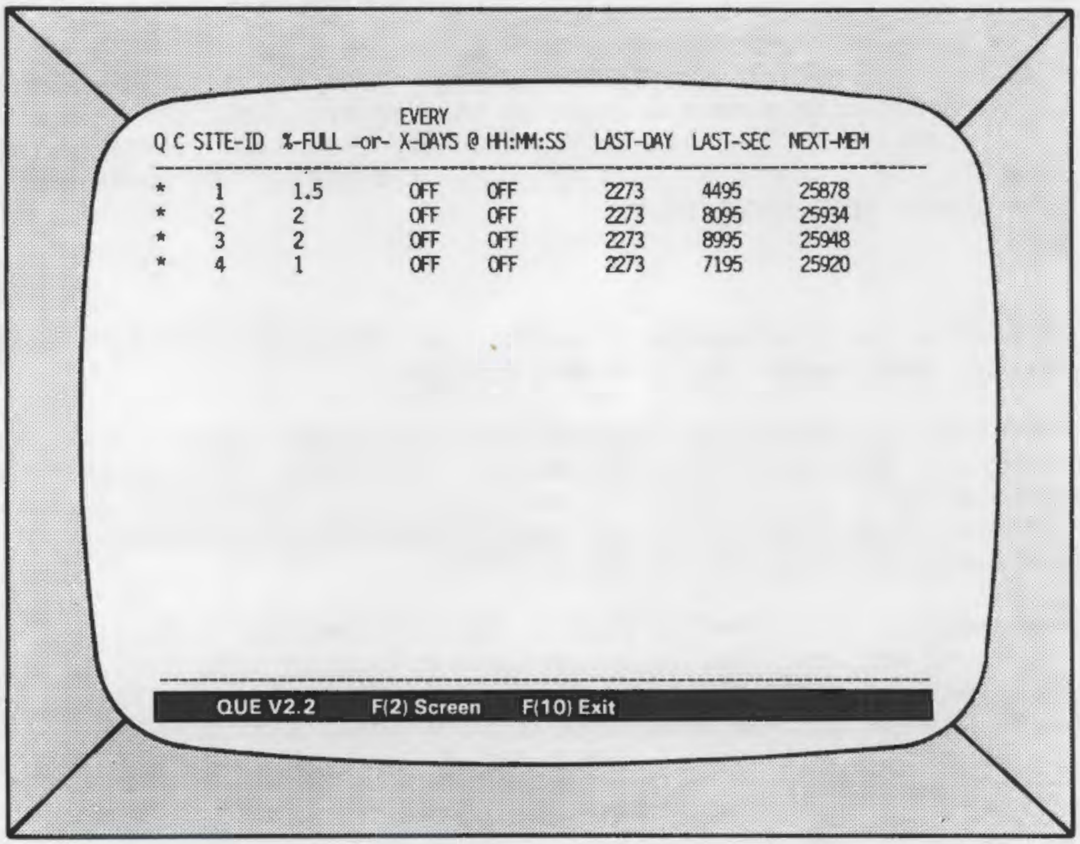
For example, if you had an FDAS in New York and one in Seattle, you could set both machines (and your data collection computer) to GMT and type {-5} for the FDAS in New York and {-8} for the FDAS in Seattle. These numbers are the difference between local standard time and GMT. All converted data would be reported in local time with corrections for daylight savings.

If all of the FDAS units in you network are in the same time zone, then you will probably want to keep the network on local time and make two time changes per year to account for daylight savings time.

There are two more screens of information in the QUE program. Most of this information is assigned default values whenever a new site number is entered.

All three screens of the QUE program can be toggled in sequence using the F2 key. After you have entered the information in the first screen, press F2 to see the second screen. Figure 12 shows the second screen for a sample four-site network. The first three columns in this screen are printed for information only and cannot be edited.

Data from each site can be collected on a percent full basis or at a fixed interval of days at a particular time of day. If you select the percent full option, then the interval columns are set to the OFF position automatically. Likewise, if you select the interval option for data scheduling, then the percent full column will be set to OFF. The default scheduling criterion is set to 50 percent of the logger's memory. You will note that the cursor starts in the %-FULL column. The meaning of each entry and the range of values that can be assigned to each entry are as follows:



Q C	SITE-ID	%-FULL -or-	EVERY X-DAYS @	HH:MM:SS	LAST-DAY	LAST-SEC	NEXT-MEM
*	1	1.5	OFF	OFF	2273	4495	25878
*	2	2	OFF	OFF	2273	8095	25934
*	3	2	OFF	OFF	2273	8995	25948
*	4	1	OFF	OFF	2273	7195	25920

FIGURE 12. QUE Second Screen, Collection Schedule Control

- %-FULL - This field contains either a numerical value (from 0 to 100) or the word "OFF." The numerical value indicates, in percent, how full the logger will be when data collection is scheduled. "OFF" indicates the use of the set interval method of collection scheduling.

- EVERY X-DAYS - This field indicates the number of days between collection events.

- @ HH:MM:SS - This entry is the time of day that data will be collected at the end of the X-DAYS interval.

- LAST DAY - This is the day of the decade on the time stamp of the most recent record in the binary data base. This entry is written by the collection program and is presented in this screen for your information.

- LAST SEC - This is the elapsed seconds of the day on the time stamp of the last record in the binary data base. It is written by the collection routine and is presented in the first screen for your information.

- NEXT MEM - This is the address (in decimal) in the logger's memory where the most recent record was collected. This entry is written by the COLLECT program along with the LAST DAY and LAST SEC entries.

The third screen of information can be edited by typing the F2 key while in the second screen. Pressing F2 while in the third screen will bring you back to the first screen. The third screen, as shown in Figure 13, is devoted to communications. The description of each column in the third screen are:

- BAUD RATE - This is the baud rate for communications. The default is 1200 baud. The baud rate should be shown on the measurement plan and is part of the installation configuration. The operator cannot switch between baud rates without visiting the site.

- PHONE-NUMBER - This is the phone number of the site's modem used in the ATDT dial-out sequence. Sixteen characters are allowed.

- WAIT CARRIER - This is the number of seconds that the data collection modem will wait for a carrier after dialing. This can be set to any value; the default value is 60 seconds.

- NOISE SEC - This is one of two parameters used in the evaluation of phone lines. It is the number of seconds to wait during the phone noise evaluation routine. The default value is 20 seconds.

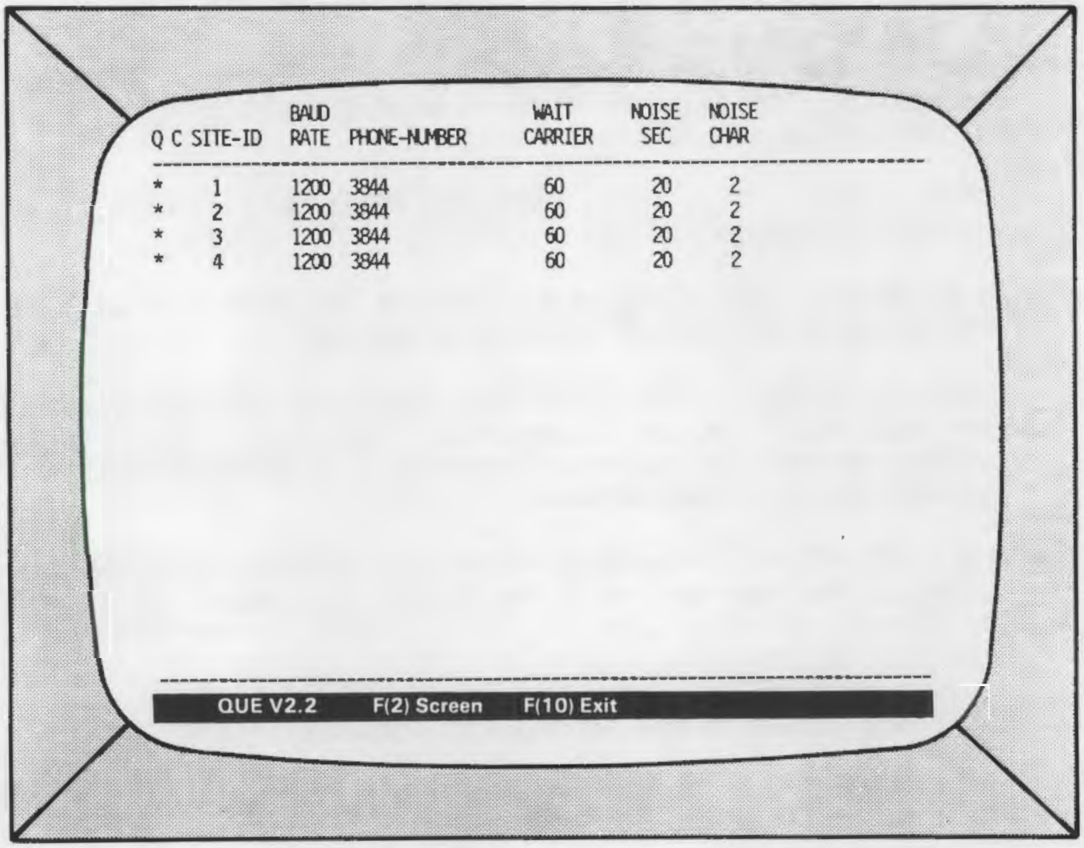


FIGURE 13. QUE Third Screen, Communication Control

- NOISE CHAR - This is the threshold number of characters in the specified number of seconds. If the actual number of random characters exceeds this value, the phone line will not pass the quality check.

This covers all of the entries to the SITE.DRV file that can be made using the QUE program. All entries in the SITE.DRV file have defaults assigned except the Q, C, site number, parameter set, processor version, and phone number. After these entries are made, the site is ready for data collection. The operator can now collect data from the site either manually or automatically, so long as the measurement parameters remain constant.

Whenever the parameters change (see Section 3.5), you will have to go through the steps described above to start collecting data for the new set of measurement parameters. Refer to Section 3 for more information on how to alter parameter sets.

The QUE program will automatically order the sites sequentially so you do not have to make entries in order. If you want to temporarily remove a site from the collection queue, simply toggle the "*" character in the Q column to a blank. If you want to permanently remove a site from the queue, then type a space in the site number column.

4.3 AUTOMATIC STATISTICS GENERATION

If you wish to routinely compile simple statistics on all of the channel data for a logger, then you can enable the statistics generator with the QUE program. If you want to manipulate data in other ways (for example, aggregations, folding, channel subset selection), then you may want to use the ACCESS program, which has more general data reduction and manipulation routines.

The statistics that can be compiled automatically are

- channel average
- maximum
- time of maximum reading
- minimum
- time of minimum reading.

These simple statistics were designed to be part of an electrical utility billing system. The statistics can be accumulated over days, weeks, months, or any combination of these three.

All statistics files are stored under the SITEnnn directory in a subdirectory called STATS. This provides a separate location for statistics data for each site. The naming convention for the binary statistics file is

NNNNXnnn.D	(daily)
NNNNXnnn.W	(weekly)
NNNNXnnn.M	(monthly)

where NNNN is the day of the decade, X is the parameter set code, and nnn is the site number.

Weekly statistics are compiled Sunday through Saturday, with NNNN being the Sunday day of the decade. For monthly statistics files, the NNNN is the day of the decade for the first day is the month.

A utility program called STATS lets you create the statistics files manually after the binary channel data has been collected. The STATS program prompts you for the site number, parameter set code, the beginning and end day of the decade, and the statistics option (D,W,M). This program uses the binary data stored under the SITEnnn directory to create the desired statistics files in the given time frame. This utility program gives you the choice of automatic statistics generation for routine tasks (QUE program) or manual operations for occasional analysis tasks (STATS program).

The TRANS program has several output options that let you convert the compiled binary statistics files into usable ASCII data files. You can convert daily, weekly, or monthly files over any time frame. For example, you could convert monthly statistics for a year and output the results in a single file. You could also convert daily statistics files for a month and output the results in a single file.

In addition to the D, W, and M output specifiers, there is the single month (SM) output option. This output format is designed to output one month's statistics for all sites in a single file. The SM output option was designed for utility billing. See Section 5.1 for more information on the TRANS program.

4.4 COLLECTING DATA FROM THE NETWORK

This section describes the operation of the central data collection routines. In this discussion, it is assumed that the measurement plans have been entered correctly and that the site has been entered into the collection queue using the QUE program. Once these steps have been completed, the data collection can proceed automatically or manually.

4.4.1 Automatic Data Collection

The PCDAS program is the general purpose data collection scheduler/ controller and the general purpose access program. It is possible to run almost all of the main programs using this one PCDAS program. To initiate automatic data collection for a network of sites that have encoded measurement plans and that have been entered into the collection queue using the QUE program, you simply execute the PCDAS program by typing PCDAS in the data collection root directory.

If you are starting a network or adding a site to an existing network, then the PCDAS program will automatically execute the COLLECT program. This will automatically collect the first data from the site. Subsequent data collection will be scheduled based on the criterion that you entered into the QUE program.

The screen shown in Figure 14 is a sample of what you normally see when the software is running. This single screen contains the information that you need to evaluate the status of the data collection for the network. The top of the screen shows the system date and time. The center portion of the screen shows the next 10 scheduled data collection events organized in chronological order. This list includes the site number, the parameter set code, and the date and time of the next polling event. The lower portion of the screen is reserved for operator messages and for instructions.

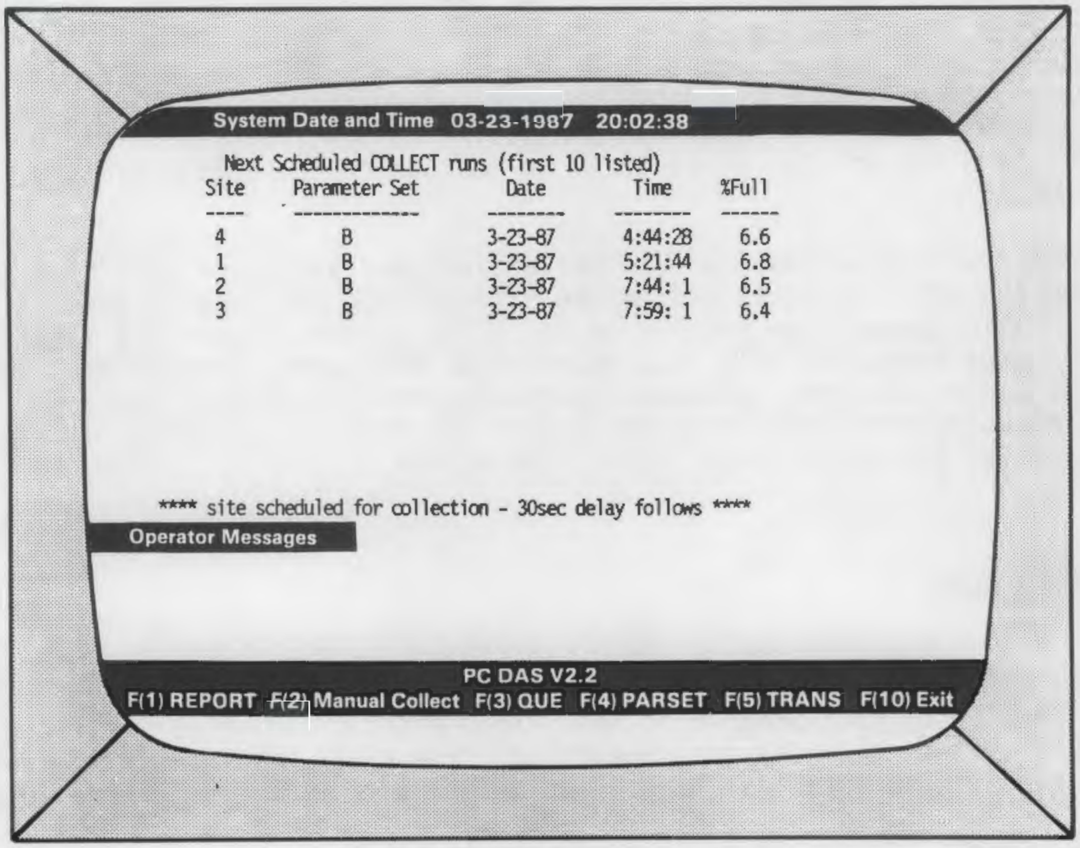


FIGURE 14. PCDAS Main System Screen

The last line of the main screen in the PCDAS program shows the function key definitions. There are function keys that allow you to execute the REPORT, QUE, PARSET, and TRANS programs, as well as a manual data collection function. The last function key F10 exits the PCDAS program and returns to DOS. Whenever you execute one of the programs using this method, the programs execute in exactly the same way, but, when exited, return to the PCDAS program rather than returning to DOS.

While the PCDAS program is running, it is constantly comparing the system clock with the dates and times in the list for polling events. When the computer's clock ticks up to the date and time for the next polling, the COLLECT program runs automatically. Before PCDAS branches to the COLLECT program, it waits for 30 seconds. If there are several sites with polling dates and times that are close together (within 30 seconds), this wait will allow for all of the polling events in the same COLLECT run. You are advised of the 30- second wait period by a message printed below the list of polling events. This message is shown in Figure 14.

When the data has been collected from the sites, the COLLECT program automatically branches back to the PCDAS scheduler/controller program, which updates the collection schedule and waits for the next polling event. In this way, the PCDAS program can automatically collect data from the network over extended periods. You are freed from routine file handling and can concentrate on monitoring the status of the network.

If an error occurs during data collection on a site, you will have to read the error messages before the site will be put back in the automatic queue. The section of the PCDAS screen (Figure 14) labeled "Operator Messages" will show which sites have error conditions. You must clear the error condition by reading the appropriate message file (using the REPORT program) before any further collection will be scheduled for a particular site. The REPORT program is covered in Section 4.5.

4.4.2 COLLECT Runs

When data is to be collected from a site, a common set of routines called the COLLECT program is run. This program is the heart of the PCDAS system. It uses the site driver file (SITE.DRV) and the files in the SITEDAT data base to collect and archive data from the network.

The COLLECT program is fully automatic, requiring no interactive input from the operator. The only interaction with this program is an F10 key that you can use to return to DOS or to PCDAS (in cases like "Oops! I forgot to set the computer's clock!").

There are several outputs from the COLLECT program. The first, and most important, is the binary data organized by site, day, and parameter set. The second output is the operator message data that is also organized by site number, day, and parameter set. The message files are written to the summary subdirectory.

The last thing that the COLLECT program does is write the time stamp of the last record collected into the SITE.DRV file, and write the date of the polling to the SCHEDULE.DAT file. In addition to providing these outputs, the COLLECT program provides screen messages so you can track the progress of the data collection in real time.

The messages that are written to the screen during the data collection are compressed and written to the disk at the same time. You can reconstruct the screen messages at your convenience using the REPORT routine. The procedure for doing this is covered in Section 4.5.1.

4.4.3 Sample COLLECT Run

The purpose of this section is to show you how to read the data collection messages that are created by the COLLECT program. For the most part, the messages are in plain English and are self- explanatory. The example presented here shows the messages for a relatively straightforward polling event. Appendix A contains a complete listing of the possible messages and their meanings for reference.

Figure 15 is an example of what the screen might look like during a polling event. The messages are written to the screen one at a time as the action is performed. Interpreting the messages is quite simple when you observe the following rules:

- The first part of any message is the time (24 hour).
- The second entry in any message is the site number.
- Messages that begin and end with "******" indicate transitions to different functions within COLLECT.

FIGURE 15. COLLECT Sample Polling Event

With these three simple rules in mind, look at Figure 15 and go through a short narrative of this collection event.

The first message shows the start of the COLLECT routine for site 1 on 3-23-1987 at 20:04:19. The routine opens the site driver file (SITE.DRV) and reads the phone number and other important data. The routine dials the site, establishes a connection, and evaluates the noise on the phone line. If the phone line quality check passes, the program verifies communications and compares the logger processor version codes. If all of these communication checks pass, then the program continues with the logger integrity routine.

The logger integrity routine evaluates the condition of the logger. It reads the parameter block from disk and compares it to the parameter block in the logger. If there are errors, then a separate correction routine is run. The results of the comparison, for this example, show zero errors. During the reading of the parameter block, the routine determines the position in the logger's memory where the next record will be written and the time stamp of the most recent record in the logger. After reading this information, the logger integrity routine checks the logger's clock.

In this example, the logger clock was correct. The logger clock is corrected if it differs by more than 60 seconds from the clock in the data collection computer.

In messages that print a clock value, the first number is the day of the decade and the second number is the seconds of the day. Figure 16 is a continuation of the same polling event shown in Figure 15.

After the clock is checked, the COLLECT program drops into the data requirements routine, which determines how much data to collect from the logger. When a site is first put in the collection queue, the collection routine decides that no previous data has been collected and schedules a full dump of the FDAS memory. In Figure 16, 73 records are scheduled for collection. The data requirements routine schedules only the most recent data, thereby eliminating redundant transmission. After data requirements are determined, individual records are transmitted in the data acquisition mode.

The data acquisition routine transmits the records one at a time and writes them to a temporary file in the data destination root directory. To conserve space, the messages that are written to the screen for each record are not written to the operator message file. These messages are intended only to show the progress of the data collection in real time. After the required number of records is collected, the program disconnects from the remote site and continues into the file manager routine where archive files are created.

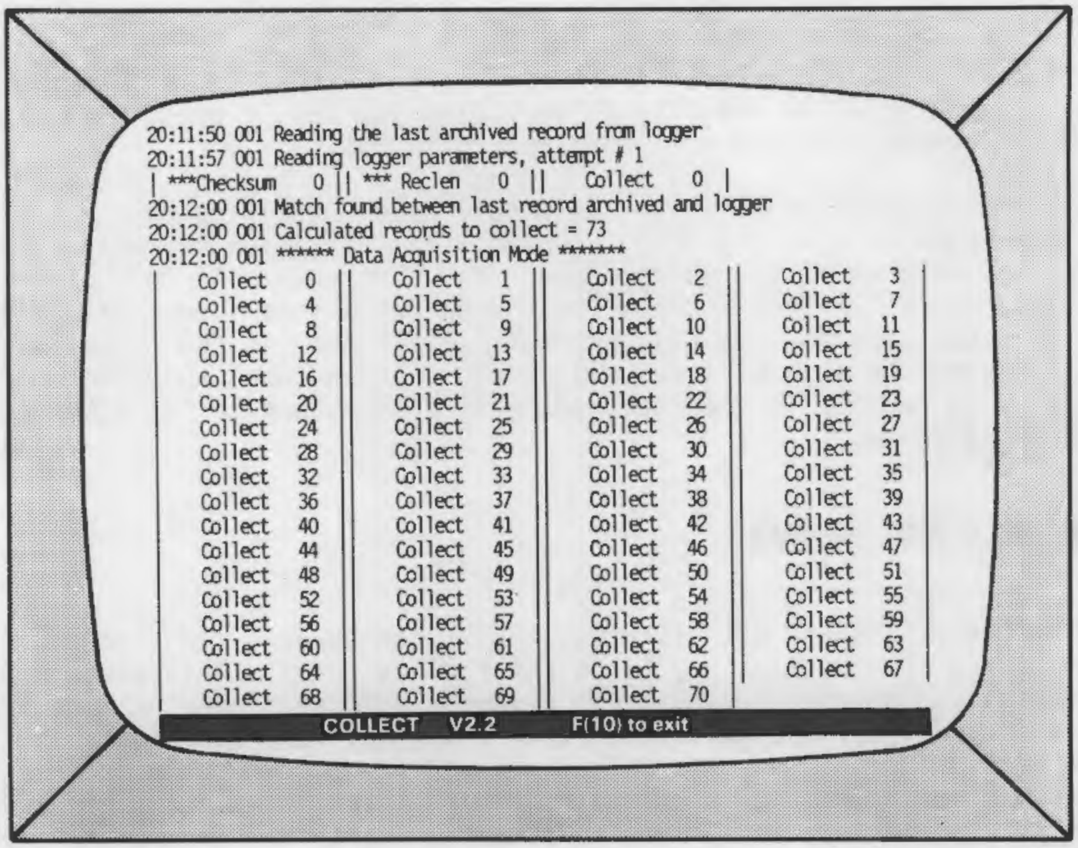


FIGURE 16. COLLECT Sample Polling Event, Continued

If the site had a power failure, then you are shown the number of records that had time stamps reconstructed. The final step in the polling event is updating the site driver file and the schedule file to account for the data collection that just took place.

If statistics generation has been enabled for the site then the statistics routine will run after the site driver file has been updated.

4.4.4 Manual Data Collection

Another way of collecting data from the network is via the manual method. This is a simple means of overriding the PCDAS scheduling so that you can collect data from any or all sites whenever you want. The manual collection method does not disable the normal scheduling that resumes after the manual polling event. There are two ways to manually run the COLLECT program.

- You can manually collect data by running two programs from DOS. First, run the QUE program (Section 4.2) and toggle the C (Collect) column to the "*" characters for all of the target sites. Exit the QUE program to DOS and then type **COLLECT** to actually run the collection routine. When COLLECT completes the data collection for the specified sites, it will update the schedule and site driver files just as it does in automatic operations, and will then return to DOS.

- You can perform manual data collection by typing **F2** in the main menu of the PCDAS program. This function key will cause a branch to the QUE program. You then enter "*" characters in the C (Collect) column for the target sites. Upon exiting the QUE program, you will automatically branch to the COLLECT program and the data will be collected. In this case, when the data collection is complete, you will branch back to the PCDAS program.

4.5 USING THE REPORT PROGRAM

Like the PARSET and QUE programs, REPORT can be executed from DOS or by typing the function key **F1** in the PCDAS program. This program has several useful routines that can help you understand the network status and organize the data collection. Perhaps the most useful routines in REPORT are the ones that allow you to view the messages that are created by the COLLECT program. These routines let you run the PCDAS unattended because any error condition messages can be reconstructed easily.

The most common use of the REPORT program will be to read error condition messages. When you are advised to read the error condition messages for a particular site, you simply press F1 to execute the REPORT program. You are then presented with the menu shown in Figure 17. The options listed will be discussed in the following sections.

4.5.1 Reading Data Collection and Error Messages

The first four options (**A** through **D**) in the REPORT program are different options for reading message files. The only way for you to clear the error condition status flags in the PCDAS screen is to actually read the messages with option **A** or **B**. This should eliminate the possibility of an error condition being missed. The first option (**A**) will list all of the messages for the most recent polling event for the sites with error conditions.

Remember, if an error condition exists for a site, no automatic polling events will be scheduled for that site. This means that the most recent messages are the ones that show the error condition. While the messages are scrolling across the screen, you can use the F2 and **F1** keys to stop and start scrolling. If you desire a hard copy, the

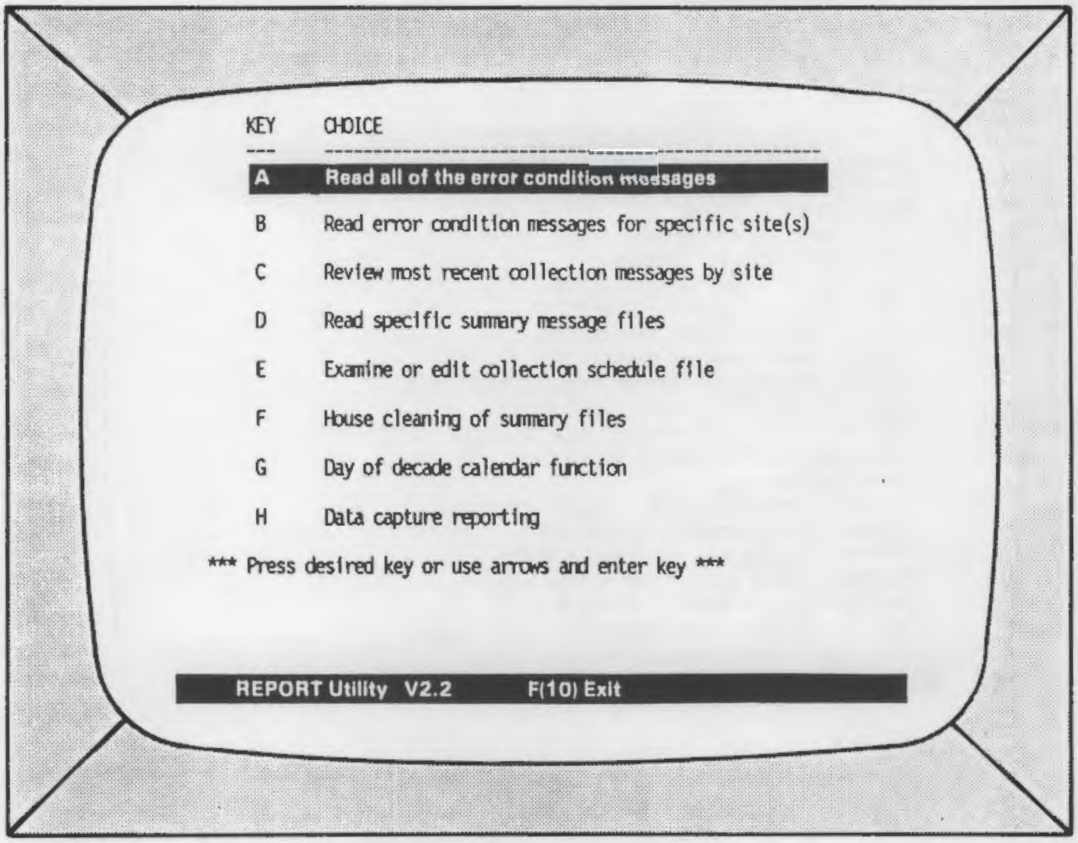


FIGURE 17. REPORT Main Menu

scrolling can be stopped with the **F2** key and the "PrtSc" function on your PC can be used. Pressing **F1** will re-start the scrolling of messages after the screen has been dumped to the printer.

The other three options that print messages use the same scrolling method. In the screen listings, error conditions are highlighted by some ">>" characters. Figure 18 shows an example of the message listing from the REPORT program. In this example, Site 1 had a polling event on 3-23-87. Note that the messages are lacking the exact time and the site number prefixes as in Figure 15. However, the site number can be seen in reverse video at the bottom of the screen, and the time of the polling event can be read from the first line in the listing.

The second option (B) in the REPORT program allows you to read error condition messages site-by-site. When you execute this option, you are prompted to type the site numbers in which you are interested. This prompt is common to all of the routines in REPORT that act site-by-site.

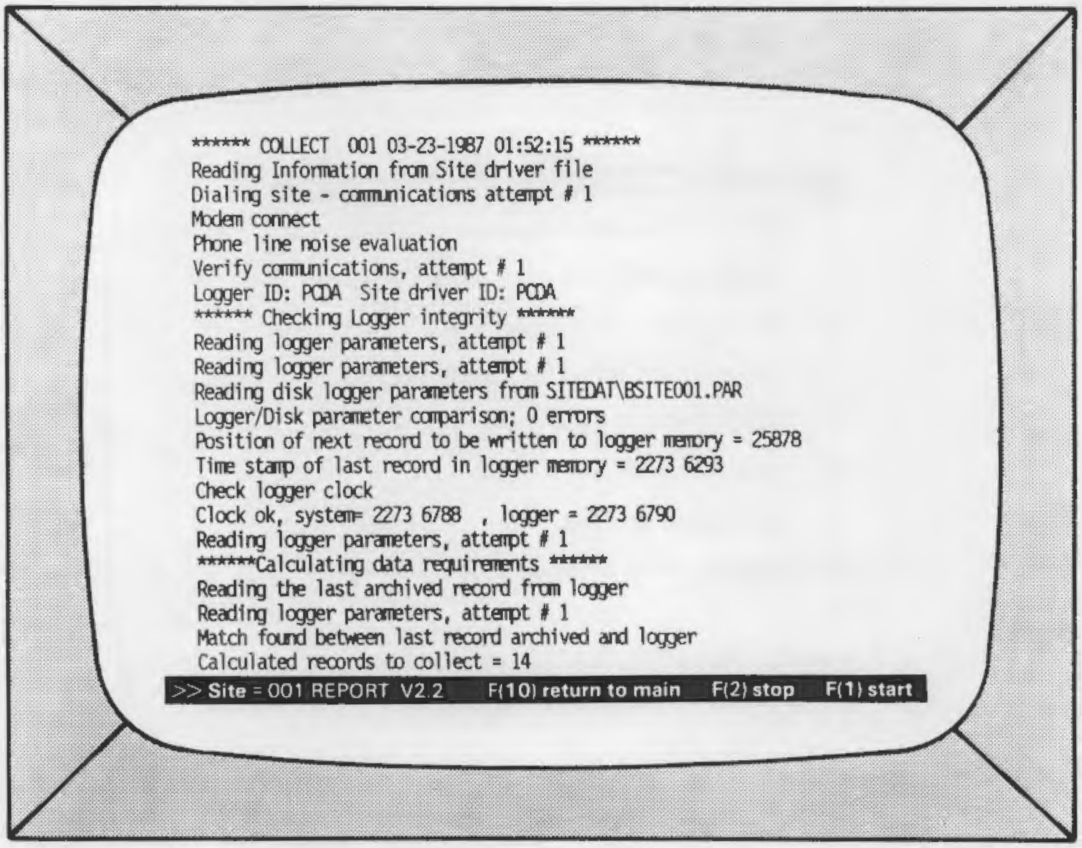


FIGURE 18. REPORT Operator Message Reconstruction

The third option (C) for reading messages allows you to read the most recent collection messages regardless of whether there is an error condition or not. This routine reconstructs the last polling event for any site or sites.

The fourth option (D) for reading messages allows you to reconstruct any specific message file. This means that you can reconstruct old polling events. In this option, you are prompted for the site number, parameter set code, and the day of the decade on which the polling took place.

In any of the four options listed above, you return to the main REPORT menu after all of the messages have been listed.

Reading data collection messages and error condition messages is an important function of the REPORT program, but it is only one of its many uses.

4.5.2 Altering the Collection Schedule

One of the most convenient features of the REPORT program is a routine that allows you to change the collection schedule. When the E option is selected, you are prompted for the sites of interest. Figure 19 shows the information presented for each of the specified sites. If more than 10 sites are specified, you can edit their collection schedule in groups of 10. At the prompt shown in Figure 19, you could type a site number whose next collection date is to be modified, or type "**all**" to change the next collection date for all sites in the network. If there are 10 or fewer sites to view, then the carriage return key will get you back to the main REPORT menu.

After you specify the site(s), you will be prompted for the date and time for the next polling event. When you have typed this information, you will see the change in the collection schedule shown on the screen. The most important item to watch is the "Percent Full" column. This will change when you make your entry. Figure 20 shows the prompts for the new collection date and time.

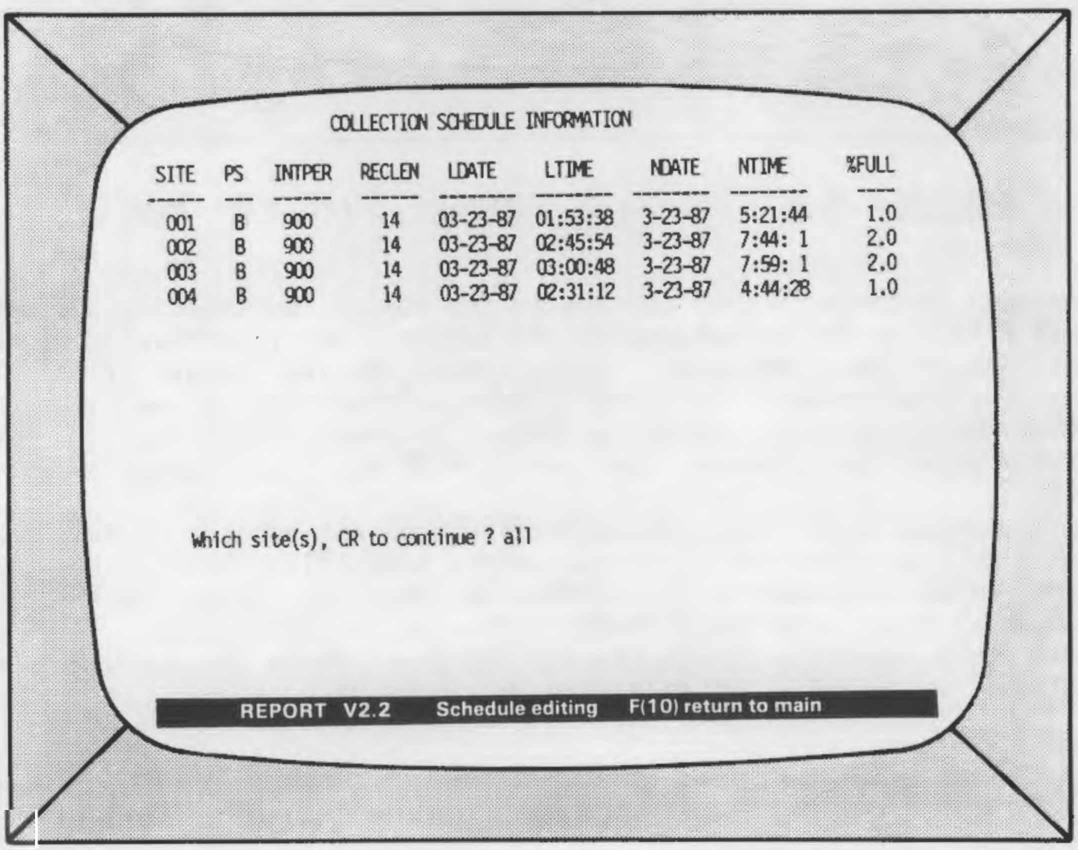


FIGURE 19. REPORT Collection Schedule Editor

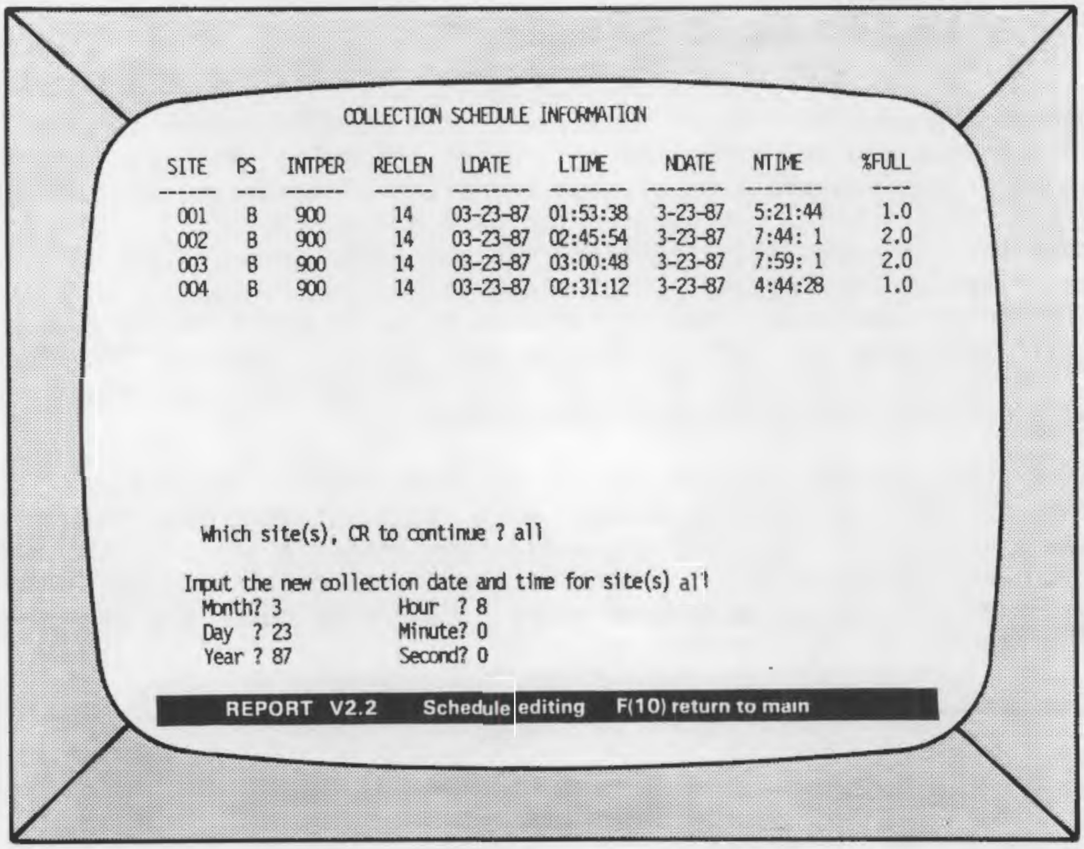


FIGURE 20. REPORT Collection Schedule Editor, Continued

You can continue to make changes to the collection schedule and watch the effect on the percentage of the logger's memory utilized. When all changes have been made, simply type a carriage return at the prompt for the site number. One final prompt will ask you if you want to save the changes that have been made. If your response is "Y", then the SCHEDULE.DAT file will be updated with the new information.

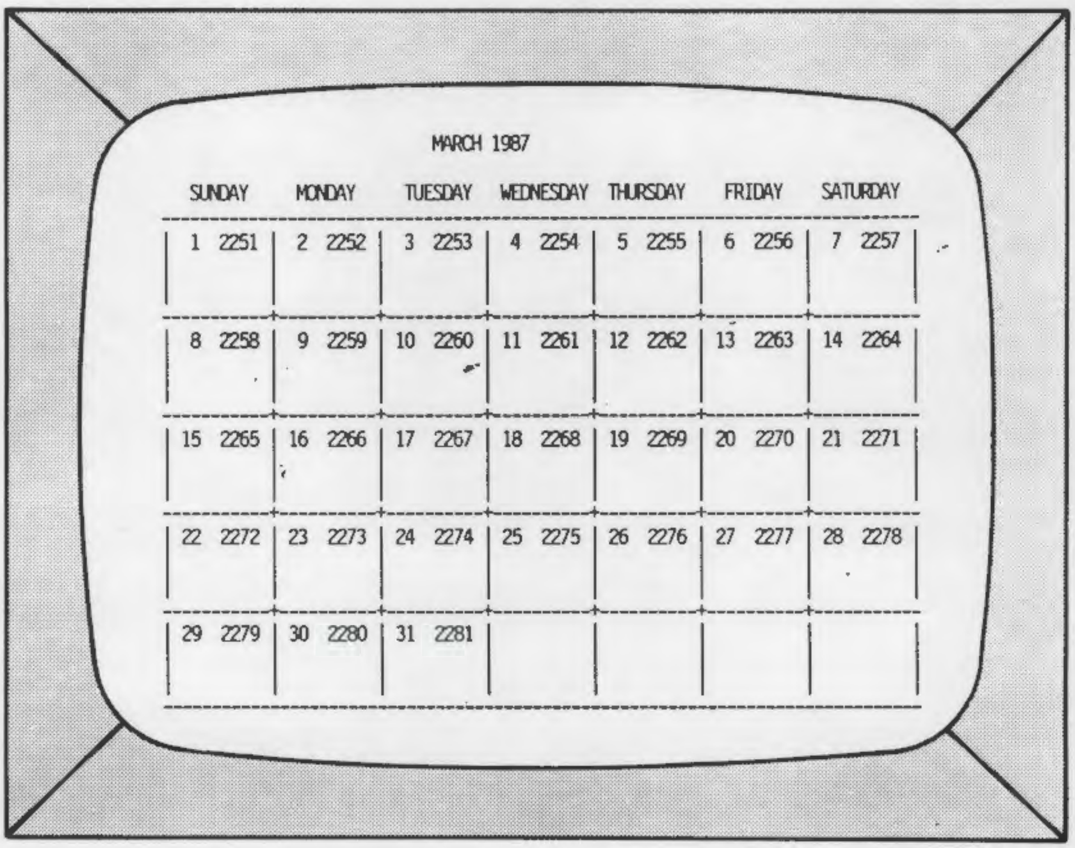
This method of altering the collection schedule applies to only the next polling event for any site. After the polling event, the PCDAS program will automatically schedule the next event based on the schedule criteria typed into the SITE.DRV file with the QUE program. If you want to permanently change the scheduling criteria, you should make changes in the SITE.DRV file (see Section 4.2).

4.5.3 Day of the Decade Calendars

A general purpose calendar maker is included in the REPORT program (option G). This routine prompts for the month and year and produces a calendar on either the screen or the printer. These calendars show the standard date and day, along with the day of the decade equivalent. Figure 21 is a sample calendar for March 1987.

4.5.4 House Cleaning Message Files

Be aware of the volume of message files that are present on the system disk. If you are running the PCDAS software on a floppy disk-based system, it may be critically important to erase unneeded message files or to back them up to a separate disk. This will avoid filling up the system disk. In a hard disk system, space may still be a concern, so it is a good idea to erase unneeded files or back up old message files on a separate disk. Option F in REPORT will delete message files for specific sites. Use this option with care, because all of the message files for the selected site(s) will be deleted for good.



MARCH 1987

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
1 2251	2 2252	3 2253	4 2254	5 2255	6 2256	7 2257
8 2258	9 2259	10 2260	11 2261	12 2262	13 2263	14 2264
15 2265	16 2266	17 2267	18 2268	19 2269	20 2270	21 2271
22 2272	23 2273	24 2274	25 2275	26 2276	27 2277	28 2278
29 2279	30 2280	31 2281				

FIGURE 21. REPORT Sample Day of Decade Calender

4.5.5 Data Capture Rate Reporting

The REPORT program has a utility that provides an easy way to view the data capture rate for a site over a 1-year period (option H). The capture rate for each day is shown along with the parameter set(s) active for that day. Appendix E is a sample data capture report for one site.

In addition to showing the daily capture rates, the reporting utility also gives monthly capture rate percentages and a yearly capture rate percentage. All of the statistics will account for differing integration periods and record lengths, if necessary.

5.0 DATA MANAGEMENT

The previous sections of this manual have described how to use PCDAS to collect data from a network of FDAS units. Before this data can be used, it must be converted into numbers with engineering units. Data conversion can be performed with either of two programs.

The TRANS program, detailed in Section 5.1, is a general data conversion utility that can be run from the PCDAS main menu. This program lets you convert all of the channel data for a particular site for a specified time interval without any reductions.

The ACCESS program is a PCDAS utility that can be run separately. This program lets you perform channel selection, aggregation, and basic time series reduction. The ACCESS utility is discussed in Section 5.2.

5.1 RUNNING THE CONVERSION PROGRAM (TRANS)

The conversion program TRANS has four major features:

- TRANS can convert data from any site or sites in the network using a simple driver file that you can edit on the screen.
- TRANS can convert the data over any interval of days for any of the sites specified. No channel reduction or time series reduction is applied.
- You can choose whatever engineering units suit you best. This is done by altering the CF1, CF2, and UNITS columns using the PARSET program. Remember that changing these values does not constitute a parameter set change (see Section 3.5).
- TRANS can convert the binary statistics files that are created automatically by the COLLECT program or by the STATS program.

The TRANS program has three limitations:

- It has no features for reducing time series data.
- TRANS can convert data from only a single parameter set per site. Separate conversion tasks can be assigned for a single site across different parameter sets.

* Although TRANS converts all of the channel data, channel subset conversion is not a feature.

The general function of the TRANS program is to convert the binary data contained in daily files into larger ASCII data files. In theory, the TRANS program could convert all of the data for all of the sites in a network in a single step.

The most important use of the TRANS program is in creating the engineering data files of any useful length (weekly, monthly) from the binary data base. This means that data conversion can be performed on a task-oriented basis.

If you are using the PCDAS software from a hard disk, run the conversion routine by simply typing **TRANS** at the DOS prompt in the root directory containing the PCDAS program files. You can also run the TRANS program by typing **F5** in the main PCDAS menu.

If you are converting data from a floppy disk-based system, you will have a separate "TRANS" disk that contains the conversion program,

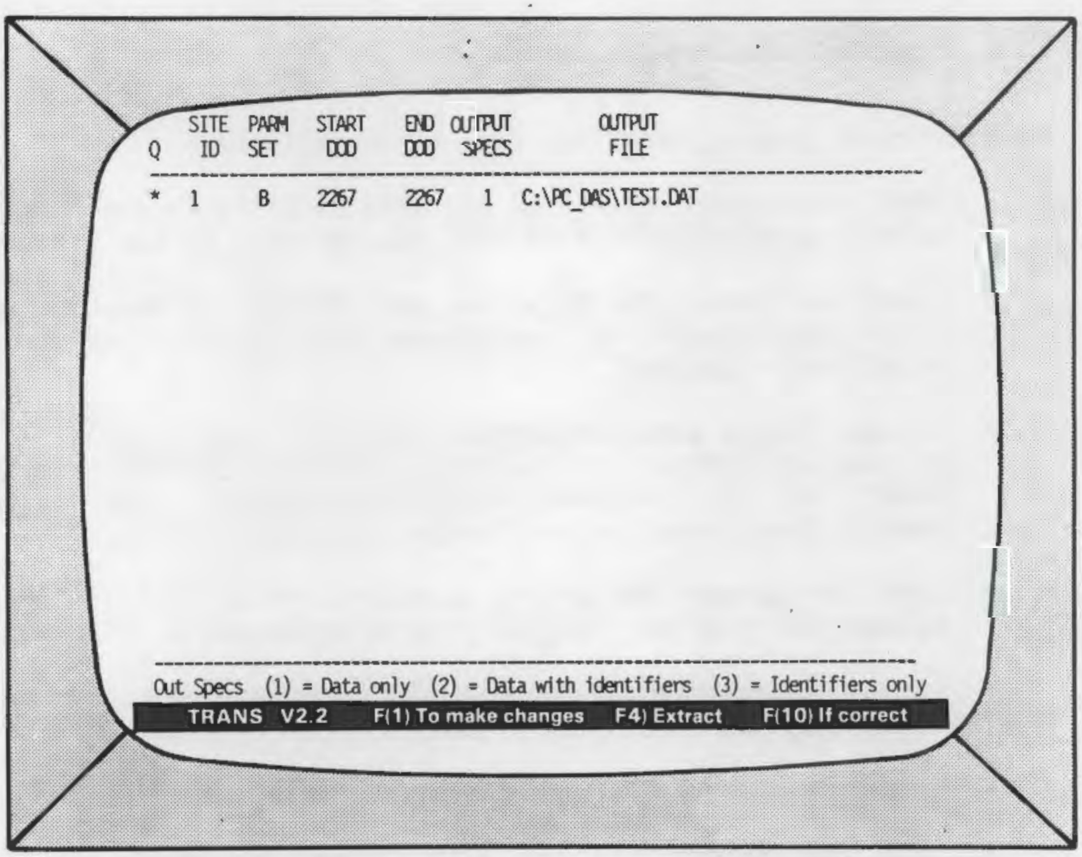


FIGURE 22. TRANS Screen Editor

driver files, and work space. Insert this disk in drive A and type **TRANS**. Drive B will contain the binary data files to be converted. Remember to keep an up-to-date copy of the SITE.DRV and USAGE.DAT files and the SITEDAT data base on the TRANS disk.

The first part of the TRANS program is a screen editor for the translation driver file (TRANS.DRV). An example of this screen editor is shown in Figure 22. To start editing, press **F1**. This puts you into the editing mode. When you are finished with your changes, press **F10** to exit the screen editor and start conversion.

The first entry is the Q column, which works exactly like the Q column in the SITE.DRV file. You can toggle this entry to the "*" character with any key to activate a particular conversion task. This feature lets you save your data conversion control information for all of the sites and parameter sets and use only the ones you need.

The next entries are the site number and parameter set code. These can simply be typed in. The next two entries are the beginning and end dates for conversion. Note that the dates are the day of the decade number rather than a conventional date. Refer to Section 4.5.3 on using the calendar printer in the REPORT program.

There are three choices for the format of data output. You can get channel identifiers only, data only, or both channel identifiers and data in the same output file. Generally, you will want to use option "2" (data with identifiers). The data option column is toggled between the choices with any key. Appendix F is a sample data file that was converted with the data-only option.

In addition to the '1-2-3' data output format options, there are a number of choices for converting statistical data. The options are:

- D - daily statistics
- W - weekly statistics
- M - monthly statistics
- any combination of D, W, and M
- SM - single month output format (append file).

The final entry is the file name for the engineering unit ASCII file. Remember that you can direct the file to any device by typing the path name as a prefix. For example, typing

C:\das\work\filename.dat

would direct the output to the subdirectory c:\das\work. You should set up a similar scratch subdirectory for your own output files if you have a hard disk. When you are editing the output file name, remember that the cursor is located at the end of existing file names. This lets you edit long path/file names quickly.

If you are converting statistical data, then the file name extension will be changed to differentiate between daily, weekly, monthly, or single-month output formats:

- filename.D - for daily statistics files
- filename.W - for weekly statistics files
- filename.M - for monthly files
- filename.SM - for single-month files; if the same file name is used for several SM conversions, then all of the single month summary information will be appended to the filename.SM file.

You should set up a work subdirectory for your own output files if you have a hard disk.

The TRANS program screen editor also has a copy feature that lets you copy information from one conversion task to another. This feature will reduce your typing effort for redundant conversion tasks. The F(2) key activates a column copy of the present cursor position entry. The F(1) key imports site numbers and parameter set codes from the SITE.DRV file for conversion tasks involving the entire collection network.

If you want to erase all of the old conversion tasks, simply delete the "TRANS.DRV" file in the PCDAS root subdirectory. This will clear all of the old data conversion tasks and let you type new tasks on a clear screen in the TRANS program.

After you have typed in the information using the TRANS screen editor, you can press F10 to exit. This will cause the program to enter the conversion routines that actually do the work. During this process, messages will be written to the screen that will allow you to track the progress of the conversion (e.g., which files are being open). Figure 23 shows what the screen might look like during the conversion process. All target ASCII data files are listed, as are the binary files to convert.

Conversion of the data may take some time to complete, depending on the amount of data you have specified. However, once you have exited the screen editor, the conversion task is completely defined and the computer will complete the conversion task automatically.

When conversion is done, the program will return to DOS or to PCDAS, depending on how you executed TRANS. You can then verify that conversion was completed by getting a directory of the created files.

If you are performing data conversion on a floppy disk-based system, you will have to copy the converted files from your TRANS disk to a

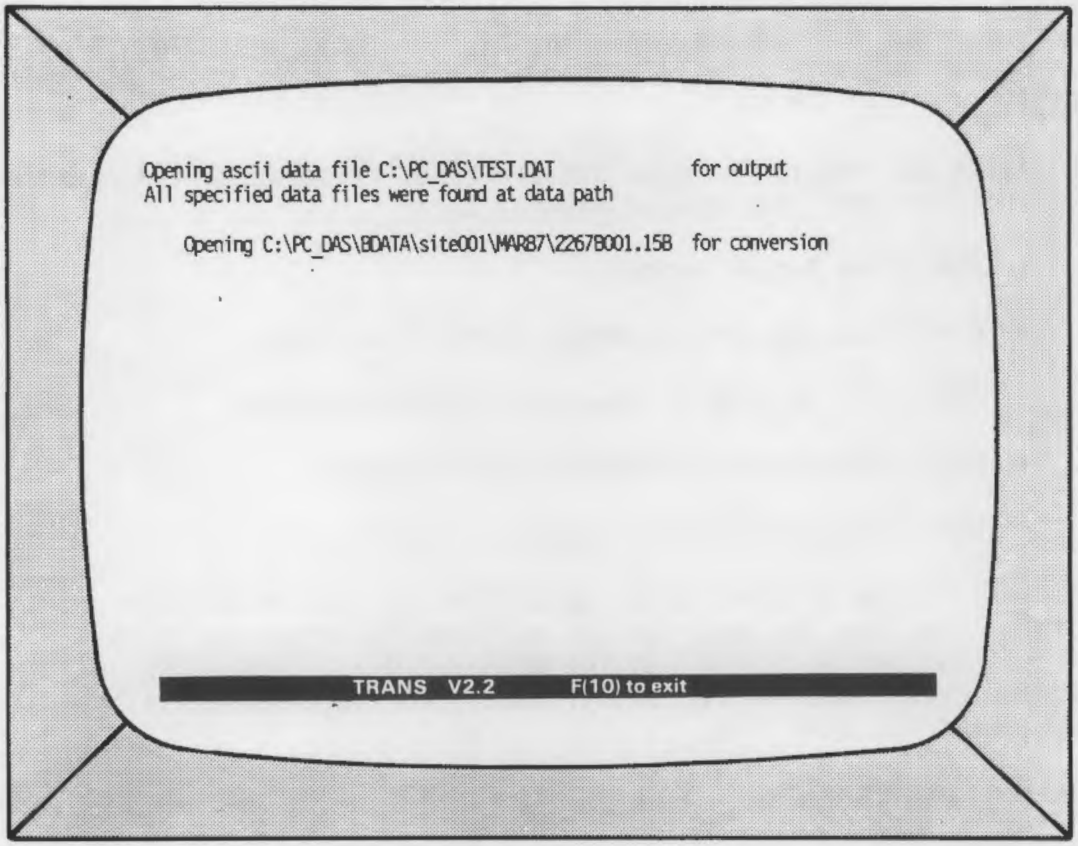


FIGURE 23. TRANS Sample Conversion Messages

separate data disk. This step is needed if you are going to preserve the available work space on the TRANS disk.

5.1.1 The EXTRACT Function

The EXTRACT function lets you extract a single record across all active sites. You specify the target time and the extract function searches the binary data base for the closest record to the target time for each site. Note that the most recent parameter set is used in the search. If the parameter set has been changed since the target time or if the data was not collected, then the EXTRACT function will report an "N.C." in the output file. This stands for "Not Collected".

The EXTRACT function can be accessed by typing an F(4) at the TRANS main menu. You will be prompted for the target day of the decade (DOD), the seconds of the day (0 to 86400 seconds), and the output file name.

5.1.2 TRANS Output Format

The format of the TRANS engineering files output is simple. Appendix F contains a sample data file that was created with option "1" (data only).

The first two records of data are the header records. Five entries are included with the regular channel data:

- LOG - the logger number
- DAY - the day of the decade in the time stamp
- SEC - the seconds of the day in the time stamp
- MIN - the closest minute for the SEC entry
- DQA - the data quality flag.

The rest of the entries in the data file are just what you might think. They are the readings (in engineering units) for each integration period for all of the measured quantities. The "Cxx" headers correspond to the channel number in the channel assignments section of the data file.

The data quality flag (DQA entry) is coded to flag certain errors in the ASCII data file. Two checks are performed on each time series record as it is converted:

1. If the 5-volt reference is turned on, then channel 8 is checked to see if it is greater than 256 or less then 254. This check assumes that the 5-volt reference data is reported in counts (CF1 = 1.0 and CF2 = 0). If the reference channel falls out of this range, then the DQA entry is 5.

   If the 5-volt reference channel does not read 255 counts, then there was an under- or overfilled integration period in the logger. This happens occasionally during clock resetting and record resequencing.

2. If there are negative entries in the record, then the DQA column is coded with minus the number of negative entries.

The coding of the DQA flag is set in the TRANS program and is arbitrary. The current coding is oriented toward electrical end-use metering. Other DQA coding methods can be adopted easily on an individual case basis. See Section 1.6 for user support.

All entries in the data file are in I6-integer format. If more digits of resolution are desired, you can multiply the conversion factors

(CF1 and CF2) by 10 raised to the nth power, where n is the number of extra digits required. Remember to change the units of the data when you do this.

You can report the data with any desired units. Just remember that the fundamental resolution of the data is defined by the original 8-bit conversion performed by the data logger.

5.2 USING THE PC DATA ACCESS TOOL (ACCESS)

The TRANS program is part of the PCDAS data collection system and is intended to perform the basic conversion of data to engineering units.

However, you will sometimes have to perform some basic types of data reduction and manipulation as the first steps in your data analysis. A more general data access tool called ACCESS has been developed to meet most of these basic reduction and manipulation needs. The ACCESS program operates on the binary data base only.

The ACCESS program is a PCDAS utility program and must be run separately by typing ACCESS in the PCDAS root directory. This program does not have a screen editor to specify the data ACCESS task; instead, a list-directed input is used.

The ACCESS program can perform two types of functions. First, the program can select and aggregate data from the raw channel data. This feature can be used to aggregate the electrical end-use shares for a building or combine temperature readings. Second, the ACCESS program can perform daily averaging, daily folding, and monthly averaging. These functions let you reduce the number of rows and columns of information before conversion and output. Of course, ACCESS can provide output of the data without row reduction in its maximum time resolution.

The ACCESS program makes heavy use of disk drives for temporary work space. For this reason, it is not recommended for floppy disk PC systems. A ramdisk (or virtual disk) is highly recommended to speed the data access task.

The inputs for the ACCESS program are similar to the CKSUM list-directed input files (Section 5.7). You can create, update, and archive the ACCESS driver files to suit your specific data analysis require- ments. You will need a non-document word processor or line editor to create or update the ACCESS driver files. A line editor that can handle long lines (greater that 80 characters) is preferred.

The next three subsections provide descriptions of the ACCESS software features, input specifications, and output format. These descriptions are intended to give you enough information to use the ACCESS software effectively. You will find that this tool has many of the advanced features expected from a data reduction tool. Hopefully, you will also find it relatively easy to use.

5.2.1 ACCESS Features

What follows is a list of the most important features of ACCESS. This list is included to give you an idea of what the software can and cannot do.

- ACCESS can perform column reductions, row reductions, and channel aggregations.

- Column reduction capabilities let you select any channel data in the PCDAS network (any site or sites) for output in a single file. The only restriction is that the site/parameter sets that you select must exist for the time frame you specify.

- Row reduction capabilities include daily average, monthly average, and daily folding. There is also a "none" option that allows you to report the data in its maximum time resolution.

- Channel aggregations let you define information in terms of other channels. These aggregations can be defined across sites if the channel numbers and parameter sets are valid for the specified time frame.

- Up to 20 aggregation equations can be defined for each aggregation/reduction set.

- Up to 10 different data loggers (sites) can participate in any aggregation equation.

- ACCESS uses a multi-step algorithm that reduces the data by column and row before the binary information is converted and aggregated into final form.

- ACCESS uses temporary disk storage for variable space, removing any constraints on the size of the aggregation/reduction or the time window of interest. The only limitations to the program are your patience and free disk space.

- ACCESS is not recommended for floppy disk-based systems. Ramdisks or virtual memory disks are supported.

- ACCESS gives fixed output format in scientific notation, providing 5 digits of resolution. Time stamp format options allow easy interface to several graphics and statistical software packages. The standard time stamp (day of the decade and seconds of the day) is also supported.

- ACCESS automatically handles missing data and daylight savings time.

5.2.2 ACCESS Input Specifications

The ACCESS program gets its inputs from a disk file that you create. Several basic rules guide the way in which these files are input.

- The file must reside in the SITEDAT subdirectory stemming from the PCDAS system root directory. This is where the conversion, logger control, checksum equation, and other data files are stored.

- The files all must have the same first name in DOS. This name is:

  AGG_RED.

  This name stands for "Aggregation_Reduction set" or A/R set. The last name, or DOS three-character extension, is the A/R set number. Any set number between 1 and 999 is allowed. The extension must be padded with zeros, so A/R set number one would be in a file named AGG_RED.001.

A sample A/R set file is shown in Figure 24. The lines in this file are numbered for demonstration only. All the characters up to and including the ":" character would not be included in an actual driver file.

This driver file demonstrates all of the features of ACCESS. The following list explains the input file shown in Figure 24, line by line:

- Line 1 - Two pieces of information must be included in the first line of the file. The first entry is the aggregation set

```
1 :001,PP&L TEST ACCESS AGGREGATION SET #1
2 :2148,2151
3 :SITE002,C,C68,C69,C70,C71
4 :SITE005,C,C71,C78,C79,C80
5 :*AGG01,  #2  OTHER,[(002C069)1.0 + (002C070)-1.0 + (002C071)-1.0]{    WH}
6 :*AGG02,  #5  OTHER,[(005C079)1.0 + (005C071)-1.0 + (005C080)-1.0]{    WH}
7 :*AGG03,GALLONS 2&5,[(005C078)1.0 + (002C068)1.0]{   GAL}
8 :OPTION=2
9 :SYNC=30
10:INTER=2
11:RAMD=E
12:DIAG=1
13:OUTFILE=C:\PCDAS\WORK\TEST.DAT
```

FIGURE 24. Sample ACCESS Driver File

number (and it should match the DOS extension for the file name). An A/R set descriptor is separated by a comma and can be as long as you like.

- Line 2 - The second line of the A/R set file must always contain two numbers separated by a comma. These numbers define the time window for the ACCESS run. The first number is the beginning day of the decade; the second number is the end day of the decade. The first two lines of the file are the only strict entry requirements. The other lines shown in the file can be entered in ANY order.

- Line 3 - All lines that begin with the keyword "SITE" (caps distinguished) identify channel data to be output in raw form. The word "SITE" must be followed by the site number and the parameter set code separated by a comma.

  The channel specifications must be separated by commas and be of the form CNN where NN is the channel number. Any number of sites and channels may be specified. ACCESS does not operate across parameter sets for individual sites, so no site number may be entered twice. Also, it is up to you to verify that the site/parameter sets exist for the given time frame. ACCESS will simply assign missing values if the parameter set was not active (no data).

  You can use the REPORT data capture reporting feature to verify time windows and parameter set codes (Section 4.5.5).

- Line 4 - Same as Line 3. Remember, data from more than one site can be output in a single A/R set run.

- Line 5 - This line defines an aggregation to perform on channel data. The first character may be a space or an "*" character. The "*" character indicates that the aggregation is to be performed, while the space character tells ACCESS to skip the equation. This allows you to define a large set of equations and use only those you need. The next 5 characters must be AGGNN where NN is the aggregation equation number. Twenty equations are allowed in any A/R set file.

  The next entry is separated by commas and must contain 11 characters. This is the aggregation description that will be included in the output file. Right-justify the characters in the field.

  The aggregation equation itself must be enclosed in "[]" characters. The form of the equation is relatively straightforward. Any channel data is identified by the characters NNNxCCC enclosed by the "()"characters. NNN is the site number, x is the parameter set code, and CCC is the channel number. Each item of channel data must have a multiplier specified.

The units for the aggregation must be enclosed by the "{}" characters and contain 6 characters, right-justified.

- Line 6 - The same as Line 5.

- Line 7 - This line includes an aggregation across loggers.

- Line 8 - A line coded with the keyword "OPTION=" will change the row reduction option. The choices are:

  1 None (report data in maximum resolution) [DEFAULT]
  2 Daily average
  3 Monthly average
  4 Daily folding
  5 Daily maximum
  6 Daily minimum

- Line 9 - Any line containing the "SYNC=" keyword will change the synchronization requirement for the data extraction. This provides a variable filter for the acceptance or rejection of ill-synchronized data (not collected on the end of the defined interval). The DEFAULT for this variable is 30 seconds (plus or minus).

- Line 10 - Any line containing the "INTER=" keyword will change the time stamp interface. The choices are:

  1 Julian date and day fraction
  2 Day of decade and minute
  3 Standard day of decade and seconds [DEFAULT].

- Line 11 - Any line containing the "RAMD=" keyword will set the drive specification for the temporary work space. The DEFAULT setting is no ram drive.

- Line 12 - Any line containing the "DIAG=" keyword will set the diagnostics option. Choices are:

  0 Do not save diagnostics information [DEFAULT]
  1 Save diagnostics in the DIAG.TMP file.

- Line 13 - Any line containing the "OUTFILE=" keyword will set the output file specification. This entry should be the last. No default is assigned.

Although it is not required, the organization shown in the A/R set driver file example is recommended. You should first specify the raw channel data to be output, then define the aggregations, and finally input the program control switches and options.

5.2.3 ACCESS Output Format

The output from the ACCESS program is a fixed length exponential format file. The first two entries are the date and time. The units on the date and time depend on the INTER variable defined in the A/R set.

INTER	Date	Time
1	Day of the century	day fraction (LOTUS)
2	Day of the decade	minute of the day (Statgraphics)
3	Day of the decade	seconds of the day (ELCAP)

All entries in the output file are 12 characters in width in exponential format. After the date and time, raw channel data (if any) is output, and the defined aggregations follow.

Three lines of header information are included to show the channel description, units, and unique ACCESS output identifiers. For channel data the ACCESS identification is of the form Cnnnxccc where nnn is the site number, x is the parameter set, and ccc is the channel number. The defined aggregations have an identification of the form AGGNN where NN is the aggregation number for the A/R set run.

5.3 WORK SPACE CONSIDERATIONS

Once the data files have been converted to ASCII files with engineering units, they are generally six times larger than their binary counterparts. If you plan to have much of this data available for analysis on the PC, you must make arrangements for lots of disk storage or perhaps tape backup. In any case, if you find yourself in a crunch for space on a hard disk, consider keeping your ASCII data on removable floppies or tape. If you are using a floppy disk-based computer system, you will have to copy the converted data off your TRANS disk to preserve the available work space.

The ACCESS program provides a means of reducing both the number of columns (channels or aggregations) and the number of rows of data before any conversion step is performed. Using this program is advised if you know the types of reductions to be performed on the data. By reducing the number of rows and columns, you will reduce the disk space requirements for the converted ASCII files.

If you use any program to convert and print hourly data for all active channels, you may have to carefully watch available disk space.

5.4 LISTING DATA ON THE PRINTER

The first step in any analysis or data verification is a visual inspection of the data. All of the data that is converted with TRANS is in row/column format, so looking at the individual records is quite easy. However, there may be more channels (columns) active than will fit across the top of a standard 8.5 x 11 page. To solve this problem, a simple program called WIDE will divide the file into 8.5 x 11-inch chunks and print them out to a different ASCII file or a printer.

To use the WIDE program, you must have the printer assigned as LPT1:. Simply type "WIDE" in the PCDAS root directory containing the WIDE.EXE program. You will be prompted to enter the file names of the files to be printed. You will also be prompted for the program (TRANS or ACCESS) that wrote the ASCII file. Note that any number of files can be printed in one job. After you have typed the last file name and typed two carriage returns, the files will be printed.

5.5 DATA ANALYSIS

Data can be analyzed on the PC using commercially available data analysis packages and the converted data from the TRANS or ACCESS programs. The analyses that can be performed on the PC are limited only by the software you choose and the available disk space on your computer.

If you want to do a simple analysis on a relatively small sample of data, you can use general purpose spreadsheet programs. For example, the data files created by TRANS can be imported into Lotus 1-2-3 using the /file/import/numbers function. This will take all of the numbers in the ASCII file and load them into the row and column cells in the worksheet. Before using this function, remember to rename your data file so that it has the ".PRN" extension. This import function will not import the column titles or the channel assignment section of the data file. However, this information can easily be reconstructed from a WIDE listing.

An alternative method for importing TRANS files to Lotus 1-2-3 is to read the file as text (/file/import/text) and then use the /data/parse function to divide the fields into numeric and label cell entries.

Once you have the data in the cells of a worksheet, you can do many kinds of simple analysis and manipulations.

The output from the ACCESS program is fixed exponential format and can be read by any statistics/graphics program that supports formatted ASCII file reads.

5.6 ENERGY BALANCE DATA QUALITY CHECKS

The cost of FDAS hardware has dramatically reduced the cost per measurement point for electrical end-use metering. In most cases it is economically feasible to completely monitor service panels, motor control centers, or switchgear enclosures. This means that both the main service feeders and the individual loads are all monitored. For multiple-phase services, each phase is monitored separately.

Figure 25 shows a typical 3-phase service panel and the associated FDAS channel numbers (in circles). Note that all three phases of the service feeders and the individual end-uses are monitored separately. This allows for multiple energy balance checks to be applied to single time series records.

The law of conservation of energy applied to an electrical service panel states:

> The energy entering the panel must be equal to the energy leaving the panel, assuming minimal losses due to resistance.

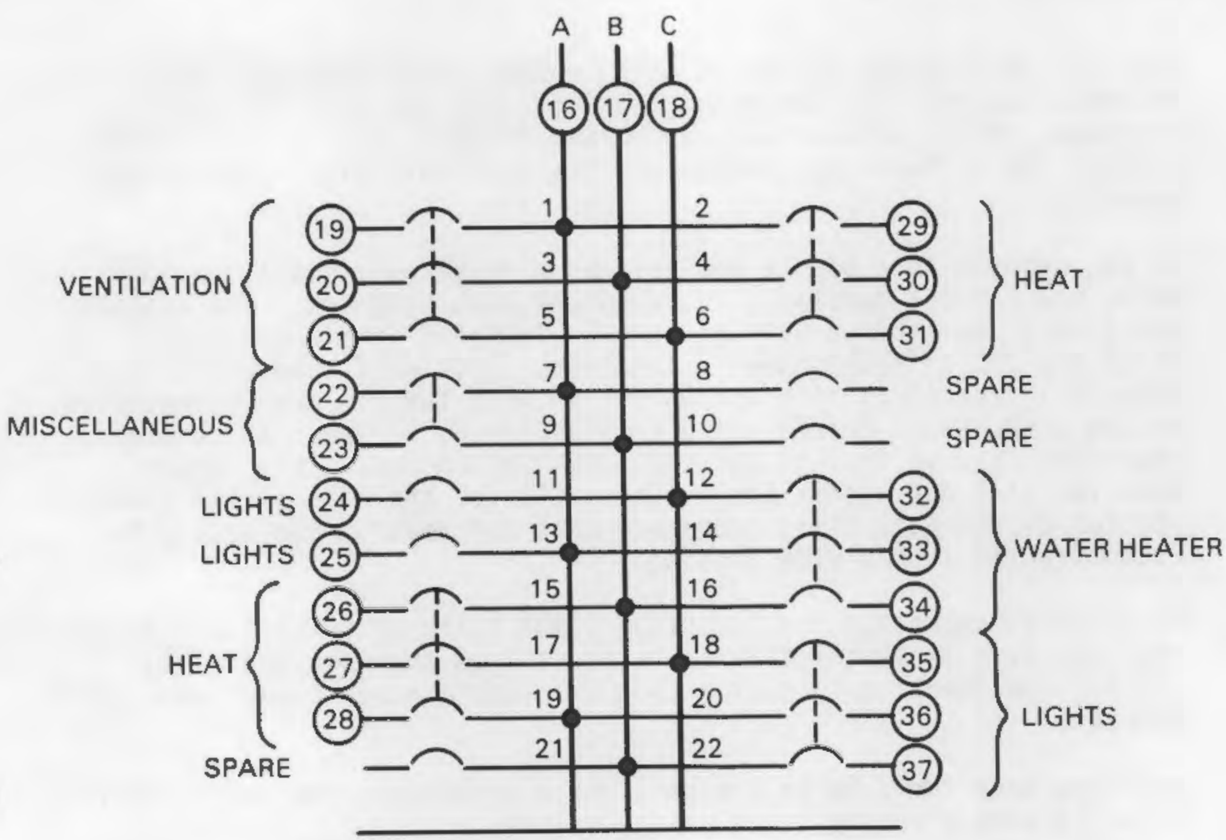


FIGURE 25. Typical Service Panel Showing FDAS Metering

This law can be applied to the panel shown in Figure 25 for the A,B and C phases. Written in terms of the FDAS channel numbers, the energy balance conditions can be stated as:

C16 = C19 + C22 + C25 + C28 + C29 + C33 + C36	(Phase A)
C17 = C20 + C23 + C26 + C30 + C34 + C37	(Phase B)
C18 = C21 + C24 + C27 + C31 + C32 + C35	(Phase C)

Each record collected from the FDAS should satisfy all three of these equations within the accuracy of the FDAS instruments. A PCDAS utility called CKSUM (for ChecK SUM) allows you to apply an arbitrary set of energy balance equations to a data file. The energy balance equations applied by the CKSUM program are a rigorous data quality check. CKSUM can also be used to catch any installation errors. Any faults in the FDAS (scaling error, inaccurate measurement plan, equipment failure) will be evident in the output of the CKSUM program.

The CKSUM program provides meaningful statistics on how each record passed or failed the defined energy balance conditions. The following section explains how to set up the input to the CKSUM program and explains the program's output.

5.7 HOW TO USE THE CKSUM PROGRAM

The CKSUM program was written to operate on engineering unit ASCII files created by the TRANS program. Because data is archived on the PC in unconverted binary form, you must run the TRANS program to create the data files. These data files need to contain channel identifiers as well as the channel data. This is done by selecting conversion option 2 in the TRANS program. Refer to Section 5.2 for more information on conversion.

Note that there is no file naming convention for ASCII data in the PCDAS system. This is a result of a design philosophy that assumes that data conversion will be performed for specific tasks only.

Before running CKSUM, you must:

- collect data into the binary archive
- convert the desired interval of data using option 2 in TRANS
- create the check sum equation file in the SITEDAT subdirectory
- create the modification file in the SITEDAT subdirectory (optional)
- clear enough working space for CKSUM to write its output.

The CKSUM program will prompt for the engineering units ASCII file name, the site number and parameter set code, and the integration period in seconds. It also prompts for two yes or no responses concerning suspected reversed CTs and channels modification.

5.7.1 File Formats and Names

The check sum equation file is required by the CKSUM program. This file should reside in the SITEDAT subdirectory with the .PAR and .DAT file for the particular site and parameter set. The naming convention is:

XSITENNN.SUM

where X is the single letter parameter set code, and NNN is the site number. Note that this convention follows the other file names (.PAR and .DAT). You should make entries to the .SUM file with a simple line editor or word processor. Be careful not to enter control codes in this file. A sample file listing is shown below.

```
1 C09 = C20 + C24 + C26 + C28 + C30 + C32 + C35 + C36 + C38
2 C10 = C22 + C25 + C27 + C29 + C31 + C33 + C34 + C37
3 C11 = C39 + C42 + C45 + C48 + C64
4 C11 = C40 + C43 + C46 + C49
5 C13 = C41 + C44 + C47 + C50
PHA PNLZ: C11 + C12*= C39 + C40 + C42 + C43 + C45 + C46 + C48 + C49
PHB PNLZ: C11 + C12 = C39 + C40 + C42 + C43 + C45 + C46 + C48 + C49
```

The conventions in this file are:

- Any line that contains a ":" character is an active equation that will be used by the CKSUM program. This allows you to enable and disable specific equations while maintaining all check sum equation information in a single file.
- Any characters that come before the ":" character in a line indicate the equation title. This lets you label the equations logically. In the above example the first five equations have simple integer labels and are unused.
- Up to 26 equations with any combination of main and feeder channels are allowed.
- An "*" character directly after a channel specification on the left side of the equation indicates that the check sum tolerance should not include the resolution for that channel. In cases where there is more that one main channel on the left side of

the equation, the sum of the non "*" channel resolutions will be used for the check sum tolerance.

The modification file lets you apply a channel-by-channel correction to the data before the energy balances are applied. This correction is in the form of a scale factor and offset:

CNN(new) = CNN(old)*scale + offset

These modifications can be used to test hypotheses of scaling of offset errors that may have been discovered from previous CKSUM runs. The file naming convention for the modification file is the same as the check sum equation file above. The modification file resides in the SITEDAT subdirectory with the .MOD extension. This file has the same "first name" as the .PAR, .DAT, and .SUM files. A sample file is shown below:

```
C45  1.0,12
C23  1.098,0
C34  .976,33
```

A space between the channel number and a comma between the scale and offset values are the only requirements for this file. Any number of the FDAS channels can have a correction applied before the CKSUM tests.

5.7.2 CKSUM Output

The CKSUM program does not prompt you for an output file name. It simply uses the input data file name with a .CKS extension. This extension should be reserved for CKSUM output files.

Because the FDAS converts the analog signals from the watt meter circuits into single byte numbers (0 to 255 counts), the energy balance equations are applied to a quantized set of data rather than to a continuum. The accuracy of the basic conversion is +- one count. For this reason, the pass/failure of any record is based on the resolution (watts per count) of the main feeder channel. If the difference between the feed and load sides of the equation is greater than one resolution of the feeder channel, then the record fails the energy balance check.

The CKSUM output includes a number of meaningful lists and statistics that can be used to judge the quality of indvidual records and the FDAS installation in general. Appendix G contains a sample CKSUM output file for a single sheck sum equation. The output consists of:

- list of channels with identifiers, engineering units, and channel resolutions (generally in watts)

- listing of the .SUM file showing the active energy balance equations

- general min/max information for input data set in both watts and counts (resolutions)

- general data analysis, including dead channels, number of records, and number of records excluded due to negative readings.

For each equation in the analysis, a set of lists and statistics is generated:

- list of records that failed the energy balance check - This list includes the time stamp of the record, and main and feed sides of the equation, difference, and active channels.

- general data analysis for the equation - This includes distribution of failed records, as well as statistics for failed records and all records.

- a histogram of the percentage of time that channels are on when records fail the specific equation

- a histogram of the percentage of time that records fail when a particular channel is on (non-zero)

- frequency distribution of the differences for all records.

5.8 FILE TRANSFER

If more complex analysis is required, or if you want to do analysis on a larger sample of data (say, a yearly regression on hourly data), you should seriously consider moving the data to a larger computer. Just as with analysis, the transfer task can be accomplished with commercially available software. Consult your mainframe system operator for suggestions on what programs would best suit the task. Generally, an error-free protocol with wild card file transfer capability is preferred.

The procedure for transferring information from the PC is simple. First, use the TRANS or the ACCESS program to create the engineering ASCII files of the correct duration (Weekly, Monthly etc..). You then use the commercial package to transfer the data files through your modem or, better yet, through a direct connect serial port to the larger computer.

6.0 TECHNICAL DESCRIPTION OF PCDAS

This section contains detailed technical descriptions of the PCDAS programs and their operations. Some of the design criteria, major limitations, and features of PCDAS are discussed. This information may be most useful to those who want to modify the code or for those who want to understand its operation in detail.

6.1 PCDAS SYSTEM DESIGN CRITERIA

PCDAS was designed to provide automated data collection and network control using readily available personal computer equipment and PNL-designed remote field data acquisition systems. Within this framework, the designers had great freedom to choose the mode of operation that best suited the task.

Some of the design criteria that guided the development of PCDAS include:

- The code should run on all IBM personal computer configurations, from a PC with two 360-Kb disk drives and 128-Kb memory to a PC/AT with fixed disks and megabytes of RAM.
- All data collection and network control operations would have both manual and automatic modes.
- The code would be user friendly, so inexperienced personnel could operate the system, but would also contain enough detailed information about the network status for experienced users.
- The code would be able to run unattended for long periods of time, requiring an operator only to interpret status and error condition messages.
- The code should enable several personal computers to access data from the same remote data loggers with one central PC acting as the network controller.
- The data archive format would be as compressed as possible so collected data would not overload the limited data storage capacity of most PC equipment.
- Error and status messages would be in plain English, where possible.

- The programs would be clearly organized and written in a higher-level language to facilitate modifications.

6.2 DATA COLLECTION PROGRAM (COLLECT)

This program is the central routine of the PCDAS system. It is a fully automatic program that accepts input from a driver file and writes to both a message file structure and a data archive. A large amount of error checking and recovery has been included in this code to prevent program crashes.

One of the major features of this code is its method of writing messages. Operator messages are written to the screen and to the message file structure at the same time. The message files written by COLLECT are in a highly compressed format to conserve disk space. Whenever these messages are viewed by the user, a message data file called "MESSAGE.DAT" is used to expand the messages into a readable form. The operator can either view the collection operations as they occur, or review duplicate collection messages at some later date. If any error conditions occur to the system during automatic operations, the user will be instructed to read the collection messages for a particular polling event and the error messages will be highlighted.

The COLLECT program goes through a sequence of events during a given polling. What follows is a detailed narrative of the events that take place during a COLLECT run.

6.2.1 Example of a COLLECT Run

After the initial dimensioning, the COLLECT program opens the USAGE.DAT file created by the INSTALL utility. The program reads the communications port number, the binary and ASCII data destination path names, the summary message file path name, and the dial-out prefix string. The usage number is incremented by 1, and the program checks to see if the number of runs since the last media replacement exceeds 200 (for floppy drive systems only).

Next, the site driver file is opened and the main data acquisition loop is entered. Records are read from this file, and sites with "*" in both Q (que) and C (collect) columns pass through the filter into the main subroutine calls.

At the beginning of any collection, a 15-minute timer is set to guard against excessive connect time in case of some failure.

Next, an appropriate summary file coded with the day of the decade, the site number, and the parameter set is opened. If more than one polling event takes place in a single day, then the message file is appended with the new polling messages.

The program then sets the top and bottom of memory, based on the type of remote processor, and then assigns several variables from the site driver file. These variables are:

- PARMSET$ single character string, one-letter parameter set code
- PROCVER$ processor version code string
- PHONENUM$ 16-character phone number string
- RECLEN record length in bytes
- INTPER integration period in seconds
- LDAY,LSEC day of decade and second of day of last archived record
- LREC position in memory of the last archived record
- NSEC,NCHAR number of seconds and the threshold characters for phone line noise evaluation.

After the variables have been set, the modem is initialized and the dial-out subroutine is executed.

The phone dialer subroutine gives the "ATDT" command to the modem and waits for one of three events. The modem will send a "CONNECT" within the specified wait period, or will send a "NO CARRIER" response, or will time out. In the first case, the subroutine immediately returns control to the main acquisition loop and the data collection, or polling, proceeds. In either of the last two cases, the dial-out will be attempted two more times. If these attempts fail, then an error message is written and the polling for that site is terminated.

After a successful connect, the program drops immediately into a phone line quality subroutine. This subroutine simply clears the communications buffer and waits the specified amount of time. If the number of random noise characters exceeds the threshold, then the line is failed and the program drops back to the phone dialer to try another phone line. The program will evaluate and fail three separate connects before giving up on the polling for a particular site. This phone line quality check is a simple random noise check. Line attenuation is not evaluated in this routine.

If the phone line does not have a lot of random noise, the program drops into the communications verification routine. This routine verifies that the field data acquisition system will respond to commands. Several commands are sent and the responses are monitored. If the processor is working, the processor version code is transmitted and checked against the entry in the site driver file. If the processor fails to respond or the version codes do not match, an error code is printed and the polling for that site is terminated.

The logger integrity check comes next. This is perhaps the most important of the subroutines, providing both network control and data quality control functions.

The first step in the logger integrity subroutine is a check of the remote logger's collection parameter block. First, the appropriate ".PAR" file is opened and parameters 5 through 146 are read in to the DPARM() array. Next, the program jumps to the logger parameter read subroutine, which returns the PARM array from the FDAS. The parameters are read out of the FDAS twice to ensure an accurate transmission. A one-for-one check is performed, and the results of the check are reported. Normally, this check will result in zero errors and the program will proceed to the next logger integrity check. However, if the check turns up an error, a jump to the AUTOPARM subroutine will be executed.

The AUTOPARM subroutine automatically corrects errors in the logger's parameter block. Parameter error can be divided into two categories, those that cause data corruption (number of active channels, bytes of resolution for active channels, or scaling on active channels) and those that do not. If the parameters caused data corruption, the program performs the following steps. First, the errors are corrected. Next, the corrections are checked against the disk parameters. Finally, if the disk and logger parameters match, the logger's memory is cleared. If the parameter error did not cause data corruption, the program goes through all of the same steps except the final memory clearing step. If the logger's parameters are correct or have been corrected, the program proceeds to check the logger's clock and record sequencing.

If the clock is found to be in error (plus or minus 60 seconds difference), it is corrected to match the clock in the data acquisition computer, and the records are re-sequenced on the end of the standard clock period. A separate check of the record sequencing is made after the clock check to ensure that records are synchronized to the end of the integration period. After the record sequencing step, a return to the main acquisition loop is executed.

After the logger integrity subroutine, a conditional statement decides if the logger's memory has been cleared due to parameter block errors. If it has, the polling for that site is terminated. Normally, the program will proceed to the data requirements subroutine, which will decide how much data is to be collected from the logger.

The data requirements subroutine is a fairly complex piece of code that selects what data needs to be collected from the logger. The complexity arises from the multitude of possibilities presented by the logger's data collection scheme. The output of this subroutine is a starting address and the number of records required; alternatively, this subroutine will position the remote logger's transmission pointer itself.

Under normal circumstances when a logger has already been polled at least once, the data requirements subroutine will position the transmission pointer just before the memory location where it collected the last record in the data archive. It will then transmit several records forward, searching for the last record in the archive. If it finds a match, the routine calculates the number of records to collect and jumps immediately to the data acquisition routine. If a match is not found, it assumes that the logger's last archived record has been over written. In this case, the routine specifies that all data in the logger should be collected.

The data collection routine transmits and checks the data required from the logger. A collection algorithm is implemented that guards against repeat records and records with transmission errors. The records are check-summed, and the record length is checked on every transmission.

A message is written to the screen for each record collected. Several diffent types of messages can be written. The data acquisition routine accounts for data transmission errors:

| Collect nnn |means that record nnn was collected successfully

| Missed nnn |means the FDAS failed to see the transmit command

| Checksum nnn |means a check sum error was detected

This routine creates a temporary file called "HEAP.TMP" on the data destination root directory. This file contains the time sequenced data from the logger for that particular polling. This file will be accessed by the file manager subroutine that creates the binary data archive. Immediately after the data is collected, the modem hangup routine is executed to disconnect the modem and minimize connect time.

The file manager subroutine opens the HEAP.TMP file and executes a two-pass algorithm. First, the data is read in and the time stamps are checked. If the logger's clock was zeroed (see logger integrity subroutine), then the time stamps of the records are reconstructed in reverse sequence from the most recent record to the first record after the power came back on. This is done using the known time stamp of the most recent record in the logger, based on the data collection computer's time. If any time stamp reconstruction is performed, the operator is notified.

The second pass of the file manager subroutine through the HEAP file actually opens archive files and stores the daily binary files.

The final step in the collection process is to open the SITE.DRV and SCHEDULE.DAT files and write the updated collection information. The next section covers more on the SITE.DRV file format and the information that is maintained in this file.

If automatic statistics generation is enabled, then the COLLECT program makes one more pass through the HEAP.TMP file. The program accumulates the raw binary data, number of records, maximum reading, minimum reading, time of maximum, and time of minimum. These statistics files are accessed using the TRANS program, which converts the binary statistical information into ASCII files.

The polling event described above is one of the central events in the PCDAS system. Much of the following discussion covers the programs that schedule and report on these polling events.

6.2.2 Data Collection Driver File (SITE.DRV and QUE)

The site driver file controls the polling events. The operator makes entries to this file and edits entries using the QUE program. The SITE.DRV file contains the information that controls:

- organization
- scheduling
- call-out sequence and phone handling.

All of the entries in this file will be explained by the categories listed above. You will note that the entries in the site driver file records are organized in the same functional order listed above.

Organizational

Q$

1 character. This variable is an indicator that shows if the site is working and is capable of collecting data. A null for this variable indicates that the site is not working and an "*" indicates that it is working. In the QUE program this entry can be toggled with any key.

PARMC$

1 character. This variable indicates that the site is ready for polling. If both the Q$ and PARM$ are the "*" character, then the QUE program automatically checks to see that the integration period and record length entries are not blank. If they are blank, QUE

accesses the site data base and calculates these values. See Section 4.2 for more discussion of QUE features. The way the PCDAS scheduler program causes a collection event is to control this entry.

<u>SI$</u>

3 characters. This is the site number (max=300).

<u>PS$</u>

1 character. This is the single-letter parameter set code. Changes to this entry cause the record length, integration period, and last archived record entries to be erased when using the QUE program. These values are calculated from the new parameter set specified.

<u>PV$</u>

5 characters. This is the processor version code. PCDAS version 2.2 works with most versions of the FDAS firmware. A change in this entry will reset the last archive time stamp entries. These will be re-entered after the next polling event for the site.

<u>WRITEP$</u>

3 characters. This is the write-protect key for a particular site. The default is "OFF" allowing the collection computer to control the site clock and parameter block. If this key is toggled to the "ON" position, PCDAS will report errors but will not correct either the clock or the parameter block.

<u>FST$</u>

7 characters. This is the first of two seven-letter codes that can be assigned to the site. The default is blank. This 7-character code can be used to subdivide the network by building type or some other criterion.

<u>SCD$</u>

7 characters. The second of two seven-letter codes that the user can assign to the site.

<u>RECLEN$</u>

4 characters. This is the number of bytes per record. It is calculated from the ".PAR" file in the site data base for that particular site. If the correct site file has not been created, the operator will be advised to create the file and return to the QUE program.

INTPER$

5 characters. This is the integration period for the particular site and parameter set. It, too, is calculated from the ".PAR" file when the record length is calculated.

STATS$

3 characters. This string contains the characters D,W,M or any combination. This indicates what type of statistics are to be routinely generated by the COLLECT program.

Scheduling

LDA$

5 characters. This is the day of the decade of the last record in the archive. It is blank until the first polling event for that site/parameter set. The COLLECT program updates this variable after each polling.

LSEC$

6 characters. This variable is the seconds of the day of the last record in the archive. With the LDA variable, it comprises the time stamp of the last record in the archive. The COLLECT program uses these two variables to find the last record and thus determine how much data is to be collected from the logger.

PFULL$

3 characters. This is the percentage of logger memory used needed to trigger a polling event. This criterion can be used alternately with the day and time variables. Default is 50 percent full.

DAYI$

3 characters. This is the number of days between polling events. If the day interval and collection time are specified, then the PFULL variable is turned to "OFF".

CTIME$

8 characters. This is the collection time associated with the DAYI variable. It is in standard hh:mm:ss format.

Phone line quality criteria variables

NSEC$

2 characters. This is the number of seconds to wait during the phone line quality check. The default value of 20 seconds is entered whenever a new site is entered in the queue.

NCHAR$

2 characters. This is the number of noise characters allowable in the NSEC wait period. This variable is set to 2 characters as a default when a new entry is made to the site driver.

PH$

16 characters. This is the site's phone number. It is used with the ATDT Hayes dial-out command. A prefix letter "P" will force pulse dialing.

BAUD$

4 characters. This is the baud rate for communications. Default is 1200 baud.

CWAIT$

3 characters. This is the number of seconds to wait for a carrier before trying to dial again. The default of 60 seconds is assigned when a new site is added to the queue.

The QUE program is a general screen editor that lets the operator make changes or additions to the SITE.DRV file.

QUE also has some automatic features that simplify network control operations. The most important automatic feature is the calculation of the record length and integration period from the site data base. This is a necessary step in setting up collection on a site and is initiated whenever an "*" character appears in the Q and C columns. If the operator forgets this step and tries to manually collect data from the site, the COLLECT program will trap the error and advise the operator. Other features of QUE include the automatic record initialization whenever the processor version is changed or when the parameter set code is changed. All default values are initialized when a new site number is entered.

The QUE program uses fold-over screens to edit all of the entries in the site driver file. Three screens of information can be cycled through with the essential information (site number, parameter set code) constant on the left side of the screen. Section 4.2 includes some samples of the three screens.

All variables in the SITE.DRV file are read into a string array in QUE called SITE$(,). The screen editor logic is simplified by the selection of simple indexing in this array. When all changes have been made, the program assigns elements in this array to the random access file field variables.

The coding for additional screens of information is relatively straightforward. Changes or additions to the site driver file should be relatively easy.

6.2.3 Data Collection Schedule File (SCHEDULE.DAT)

Two options are available for deciding when to collect data from the site. Data can be collected manually on some set schedule (e.g., weekly), or the PCDAS software can set the schedule and automatically collect data. As discussed in previous sections, the polling routine can either be run directly from DOS or be run automatically. The schedule file and the schedule choices entered in the site driver file determine when data will be collected if PCDAS is run in the automatic mode.

The schedule file contains the site number, parameter set code, the date and time of the last polling event, and the date and time of the next scheduled polling event. These entries in the schedule file are written by both the COLLECT and PCDAS programs. The interaction between programs goes like this:

> Whenever the COLLECT program is run, the last action is the update of the schedule file. The program writes the site number parameter set code and the date and time of the polling. It also erases any entry for the date and time of the next polling event.
>
> In automatic mode (when PCDAS is running), control is transferred to the PCDAS program when COLLECT ends. PCDAS then makes several decisions based on the condition of the site driver and schedule files.
>
> - If there is no entry in the schedule file for a corresponding entry in the site driver file, then an immediate polling event is scheduled (new site).
> - If the date and time of the next scheduled polling event in the schedule file is blank, the PCDAS program accesses the site driver file and calculates the scheduled time based on the date and time of the last polling and the collection criteria (% full or specified interval).
> - PCDAS performs no calculations if the entries in the schedule file are complete.

During normal operations, the schedule file will be complete and PCDAS will not schedule any immediate polling events. Section 6.5 contains more information on the PCDAS scheduler and controller program.

6.3 SUMMARY AND ERROR MESSAGE REPORTING PROGRAM (REPORT)

One of the most important features of the PCDAS software is the clear presentation of system operations. During the actual polling event, messages written to the screen show the condition of the logger and the status of the data collection. More important, the PCDAS software stores this message information so the operator can let the polling events run unattended. The COLLECT program interacts with the PCDAS scheduler/controller to inform the operator of any error conditions in the network. If there is an error, the operator can easily access the message data base and determine the nature and perhaps the cause of the error. The REPORT utility contains the subroutines that help the operator understand the status of the network. Appendix A contains a list of the error and summary messages and includes a short explanation of each message.

The REPORT program, like most other programs in the PCDAS software, can be run either manually from DOS or from the PCDAS scheduler/controller. The method of error reporting is outlined below.

When an error occurs during a polling event, the COLLECT program raises a flag that is read by the PCDAS program. When control is returned to PCDAS, the collection messages for that particular site are flagged. Also, so long as an error condition flag is raised, PCDAS will not schedule polling events for that site. The only way to clear the flag is to read the error condition messages.

To read the messages, the operator executes the REPORT program from PCDAS. This program has several routines that conveniently reconstruct the messages from the binary message data base, highlighting the important error messages. Depending on the routine that is chosen, the operator can:

- Read all of the error condition messages that have accumulated.
- Read the error condition messages for a particular site.
- Read the most recent collection message by site.
- Read specific message files.

There are several ways to reconstruct messages using REPORT; however, all of the routines access the same binary message data base. The binary message file can be thought of as a list of pointers and arguments. The pointers point to an actual message in a reference file, and the associated arguments are used to fill in the blanks in the messages. The reference message file (MESSAGE.DAT) contains approximately 110 separate summary and error condition messages. This file is used with individual binary messages files to expand the messages into a readable form. This method of storing messages has numerous advantages in terms of access time and disk storage utilization.

In addition to expanding messages from the binary message files, the REPORT utility also includes several housekeeping and scheduling routines.

Housekeeping routines include a message file eraser that gets rid of old message files that are not needed and a calender subroutine that will print a standard calender for any month after January 1981. The calender routine can print the information on a printer or to the screen. REPORT also includes a data capture report generator that is essential for determining the capture rate for days, months, and years.

Perhaps the most useful REPORT utility is the schedule editor. This routine lets the operator look at the SCHEDULE.DAT file and edit the entries. An important feature of this routine is that it shows the percentage of logger memory utilized at the time of the next polling. If the collection date is changed and the percentage of logger memory exceeds 100 percent, the operator must choose a different collection date and time. This routine can be used to set the polling time for all of the sites in the network by typing **all** at the site number prompt.

6.4 TERMINAL PROGRAM AND MANUAL OPERATIONS (PARSET)

The PARSET program is a general purpose program that provides many of the organizational functions of PCDAS. Setting up a remote collection system involves a considerable amount of coordination with the people installing the equipment. The PARSET program has a number of utilities that help in this effort. The program includes routines to:

- use the computer's modem to communicate with a site directly - The dumb terminal routine interacts with the rest of the PARSET program to provide several manual logger control functions. This routine includes automatic dialing from phone numbers in the site data base.
- examine or change the logger's clock
- load parameters from the site data base on disk to the remote logger
- manually check the logger's parameters against the site data base

- input, edit, or print measurement plans.

Like other programs in the PCDAS software, this program can be executed directly from DOS or can be called from the main PCDAS program. On startup, the program initializes a considerable number of arrays to handle the screen editing of the parameter sets. After the initial dimensioning, PARSET opens the USAGE.DAT file, reads the appropriate communications port number, and opens the port. The main menu is then printed to the screen.

The main menu offers several choices. Option "A" (parameter set editor) requires the user to specify a site and parameter set with option "D". Option "B" (manual logger control) requires the user to establish a connection with the logger with option "C" (dumb terminal). This option also requires the user to specify a site and parameter set with option "D" if parameters are to be loaded to the logger. Option "C" (dumb terminal) does not require the user to specify the site number or parameter set code. Functional descriptions of these main subroutines follow.

Option "A" (Parameter Set Editor)

This subroutine lets the user input the measurement plans and store them to disk, edit measurement plans, and print measurement plans. When this routine is executed, it first reads and processes the ".PAR and ".DAT" files from the site data base. If no site and parameter set code has been specified, then the user will be prompted to use option "D" and the main menu will return.

The first step in reading the parameters is to input data from the ".PAR" file. This file is simply an image of the logger's parameter block. It contains information on which channels are active (collecting data), how many bytes of resolution there are for each channel, and the scaling factor for each channel. The information on active channels and bytes of resolution is contained in binary. A "1" in a particular bit means that the channel is active, a "0" means inactive. In the bytes of resolution data, a "0" bit means two-byte resolution for that channel and "1" means one byte.

After reading the ".PAR" file, the program opens the ".DAT" file for the particular site and parameter set. This file contains descriptive information as well as the conversion data. The entries are:

- TI$ — eleven-character descriptor field for the measurement being made by a particular channel
- CC1$ — six-character number for the first calibration constant (multiplicative)
- CC2$ — six characters for the second calibration constant

- UNT$ six-characters to specify the units on the engineering unit data.

- MAXVAL the maximum value (in counts) expected on a particular channel for any given reading

- OF$ the counts offset for any channel (use primarily with energy use measurements).

All of the information in the ".PAR" file and the ".DAT" file is read into arrays for use in the screen editor portion of the program. When the program is finished reading in the information from the site data base, the user is presented with the "parameter change menu." The user can change the integration period, print the measurement plan to a file or the printer, save the parameters to disk, or edit any group of 16 channels. If the user specifies a site and parameter set that is not in the site data base, then default values will be assigned. The defaults are:

- all channels inactive

- all channels have two-byte resolution

- maximum value for any scan set to 255

- all descriptors and constants blank.

When typing an entry, the user can backspace to make corrections and use either the carriage return or arrow keys to complete the entry.

After the entries have been made, the work can be saved to disk using the "S" option in the parameter change menu or by exiting to the main menu. In either case, a prompt will request the site number and the parameter set code. If the files already exist on disk, the prompt will be a "Cancel Action/Replace" message. When exiting the parameter change routine, the prompt will always ask if the work should be saved. If no entries or changes have been made, the user can simply type "N" at the prompt.

If a change is made that affects FDAS operations, and the same site number and parameter set code is used during the save option, then an error message will appear and the save will be aborted. See Section 3.3 for more information on parameter set changes.

The second main functional area of PARSET is the routine for handling the logger's clock and parameter block.

<u>Option "B" (Manual Logger Control)</u>

After establishing communications with a logger using option "C" in the main menu, the user can return to the main menu and execute option "B." This routine first reads the logger's clock and then presents a list of choices. Included at the bottom of the menu is the time and date of the logger's clock. If this clock does not match the data collection computer's time, the user can execute the "D", "T", or "M" options to change the clock.

The "P" option is used to load parameters to a logger. Its primary use is for setting up a site after installation, because the PCDAS software includes routines to automatically correct the parameter block once data collection has started.

The "P" option causes the program to access the ".PAR" file and read the parameter block image from disk. The parameter block on the logger is then zeroed, and the parameters are changed one at a time. The entire process can take up to 3 minutes to perform, depending on the number of activated channels and the communications baud rate. When the routine finishes the parameter change, it reads the parameters out of the logger and checks its work. Any errors are automatically corrected, and the process is repeated if necessary, until all parameters in the logger match the disk parameters.

If the user does not want to change the logger's parameters, option "C" can be used to check the parameter block for accuracy without altering any of the parameters or the clock.

<u>Option "C" (Dumb Terminal)</u>

This routine puts the user in direct communication with the modem on the data collection computer. All of the characters typed will be interpreted as commands by the Hayes (or compatible) modem. By typing the AT DT dialing sequence, the user can call the remote logger and establish a connection. Once connected to the logger, there are several commands that the logger will recognize.

Some of the most useful of these commands common to PCDA and PCDB processors are:

- #   parameter block listing in ASCII
- @   scans of analog channels in counts (64)
- $ gives processor ID code and SOP version in ASCII
- %.   erases logger memory (BE CAREFUL)
- *.   performs software reset on the logger
- (41)bb))30'"   scans of digital channels (48).

The user can enter some of these commands to verify that the site is communicating. Once the connection is established, the user can press F10 to exit the terminal routine to the PARSET main routine. Any of the other PARSET routines can then be executed while there is a connection to a logger. Specifically, the "B" option described above can be used to control the logger's parameter block and clock.

The PARSET program provides several manual operations on the data collection network, as well as a screen editor for the input of the measurement plan. Because this program has many different functions, it is the largest of the PCDAS programs. In contrast to the PARSET program, the PCDAS scheduler and controller automates many of the collection functions.

6.5 AUTOMATIC OPERATIONS (PCDAS)

All of the programs described in Sections 6.2 through 6.4 can be executed either from DOS or from the PCDAS program. Normally, the operator will collect data or enter measurement plans by executing the PCDAS program. By typing "PCDAS" at the DOS prompt, the operator has access to nearly all of the data collection software. Put simply, the PCDAS program is a scheduler program that works on both the SITE.DRV and SCHEDULE.DAT files to schedule collection events for sites in the collection queue. The main logic in this program is a loop that continually checks to see if any site has met or exceeded the collection time criteria. While the program performs this task, the operator can interrupt the loop and execute any of the other programs. A single screen of information shows the next 10 scheduled collection events, the operator messages, and the function key menu for the other PCDAS functions.

When this program is started, it accesses the USAGE.DAT file and reads the path name of the data destination root directory.

The scheduler function works by going through a series of conditional statements based on the condition of the SITE.DRV file and the SCHEDULE.DAT file.

- When the value of the site number in the SITE.DRV file is zero (end of file condition), the program assumes there are no more sites in the queue.
- If the "Q" column in the SITE.DRV file is not "*" for any site, then it is skipped.
- All "*" characters in the "C" (collect) column are erased. The program will use these flags to control data collection.
- If there are no entries in the SCHEDULE.DAT file for either the last collection or next collection times, then an immediate collection is scheduled. This is used only when setting up the network or when adding a site.

- If there is an entry for the last collection date but none for the next collection date, then the program calculates the next collection date based on the information for the SITE.DRV file. The next collection date is calculated based on either a percent full or on a set interval basis. Remember that the COLLECT program automatically erases the date of the next collection and updates the date of the last collection each time it collects data for a site.

When the PCDAS program has gone through the five conditional steps above for each site, it continues to the main schedule loop. In this loop, the site number, parameter set, and the time and date of the next collection are written to the screen for the next 10 polling events. The function keys are enabled allowing the user to:

- manually collect data from any or all sites (F2) - This is performed by first executing the QUE program. The operator enters "*" characters in the "C" column for the sites to have data collected. A branch to the COLLECT program is automatically executed when the operator exits the QUE program. After data is collected, a branch to the PCDAS scheduler/controller is executed, bringing the user back to the starting point.

- execute the QUE program (F3) - The operator can execute the QUE program to modify or add to the SITE.DRV file. A branch to the PCDAS program is executed upon exit.

- execute the PARSET program (F4) - The site data base can be updated or the operator can perform manual operations on the network. Again, when this program is exited, an automatic branch back to the PCDAS program is performed.

- execute the REPORT program (F1) - The operator can read error messages and thus clear the message portion of the PCDAS main screen, alter the SCHEDULE.DAT file, produce day of the decade calenders, or do file housekeeping.

- execute the TRANS program (F5) - This program converts data from the binary data base and creates ASCII files.

- exit the PCDAS scheduler/controller to DOS.

The PCDAS program uses CHAIN commands to execute the various programs. Whenever control is branched back to PCDAS using a CHAIN command, the program starts execution from the top and the schedule routines are run again. Changes to the collection schedule using REPORT will automatically be taken care of when PCDAS is re-executed.

When a collection time is reached for a particular site, the PCDAS program goes into a 30-second delay loop and then runs through the scheduler loop one more time to pick up other sites that may have collection times within 30 seconds of that site's collection time. This minimizes the access calls to the COLLECT program for networks where a large number of sites have synchronized data collection. If an error is encountered during the data collection, a flag is raised for that site. On return to the PCDAS program, the sites with error conditions can be seen in the lower portion of the screen. If a site has an outstanding error condition (one that has not been read using REPORT), no new collection events will be scheduled for that site.

When the operator exits the PCDAS program (F10) and returns to DOS, the program automatically writes the ATTN.DAT file, which contains a list of the sites with outstanding error conditions. On startup, the program reads this file and reminds the operator of the error conditions.

6.6 DATA, FILE STRUCTURES, AND CONVERSION

The PCDAS software makes extensive use of disk storage for its operations. This minimizes the amount of memory needed in the PC collection computer. Rather than storing large arrays of data or site information in memory, the collection computer uses information on disk and acts as a filter between the remote sites and the binary data on disk. The programs write data to disk frequently when they are running. The formats of the files that PCDAS uses are fixed.

The information stored on disk can be divided into two broad categories. First, there is data from the logger. This data is in binary format and is in the same format as the data stored in the logger's memory. That is, the COLLECT program does not alter the format of the data records when they are collected. When the data is converted to ASCII files with engineering units, the format and size of the files change.

The second broad category of information is support or system data that the PCDAS routines use to collect the data from the loggers. This includes the driver files, message data base, and the site information data base.

To understand how data is collected from the network of sensors, it is important to know how the logger reduces data and forms binary records. The discussion below is divided into four sections describing binary data in the FDAS, binary data on the PC, the format of PCDAS system files, and, finally, the data conversion process.

6.6.1 Binary Data on the FDAS

The data storage and transmission of data on the field data acquisition system were designed to be highly compressed. This compressed format allows the remote unit to make good use of the limited space (15360 bytes) for data storage and facilitates the quick retrieval of data with a modem.

In essence, the binary record is a list of scaled summations of readings from the active channels. The way these sums are taken and the method of scaling comprise the data reduction in the FOAS. This process differs slightly between digital and analog inputs, so these will be discussed separately.

Digital Channels

There are 48 digital input/output channels of the remote logger that are capable of measuring contact closures or 5-vdc pulses up to 60 Hz. The output capability of these channels has not been implemented to date in any of the firmware versions written for the logger.

All 48 channels have an associated temporary single-byte accumulator that sums the pulses over three seconds. At the end of each 3 seconds, that byte is added to a three-byte accumulator for the channels and is cleared for the next 3-second accumulation. The scans are added to this sum until the end of the integration period. When it is time to write a record to the logger's battery backed memory, the processor goes to the control parameters in memory to decide which channels are active and how to scale the three-byte sums.

For each active channel, the processor scales the three-byte sum by performing a number of right shifts on the binary sum. This shifting actually truncates the least significant bits in the three-byte sum. The number of right shifts is called the scaling factor. This number depends on the length of the integration period, the maximum value expected from the channel in any given scan, and the number of bytes of the three-byte sum to be saved. Either one or two bytes can be saved.

An example of this reduction process and the ensuing expansion of the data when it is converted should be helpful.

Sample Oata Reduction:

The first step in the reduction of the data is the scaling of a three-byte binary sum to either one or two bytes. If you had a binary sum that looked like

0 0 1 0 1 1 0 1   1 0 0 1 1 0 0 1   1 1 0 0 1 1 0 0

you would have to perform 14 right shifts if you want to save a single-byte representation of this sum. The resulting shifted binary number would look like this:

0 0 0 0 0 0 0 0   0 0 0 0 0 0 0 0   1 0 1 1 0 1 1 0

Note that only six right shifts would have to be performed to save a two-byte representation of the three-byte sum. The result after performing six right shifts would be

0 0 0 0 0 0 0 0   1 0 1 1 0 1 1 0   0 1 1 0 0 1 1 1

The two-byte representation obviously has eight more significant bits than the single-byte representation. The numbers are expanded by multiplying by 2 raised to the number of right shifts (2^shifts). To demonstrate the effect of this truncation, the above example is worked in decimal;

Original sum = 2,988,492

One byte reduced and expanded = 2,981,888 or 0.2% error

Two byte reduced and expanded = 2,988,480 or 0.0004%

For most measurements, saving a one-byte representation of the three-byte sum does not introduce significant round-off error in the data.

Analog Channels

There are 64 analog channels on the FOAS that are capable of reading dc voltage signals between 0 and 5 volts. The analog to digital converters have eight bits of precision, resulting in a digital scan between 0 and 255 counts for the given dc voltage range. The reduction of data for these channels closely follows the procedures performed on the digital channels. There are some subtle differences, however.

Scans are made of all 64 analog channels each second, and the resulting bytes are added directly to a three-byte accumulator for each channel. At the end of the integration period, the scaling factors are applied to active channels and data is stored in battery-backed memory. Again, the length of the integration period, the number of bytes to be saved (either one or two bytes), and the maximum value expected in any given scan all affect the number of right shifts performed on the three-byte sums.

The record that is created at the end of the integration period has a fixed format.

Binary Record Format

The record written to the logger's memory at the end of each integration period has several distinct parts. The first nine bytes are the record's header that includes a checksum of the record, the record length in bytes, and the time stamp of the end of the integration period. These nine bytes are followed by the digital data and then the analog data.

The minimum record length is nine if no channels are turned on; and the maximum record length is

9 + 48*2 + 64*2 = 233 bytes.

The maximum record occurs when all channels are active with two bytes saved for each channel.

The format explained here is in general terms. This record format can be better understood by working through the example in Section 6.6.4. This will show you how a record is formed and how it is converted to engineering units.

The binary records written by the logger are archived in the data acquisition system in the same format as they were stored in the remote logger unit. The only change is in the file structure and organization of this data.

6.6.2 Binary Data on the PC

When data acquisition from a network of FDAS units was envisioned on personal computer equipment, the designers recognized the necessity for a compressed record format for the data archive. A natural choice was the format already implemented on the remote logger. Using this data format has several advantages for data acquisition on a PC:

- The compressed record format makes very efficient use of the limited data storage capacity of most personal computer equipment.

- Because the format of the data does not change, the chance of the data being corrupted during acquisition is minimized.

The binary data on the PC is organized by site number, parameter set code, and by the date. Each site has its own subdirectory for the storage of data. The individual binary files are stored in monthly subdirectories. The daily files have the following naming convention:

XXXXZNNN.YYb

where XXXX is the day of the decade (DOD) for this daily file

Z is the single letter parameter set code

NNN is the three-digit site number

YY is the integration period in minutes.

This file structure and naming convention has proven to be a convenient way to organize the data on the PC.

6.6.3 PCDAS System File Formats

The data collection software uses a highly organized set of driver and data files for its operations. The formats of these files are fixed to allow them to be read as random access files by the PCDAS programs. If changes are to be made to any of the file formats, the programmer must make changes to all of the programs that access the affected files.

To aid the programmer, a list of driver and data files and their respective formats has been compiled. Included in this list are the purpose or function of the file, the programs that access the file, the variables, and the format (number of characters) for each variable. The variable names listed are those shown in the FIELD statements for the programs using the files.

File Name: **SITE.DRV**

Function: Contains specific information on each site in the collection queue. This file controls the actual collection process. The operator can make changes to this file to alter the way data is collected from the network by using the QUE program.

Programs: COLLECT, PCDAS, REPORT, QUE

Variables: Q$, 1 character. This is either an "*" or a null, indicating whether or not the site is in the collection queue.

PARMC$, 1 character. An "*" or null character determining if the site should be polled.

SI$, 3 characters. The site number.

PS$, 1 character. Parameter set code.

PV$, 5 characters. The processor version identifier (ie PCDA).

PH$, 16 characters. The phone number of the site. Used in the ATDT dial-out command.

RECLEN$, 4 characters. The record length of the binary data from the logger (range 9 to 233).

INTPER$, 5 characters. The integration period in seconds.

LDA$, 5 characters. The day of the decade time stamp on the most recent record in the binary data base.

LSEC$, 6 characters. The seconds of the day time stamp on the most recent record in the binary data base.

LREC$, 6 characters. The position in the logger's memory where the most recent record was read (in decimal).

BAUD$, 4 characters. The baud rate for communications, either 300 or 1200 bits per second.

WRITEP$, 3 characters. The write-protect switch.

NSEC$, 2 characters. The number of seconds to wait during the phone line noise evaluation routine.

NCHAR$, 2 characters. The number of characters allowable in NSEC seconds.

CWAIT$, 3 characters. The number of seconds to wait for a carrier from the remote modem.

DC1$, 7 characters. Seven-character descriptor for the site.

DC2$, 7 characters. An additional seven-character descriptor.

PFULL$, 3 characters. The percentage of logger memory used needed to trigger a polling event.

DAYI$, 3 characters. The interval between pollings in days.

CTIME$, 8 characters. The time of day for the polling.

STATS$, 3 characters. Automatic statistics generator specifier.

File Name: USAGE.DAT

Function: Provides information on the data collection computer hardware. Used by most programs to access the modem and subdirectories. This is a sequential access file rather than a random access file.

Programs: PARSET, COLLECT, REPORT, INSTALL, WIDE, TRANS

Variables: USE - This is an integer number that shows the total number of runs on the COLLECT program. Used to determine media wear.

A$ - A single character determining if media protection messages should be printed.
COMNUM$ - This is the communications port for the modem.

DATAD$ - This is the path name for the binary data destination root directory.

SUMDEST$ - This is the path name of the summary message file root directory.

ADDPHONE$ - This is the dial-out prefix string.

PR - An integer defining the printer type.

ADEST$ - The path name for the ASCII data root directory

File Name: MESSAGE.DAT

Function: This is the file that contains the expanded error and summary messages. The messages are accessed using a single character search based on the entries to the individual binary message files.

Programs: COLLECT, REPORT

Variables: MESS$, 62 characters. The MESSAGE.DAT file included with V2.2 of PCDAS contains 114 messages in this random access file. Appendix A contains a complete listing of these messages.

File Name: SCHEDULE.DAT

Function: Provides scheduling information for data collection. This file can be altered with the REPORT program providing manual control of data collection scheduling. Normally, this file is acted on by the PCDAS program based on the scheduling data entered in the SITE.DRV file.

Programs: COLLECT, PCDAS, REPORT

Variables: SID$, 3 characters. The site number.

PST$, 1 character. The single-letter parameter set code.

LDATE$, 8 characters. The date of the last data collection. This is a conventional mm/dd/yy date rather than a day of the decade number.

LTIME$, 8 characters. The time of the last data collection. This is a standard hh:mm:ss time rather than seconds since midnight.

NDATE$, 8 characters. Similar to LDATE$, but records the date of the next expected data collection.

NTIME$, 8 characters. Similar to LTIME$, but records the time of the next expected data collection.

File Name: **ATTN.DAT**

Function: Keeps a record of the unread error messages. If the operator exits the PCDAS program before clearing all of the error condition messages, this file will be written.

Programs: PCDAS

Variables: SID$, 3 characters. The site number.

MFILE$, 20 characters. The message file name containing the uncleared error conditions.

File Name: **TRANS.DRV**

Function: Provides control of the data conversion process. This file is edited and updated entirely from within the TRANS program.

Programs: TRANS

Variables: Q$, 1 character. This is either an "*" character indicating that conversion is to take place on the site, or it is a blank indicating no conversion.

SI$, 3 characters. The site number.

PS$, 1 character. The single-letter parameter set code.

DAYSTART$, 4 characters. The day of the decade to start conversion.

DAYEND$, 4 characters. The day of the decade to end conversion.

OUTSPEC$, 3 characters. Determines the output specifications for the resulting ASCII file.

OUTFILE$, 30 characters. The complete specification of the output ASCII file. No naming convention applied. Includes the entire path name.

6.6.4 Data Conversion Process

The first step in the conversion process is to expand the one- or two-byte representations of the three-byte sum. The next step is to divide by the number of seconds in the integration period to obtain the average scan value during the period. The multiplicative factor (CF1) and the additive factor (CF2) are then applied to obtain the average value of the scans in engineering units.

For channels with two bytes saved, the above conversion method can be summarized by the following equation:

units = {[((byte1*256)+byte2)*(2^scale) / period] * CF1 } + CF2

For one-byte channels the conversion to engineering units can be represented by the following equation:

units = [byte1 * (2^scale) / period] * CF1 + CF2

To explain the reduction and conversion of data in a 16K logger, a detailed example has been worked. Table 1 shows the channel numbers, description of measurement, and the bytes saved from the three-byte sum. Note that the wind vane measurement has zero assigned to the Bytes Saved column. In this case, data is reduced differently. A special channel on the logger has been reserved for wind vane measurements (channel 5). A snapshot of the scan value for this channel is automatically included in the data header in byte number nine. Figure 26 shows a sample record that corresponds to Table 1. The first 9 bytes are the header and wind vane, the following 7 bytes are the digital information, and the remaining 25 bytes are the scaled analog readings.

Figure 27 shows the definition of the byte and explains the associated values. The steps in the conversion of the digital and analog data are shown in Table 2.

6.6.5 The Format of Data with Engineering Units

Remember that the data on the PC is in the same format as the data in the memory of the remote logger. The only difference is that the data on the PC is organized into daily files in a subdirectory structure with a set file name convention.

TABLE 1. Sample Use of the FDAS Data Channels

Digital Channel Number	Sensor Type	Quantity Measured	Bytes Saved	Analog Channel Number	Sensor Name		Quantity Measured	Bytes Saved
65			0	1	thermistor		indoor temp	1
66	anemometer	wind speed	1	2	thermistor		outdoor temp	1
67			0	3	pyranometer		insolation	1
68	utility KWHM	total elect.	2	4	humidity		indoor Rel H	1
69	utility KWHM	heating	2	5	wind vane		wind direct	0
70	utility KWHM	hot water	2	6	thermocouple		fireplac temp	1
71			0	7	thermistor		attic temp	1
72			0	8	reference		five volts	1
73			0	9				0
74			0	10				0
75			0	11				0
76			0	12				0
77			0	13				0
78			0	14				0
79			0	15				0
80			0	16				0
81			0	17	BNW pwr	mtr	total ph A	2
82			0	18	"	"	space heat	1
83			0	19	"	"	range	1
84			0	20	"	"	dryer	1
85			0	21	"	"	disposal	1
86			0	22	"	"	microwave	1
87			0	23	"	"	refrigerator	1
88			0	24	"	"	lites/conv.	1
89			0	25	"	"	total ph B	2
90			0	26	"	"	hot water	1
91			0	27	"	"	range	1
92			0	28	"	"	washer	1
93			0	29	"	"	dishwasher	1
94			0	30	"	"	hot tub	1
95			0	31	"	"	lites/conv.	1
96			0	32	"	"	lites/conv.	1
97			0	33				0
98			0	34				0
99			0	35				0
100			0	36				0
101			0	37				0
102			0	38				0
103			0	39				0
104			0	40				0
105			0	41				0
106			0	42				0
107			0	43				0

```
                    header                  data from digital chan.
byte number :           (5)                 (10)              (15)
data record : 010 083 041 005 156 001 067 112 128 008 003 159 001 244 001


                        data from analog channels
byte number :           (20)                (25)              (30)
data (cont) : 119 076 073 042 086 059 087 149 083 202 122 000 000 000 000


                   data form analog channels (cont.)
byte number :           (35)                (40)
data (cont) : 029 014 094 186 096 043 023 000 106 008 019
```

FIGURE 26. Sample Data Record

field	byte#	raw value	significance
check sum	1,2	2643	total of byte numbers 3 through 41
record length	3	41	record is 41 bytes in length
day number	4,5	1436	calendar date is 12/06/84
time of day	6,7,8	82800	integration period covers 22:55:00 to 23:00:00 . Units in seconds since midnite.
spare	9	128	a snap shot of the wind vane indicating that the wind direction is 180 degrees from north.

FIGURE 27. Sample Header for Data Record

The routine to convert the data to engineering units has three format options: channel identifiers and units only, channel identifiers and data, or data only. The FORTRAN format for the channel identifiers is:

```
1X
I3measurement number
1X
A11channel descriptor field
2X
A3channel identifier
2X
A6engineering units descriptor.
```

TABLE 2. Interpretation of a Data Record

Digital Channel Number	Quantity Measured	Bytes Saved	Raw Value	Scaling Factor	Scaled Raw Value	Avg Raw Value (Hz)	Calibrated Value	Units
66	Wind Speed	1	8	7	1031	3.44	260	cm/sec
68	Total Elect.	2	927	0	927	3.09	11124	Watts
69	Heating	2	500	0	500	1.66	6000	Watts
70	Hot Water	2	375	0	375	1.25	4500	Watts

Analog Channel Number	Quantity Measured	Bytes Saved	Raw Value	Scaling Factor	Scaled Raw Value	Avg Raw Scan (Counts)	Calibrated Value	Units
1	Indoor Temp.	1	76	9	38912	130	295	Kelvin
2	Outdoor Temp.	1	73	9	37376	124	285	Kelvin
3	Insolation	1	42	9	21504	71	413	Wt/cm2
4	Indoor Rel H	1	86	9	44032	146	57	Percent
5	Wind Dir	0						
6	Firepl. Temp.	1	59	9	30208	100	1960	mvolts
7	Attic Temp.	1	87	9	44544	148	297	Kelvin
8	Reference	1	149	9	76288	254	5	volts
17	Total ph A	2	21450	1	42900	143	10125	Watts
18	Space Heat	1	122	9	62464	208	2988	Watts
19	Range	1	0	9	0	0	0	Watts
20	Dryer	1	0	9	0	0	0	Watts
21	Disposal	1	0	9	0	0	0	Watts
22	Microwave	1	0	9	0	0	0	Watts
23	Refrigerator	1	29	9	14848	49	426	Watts
24	Lites/conv.	1	14	9	7168	23	107	Watts
25	Total ph B	2	24250	1	48500	161	11000	Watts
26	Hot Water	1	96	9	49152	163	2247	Watts
27	Range	1	43	9	22016	73	724	Watts
28	Washer	1	23	9	11776	39	327	Watts
29	Dishwasher	1	0	9	0	0	0	Watts
30	Hot Tub	1	106	9	54272	180	4272	Watts
31	Lites/conv.	1	8	9	4096	13	56	Watts
32	Lites/conv.	1	19	9	9728	32	293	Watts

The format for the ASCII data is:

1X I6 for each channel

Note that the first two records are header information.

REFERENCES

IBM. 1984. Disk Operating System Version 3.10 Reference. International Business Machines Corporation, Boca Raton, Florida.

IBM. 1985. BASIC Compiler Version 2.00 Fundamentals. International Business Machines Corporation, Boca Raton, Florida.

Tomich, S.D., and G.J. Schuster. 1985. End-Use Load and Conservation Assessment Program Description of Field Data Acquisition System for Building Energy Monitoring. DOE/BPA/13795-4, Bonneville Power Administration, Portland, Oregon.

APPENDIX A

ERROR AND SUMMARY MESSAGES

APPENDIX A

ERROR AND SUMMARY MESSAGES

Messages numbered 1 through 30 and 50 through 76 are error messages from the disk operating system. If you should get one of these errors during operation, note the error message, the line number, and the conditions when the error occurred. See section 1.6 on user support in the introductory chapter of this manual.

What follows is an alphabetically sorted list of all of the messages that can be written by the PCDAS COLLECT routine. In the list, the "&" character is a dummy argument that is replaced with a value in an actual message. A prefix of "E" on the message number means that the message indicates an error condition. It the prefix is "M," the message does not indicate an error condition.

MESSAGE	MESSAGE NUMBER
****** COLLECT & & & ******	M(# 31)

First message displayed when the collect program starts. Gives the site number, date, and time of polling.

****** Checking Logger Integrity ******	M(# 83)

Inticates entry to the logger integrity subroutine. This subroutine is executed before any data is collected. It determines if the logger is still collecting good data.

****** Data Acquisition Mode *******	M(# 95)

Indicates entry to the data acquisition subroutine.

****** File Manager/Archiver ******	M(# 98)

Indicates entry into the subroutine that archives the data to disk.

****** AUTOPARM & & & ******	E(# 103)

Indicates entry to the subroutine that automatically corrects parameter block errors. This subroutine is part of the logger integrity check.

****** Statistics Routine ****** M(# 110)

Indicates entry to a running statistics routine. This routine is executed after the file manager subroutine.

******Calculating data requirements ****** M(# 48)

Indicates entry to the subroutine that calculates the number of records needed from the logger.

5-volt reference must be on to compile statistics M(# 113)

Message from statistics routine. The logger's 5-volt reference channel is used by the routine to determine if a record represents a complete integration period.

15-Minute time-out on this polling - disconnect E(# 109)

If the program should hang up in a loop somewhere, the polling will terminate after 15 minutes. This guards against excessive connect time.

Abort acquisition mode: Badsum= & , Badrec= & , Ntry= & M(# 97)

If the data acquisition should fail in the middle of transmission, you will get this message. Badsum is the number of failed checksum attempts, Badrec is the number of mismatched record lengths, and Ntry is the number of transmission attempts.

Advanced feature at line & DOS error E(# 73)

Archive's time = & & ; New record's time = & & M(# 80)

Shows the time stamp of the last record in the archive and the time stamp of the record found in the logger if the last archived record cannot be found in the logger's memory. Part of the data requirements subroutine.

Bad record number at line & DOS error E(# 63)

Bad file name at line & DOS error E(# 64)

Bad file mode at line & DOS error E(# 54)

Bad file number at line & DOS error E(# 52)

Calculated records to collect = & M(# 35)

Output message from the data requirements subroutine. Shows the number of records collected by the logger since the last polling.

Calling time out -- not connected		**M(# 40)**

If the wait for carrier exceeds the number of seconds specified in the SITE.DRV file (see QUE program), then a time-out occurs.

Can't continue at line &	DOS error	E(# 17)
Cannot read paramters in three attempts - check line noise		M(# 93)

If PCDAS fails to read the logger's parameters after three attempts, it will re-evaluate the phone line noise.

Check logger clock		M(# 90)

Part of the logger integrity subroutine. The logger's clock is checked on each polling.

Clearing logger memory		E(# 107)

Part of the automatic parameter correction routine. If the parameter errors are found to cause data corruption, the logger's memory is cleared and data collection is initialized.

Clock ok, system= & & , logger = & &		M(# 91)

Shows DAS computer system time and logger's time in day of the decade and seconds since midnight.

Communications buffer overflow at line &	DOS error	E(# 69)
Correcting parameters		E(# 106)

First message in automatic parameter correction subroutine.

CPU IDs do not match		E(# 46)

Part of communications check. If the IDs do not match, the polling will be terminated.

Device unavailable at line &	DOS error	E(# 68)
Device timeout at line &	DOS error	E(# 24)
Device fault at line &	DOS error	E(# 25)
Device I/O error at line &	DOS error	E(# 57)

Dialing site - communications attempt # &		M(# 37)
Indicates that dialing has started. Shows the communications attempt number.		
Direct statement in file at line &	DOS error	E(# 66)
Disconnected from remote site		M(# 41)
Indicates that the modem has been hung up.		
Disk full at line &	DOS error	E(# 61)
Disk not ready at line &	OOS error	E(# 71)
Disk media error at line &	DOS error	E(# 72)
Disk write protect at line &	DOS error	E(# 70)
Division by zero at line &	DOS error	E(# 11)
Duplicate definition at line &	DOS error	E(# 10)
Field overflow at line &	DOS error	E(# 50)
File already exists at line &	DOS error	E(# 58)
File already open at line &	DOS error	E(# 55)
File not found at line &	DOS error	E(# 53)
FOR without NEXT at line &	DOS error	E(# 26)
Illegal direct at line &	DOS error	E(# 12)
Illegal function call at line &	DOS error	E(# 5)
Input past end at line &	DOS error	E(# 62)
Internal error at line &	DOS error	E(# 51)
Line buffer overflow at line &	DOS error	E(# 23)
Logger clock in error, site is write-protected		E(# 114)
The logger's clock is in error but was not corrected because the site is write-protected.		
Logger does not respond to commands		E(# 47)
Logger answers phone and connects but logger's processor does not respond to commands.		

Logger ID: & Site driver ID: & M(# 45)

If logger processor ID matches that found in the SITE.DRV file, then this message results.

Logger/Disk parameter comparison; & errors M(# 85)

Shows the result of the logger parameter comparison with the disk parameters. Shows the number of errors found. Part of logger integrity subroutine.

Match found between last record archived and logger M(# 81)

Indicates that the last record in the data archive was located in the logger's memory. Part of the data requirements subroutine.

Missing operand at line & DOS error E(# 22)

Modem connect M(# 38)

Indicates that the data acquisition compter's modem recieved a carrier from the logger's modem and established a connection.

NEXT without FOR at line & DOS error E(# 1)

No record length entry in the site driver file E(# 108)

This message indicates that no record length has been calculated for a site entered in the SITE.DRV file. Refer to Section 4.2 on the QUE program to find out how to perform this calculation.

No previous data for this logger/Parm set - full dump M(# 49)

Indicates that no previous data exists in the archive for the particular site and parameter set. All data will be collected.

No records collected M(# 111)

Statistics routine message. No records were collected.

No records to archive M(# 99)

Indicates that the archive subroutine had no records to archive because no records were collected.

No RESUME at line & DOS error E(# 19)

No new records since last logger polling **M(# 82)**

Indicates that no new records have been collected since the last polling.

No carrier signal detected **M(# 39)**

A "NO CARRIER" response was recieved from the data acquisition computer's modem.

No record found at last collection position **M(# 78)**

PCDAS did not find the last archived record in the logger's memory. This may be caused by overwriting or by memory being cleared. Part of Data Requirements subroutine.

Noisy phone line -- & noise characters - redial **M(# 43)**

Shows the result of the phone line test if the number of noise characters on the line exceeds the threshold set in the SITE.DRV file.

Number of records acquired = & **M(# 96)**

Shows the number of data records successfully collected from the logger. This should match the number of records expected.

Opening & for archive **M(# 101)**

Shows the file being opened for archive.

Opening & for statistics **M(# 112)**

Shows the statistics file being opened.

Out of string space at line &	DOS error	**E(# 14)**
Out of memory at line &	DOS error	**E(# 7)**
Out of paper at line &	DOS error	**E(# 27)**
Out of data at line &	DOS error	**E(# 4)**
Overflow at line &	DOS error	**E(# 6)**

Parameter errors did NOT cause data corruption **E(# 105)**

The errors found in the logger's parameter block did not affect data collection. The error will be corrected, and data collection will continue.

Parameter errors caused data corruption E(# 104)

The error found in the logger's parameter block did cause the data to be corrupted. The errors will be corrected, the logger's memory cleared, and collection re-initialized.

Parameter errors - logger is write protected - next site E(# 86)

Message indicating that parameter errors were detected but the site is write-protected. The errors will not be corrected.

Path not found at line & DOS error E(# 76)

Path/file access error at line & DOS error E(# 75)

Phone line noise evaluation M(# 42)

Indicates entry into the phone line quality evaluation routine.

Position of next record to be written to logger memory = & M(# 87)

Shows the position in the logger's memory where the next record will be written. Bottom of memory = 17408 and top of memory = 32767.

Re-sequencing on the end of the integration period M(# 102)

Indicates that the records were re-sequenced on the end of the collection interval. Part of Logger integrity subroutine.

Reading information from Site Driver file M(# 32)

Indicates that PCDAS is reading from the SITE.DRV file to get collection information for a particular site.

Reading logger parameters, attempt # & M(# 94)

Indicates that PCDAS is reading the logger's parameter block. This message also shows the attempt number. The COLLECT routine requires several parameter block reads during a normal polling.

Reading disk logger parameters from & M(# 84)

Shows the file name of disk parameter file being read by COLLECT.

Reading the last archived record from logger		**M(# 77)**

Indicates that the last archived record will be read from the logger. Part of Data Requirements subroutine.

Reconstructed time stamps of & records		**M(# 100)**

If the logger has experienced a power failure since the last polling, the clock will be zeroed. The time stamps for the records collected after the power failure will be corrected.

Record's time stamp does not match archive		**M(# 79)**

Indicates that the last archived record was not found in the logger's memory.

Rename across disks at line &	**DOS error**	**E(# 74)**
Reset Clock, system= & & ,logger= & &		**E(# 92)**

Shows the collection computer time and logger time if the clock needs to be reset.

RESUME without error at line &	**DOS error**	**E(# 20)**
RETURN without GOSUB at line &	**DOS error**	**E(# 3)**
Site not in Que	**Not used**	**(# 33)**
String too long at line &	**DOS error**	**E(# 15)**
String formula too complex at line &	**DOS error**	**E(# 16)**
Subscript out of range at line &	**DOS error**	**E(# 9)**
Syntax error at line &	**DOS error**	**E(# 2)**
Three communications attempts failed - next site		**E(# 34)**

If COLLECT cannot get data after three attempts, it gives up and goes on to the next site in the queue.

Time stamp of last record in logger memory = & &		**M(# 88)**

Shows the time stamp of the most recent record in the logger's memory. Time stamp is in day of the decade and seconds since midnight.

Too many files at line &	**DOS error**	**E(# 67)**
Type mismatch at line &	**DOS error**	**E(# 13)**

Undefined user function at line &	**DOS error**	**E(# 18)**
Undefined line number at line &	**DOS error**	**E(# 8)**
Update site driver file		**M(# 36)**

COLLECT updates the site driver file with the time stamp of the last collected record only after it has archived the data to disk.

Verify communications, attempt # & **M(# 44)**

Indicates entry to the communication verification subroutine.

Waiting until next record to be written to logger memory **M(# 89)**

If polling takes place within 5 minutes of the writing of the next record in the logger's memory, it will wait for that record.

WEND without WHILE at line &	**DOS error**	**E(# 30)**
WHILE without WEND at line &	**DOS error**	**E(# 29)**

APPENDIX B

SAMPLE NETWORK DATA FLOW CALCULATIONS

APPENDIX B

SAMPLE NETWORK DATA FLOW CALCULATIONS

There are some very simple calculations that can be performed to determine the quantity of data to be acquired for any remote metering project. Using similar calculations you can determine the maximum polling interval (time between calls) for any logger.

The calculations shown here can be used to estimate the storage space requirements for converted ASCII files. Just remember that data stored in ASCII format requires roughly six times more disk space than compressed binary files.

In all of the following calculations, the logger's record length is important. You can read this value from the first screen displayed in the QUE program if the site exists, or you can estimate it by adding 9 to the number of measurements expected to be made. In the case of data flow calculations for a whole study, you can either add all of the record lengths shown on the first screen in QUE for existing networks or estimate the total number of bytes per record for the whole network. Do this by estimating the number of measurement points in the study and adding 9 times the number of FDAS data loggers.

The integration period is another important number in these calculations. The value is always in seconds (3600 for hourly, 900 for 15-minute intervals).

The number of bytes available in a 16K FDAS is a constant 15360 bytes.

1. Calculation of maximum polling interval

MRIL = Maximum Records In Logger = Integer(15360/record length)

PI = Polling Interval in days = Integration period * MRIL / 86400

Example:

Record length	56 bytes
Integration period	3600 seconds

MRIL = INT(15360/56) = 274 records

PI = 3600 * 274 / 86400 = 11.4 days

2. Calculation of data storage requirements for an entire network

Example:

Study length	2 years
Integration period	3600 seconds
Total record length	200 bytes per hour

Total records in study = (24 hours/day)(365 days/yr)(2 yr)
= 17520 hours

Total number of bytes in study = 17520 * 200 = 3.5 million bytes

APPENDIX C

FDAS SAMPLING METHOD AND SAMPLE CONVERSION FACTORS

APPENDIX C

FDAS SAMPLING METHOD AND SAMPLE CONVERSION FACTORS

Before we discuss data conversion, it is important that you have an understanding of the FDAS sampling method. The FDAS accepts both analog and digital input signals, so this discussion will be divided into two parts.

Analog channels:

The 16k byte FDAS is equipped with 8 bit analog to digital converters. Electrical signals (0 to 5 Volts dc) are converted to binary numbers. Every second, a reading from the A/D converters (0 to 255) for each channel is added to a 3-byte accumulator. At the end of the integration period, a number of right shifts are performed on this accumulator. Only the least significant bits are truncated in this scaling process. The number of right shifts performed is called the scaling factor.

Depending on the desired accuracy, either the most significant byte in the accumulator (1-byte accurcay) or the most significant two bytes in the accumulator (2-byte accuracy) are written to data memory for each active channel. This scaled summation of readings is transmitted to the data collection computer to be archived.

When the data is converted, any offset counts are subtracted and the scaling factor is applied in reverse to obtain the original summation value. This value is divided by the number of seconds in the integration period to give the average reading (0 to 255 counts) over that interval. CF1 and CF2 are applied to this average reading.

Sample CF1 and CF2 calculations follow

Energy usage:

Example : -Analog watt-meter channel
-400A current transformer
-240 Volts AC service
-12 Counts offset for circuit (from calibration sheet)
-100 k ohm scaling resistor
-1.307e-3 Volts AC/cnt (channel sensitivity from cal.sheet)

In general,

CF1 = (Channel sensitivity/0.333)(CT scaling)(CT rating * Voltage)
CF1 = (Volts AC/cnt/0.333)(R/standard kohm)(CT rating * Voltage)

For this example,

CF1 = (1.307e-3/0.333) (100/100) (400 x 240) = 376.8 watts/count
CF2 = 0
OFF = 12 counts.

Example : -Analog watt meter channel
-100A current transformer
-120 Volts AC service
-11 Counts offset (from calibration sheet for channel)
-200 kohm scaling resistor
-1.298e-3 Volts AC/cnt (from calibration sheet)

CF1 = (1.298e-3/0.333) (100/200) (100 x 120) = 23.39 watts/count
CF2 = 0
OFF = 11 counts.

Example : -Digital (KCD) watt meter channel
-2000:5 standard current reduction CT with 5A CT on the large CT secondary
-480/277 Volts AC service
-12 count offset from the calibration sheet
-20 kohm scale resistor
-1.296e-3 Volts AC/cnt channel sensitivity

CF1 = (1.296e-3/0.333) (10kohm/20kohm) (2000 X 277)
CF1 = 1078.1 watts/count

CF2 = 0
OFF = 12 counts.

<u>Environmental conditions:</u>

Example: Relative humidity sensor
0 to 100% humidity = 0 to 5 Volts dc

CF1 = 100%/5 Volts dc = 100%/255 counts = 0.392%/count
CF2 = 0
OFF = 0 counts.

Example: Wind direction sensor
0 to 360 degree clockwise from north = 0 to 5 Volts dc

CF1 = 360 degrees / 255 counts = 1.412 degrees/count
CF2 = 0
OFF = 0 counts.

In general, CF1 is the slope of the line relating the measured quantity with counts. This slope or response can be determined simply in many cases like the wind direction or humidity sensor, or may rely on empirical data from the laboratory. In each case the units of CF1 are:

Engineering units/count

CF2 is the offset in engineering units. It most cases it will be zero.

Digital Channels

The FDAS reduces digital data in a slightly different way than the analog data. A 5-Volts dc pulse on any of the digital channels adds one to a single-byte temporary accumulator for the channel. Every 3 seconds these temporary accumulators are added to a three-byte accumulator just like the analog channels. At the end of the integration period, this three-byte sum is scaled and written to memory just like the analog sums. A header of nine bytes is included with the scaled analog and digital sums to make a data record.

As with the analog readings, either one or two bytes of the digital summation can be written to the record. These records of scaled summations are transmitted to the data collection computer for archiving.

When the data is converted to engineering units, the scaling is applied in reverse and the resulting number is divided by the integration period, leaving a number representing the average cycles per second over the sampling period. The CF1 constant is the conversion factor between the average Hz and the engineering value.

Energy measurements

Example: Pulse-initiating watt-hour meter
7.2 watt-hours per pulse
3-wire form C contact
2-wire measurement

In a two-wire measurement, only half of the total pulses are monitored, so each pulse actually represents 14.4 watt-hours of energy used.

CF1 = (watt-hours/pulse) (3600 seconds/hour)

CF1 = (14.4)(3600) = 518400 watt-seconds/pulse
CF2 = 0.

Engineering units for the converted data will be

(watt seconds/pulse) (avg pulses/second) = avg watts for interval.

Flow measurement

Example: Pulse-initiating flow-meter 1 pulse per gallon
Form C 3-wire contact closure
Monitoring only 2-wire contact closure

CF1 = (gallons/pulse) (integration period)

CF1 = (2)(3600) = 7200 gallons seconds/pulse.
Engineering units are in total gallons used over the interval.

(gallons seconds/pulse) (avg pulses/second) = gallons used.

Scaling factor considerations:

In both digital and analog channels, a number of right shifts are performed to scale the three-byte sums. The number of right shifts depends on the integration period, the maximum expected scan value for the channel, and the number of bytes to save (one or two). If the maximun possible reading is small, then a great deal of accuracy may be truncated if you do not select an appropriate scaling factor.

In the case of the flow meter shown above, it is possible that the logger may have to account for every pulse over the integration period (no truncation error). The scale factor for a channel must be zero to eliminate truncation error. Do this by adjusting the maximum value in the MVL column using the PARSET program until the calulated scaling factor in the SF column is zero. In most applications, the truncation error cause by the scaling of analog data is insignificant.

In the case of the pulse-initiating watt-hour meters, it is also important to adjust the scaling factor using the MVL column. It is important to set the maximum value to the highest possible pulse rate. For this case a maximum could be calulated from the size of the circuit breaker on the service with the watt meter. For a 200A 120 Volt service

(200A X 120 Volts) / (14.4 watt-hour/pulse) = 1667 pulses/hour

= 0.436 pulses/sec

thus, in this case, the maximum pulse rate is less than 1 Hz. If this pulse rate is entered as the maximum value, then PARSET will automatically calculate the correct scaling factor.

Note that the maximum value (MVL) for analog channels is in counts and the maximum value for digital channels is in Hz.

APPENDIX D

PCDX PROCESSOR COMMAND SET

APPENDIX D

PCDX PROCESSOR COMMAND SET

The following is a list of the single-character commands that are used in the PCDA and PCDB version of the FDAS firmware. These commands are used by the PCDAS software to acquire data from the logger and control the data collection. The command characters can also be issued to the logger with a dumb terminal. Several commands are used primarily by the data acquisition system to control the logger and are marked with a "W" as a warning.

** WARNING **

Issuing command marked with a "W" using a terminal may interrupt normal FDAS control using PCDAS.

Command Char	ASCII equiv	W	Function
;	59	W	Subtracts 1 from a parameter or parameter block pointer.
<	60	W	Adds 1 to a parameter or parameter block pointer.
=	61	W	Adds 10 to a parameter or parameter block pointer.
>	62	W	Adds 50 to a parameter or parameter block pointer.
"	34		Displays memory contents of locations addressed by the start address and range in ASCII.
'	39		Loads the start address and range.
%	37	W	Control command to clear memory.
.	46	W	Execute control command. Two characters are normally required to execute a logger contol command.
$	36		Returns the processor ID code in binary and the the firmware version ID in ASCII.
,	44		Displays memory contents of locations addressed by the start address and range in binary.
@	64		Gives continuous scans of all 64 analog channels in ASCII. Units of scans are counts (0 to 255).
&	38	W	Moves the logger into the parameter change mode. Displays 16 bytes in ASCII showing the designated parameter.

^	94	W	Control command to clear the parameter block.
*	42	W	Control command to cause a warm reset of the logger.
#	35		Displays the parameter block in ASCII.
[	91		Transmits the first data record in memory in binary.
\	92		Transmits the next data record in binary.
]	93		Re-transmits the previous record.
(	40		Sets pointer to first byte in start address.
)	41		Moves pointer to next byte.

APPENDIX E

SAMPLE DATA CAPTURE RATE REPORT

APPENDIX E

SAMPLE DATA CAPTURE RATE REPORT

Data Capture Report for Site Number 001
for 1986

Week	SUN	MON	TUE	WED	THR	FRI	SAT	Month
12/29 - 1/ 4								
1/ 5 - 1/11								
1/12 - 1/18								
1/19 - 1/25								
1/26 - 2/ 1								1-> 0
1/26 - 2/ 1								
2/ 2 - 2/ 8								
2/ 9 - 2/15								
2/16 - 2/22								
2/23 - 3/ 1								2-> 0
2/23 - 3/ 1								
3/ 2 - 3/ 8							A014A	
3/ 9 - 3/15	A100A	A100A	A100A	A083A	A100A	A100A	A100A	
3/16 - 3/22	A100A	A100A	A100A	A100A	A100A	A017B	B100B	
3/23 - 3/29	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
3/30 - 4/ 5	B100B	B100B						3-> 71.4
3/30 - 4/ 5			B100B	B100B	B100B	B100B	B100B	
4/ 6 - 4/12	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
4/13 - 4/19	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
4/20 - 4/26	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
4/27 - 5/ 3	B100B	B100B	B100B	B100B				4-> 100
4/27 - 5/ 3					B100B	B100B	B100B	
5/ 4 - 5/10	B100B	B100B	B079B					
5/11 - 5/17			B008B	B100B	B100B	B100B	B100B	
5/18 - 5/24	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
5/25 - 5/31	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
6/ 1 - 6/ 7								5-> 77.02
6/ 1 - 6/ 7	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
6/ 8 - 6/14	B100B	B096B	B100B	B100B	B100B	B100B	B100B	
6/15 - 6/21	B100B	B100B	B100B	B104B	B096B	B100B	B100B	
6/22 - 6/28	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
6/29 - 7/ 5	B100B	B100B						6-> 99.86
6/29 - 7/ 5			B100B	B100B	B100B	B100B	B100B	
7/ 6 - 7/12	B100B	B100B	B100B	B100B	B100B	B100B	B100B	
7/13 - 7/19	B100B	B100B	B100B	B100B	B100B	B100B	B100B	

```
 7/20 -  7/26 | B100B B100B B100B B100B B100B B100B B100B |
 7/27 -  8/ 2 | B100B B100B B100B B100B B100B             |  7-> 100
--------------|-------------------------------------------|
 7/27 -  8/ 2 |                               B100B B050B |
 8/ 3 -  8/ 9 | B100B B100B B100B B100B B100B B100B B100B |
 8/10 -  8/16 | B100B B100B B096B B100B B100B B100B B100B |
 8/17 -  8/23 | B100B B100B B100B B100B B100B B100B B100B |
 8/24 -  8/30 | B100B B100B B100B B100B B100B B100B B100B |
 8/31 -  9/ 6 | B100B                                     |  8-> 98.25
--------------|-------------------------------------------|
 8/31 -  9/ 6 |       B100B B100B B100B B100B B100B B100B |
 9/ 7 -  9/13 | B100B B100B B100B B100B B100B B100B B100B |
 9/14 -  9/20 | B100B B100B B100B B100B B100B B100B B100B |
 9/21 -  9/27 | B100B B100B B100B B100B B100B B100B B100B |
 9/28 - 10/ 4 | B100B B100B B100B                         |  9-> 100
--------------|-------------------------------------------|
 9/28 - 10/ 4 |                   B100B B100B B100B B100B |
10/ 5 - 10/11 | B100B B100B B100B B100B B100B B100B B100B |
10/12 - 10/18 | B100B B100B B100B B100B B100B B100B B100B |
10/19 - 10/25 | B100B B100B B100B B100B B100B B100B B100B |
10/26 - 11/ 1 | B100B B100B B100B B100B B100B B100B       |  10-> 100
--------------|-------------------------------------------|
10/26 - 11/ 1 |                                     B100B |
11/ 2 - 11/ 8 | B100B B100B B100B B100B B100B B100B B100B |
11/ 9 - 11/15 | B100B B100B B100B B100B B067B       B038B |
11/16 - 11/22 | B104B B100B B100B B100B B100B             |
11/23 - 11/29 |       B008B B100B B096B B100B B100B B100B |
11/30 - 12/ 6 | B100B                                     |  11-> 80.42

--------------|-------------------------------------------|
11/30 - 12/ 6 |       B100B B100B B100B B100B B100B B100B |
12/ 7 - 12/13 | B100B B100B B100B B100B B100B B100B B100B |
12/14 - 12/20 | B100B B100B B100B B100B B100B B100B B100B |
12/21 - 12/27 | B100B B100B B100B B100B B100B B100B B100B |
12/28 -  1/ 3 | B100B                                     |  12->-1.43
-------------------------------------------------------------------
Data Capture rate for  1986  is  68.79333

Parameter sets used this year
Parm Set   Reclen    Intper
--------   ------    ------
  A          43        15
  B          39        60
```

APPENDIX F

SAMPLE ASCII DATA FILE

APPENDIX F

SAMPLE ASCII DATA FILE

					DEG F	DEG F	DEG F	COUNTS	WATTS
LOG	DAY	SEC	MIN	DQA	C01	C02	C06	C08	C17
1	2272	895	15	0	93	92	102	255	51
1	2272	1795	30	0	93	92	102	255	51
1	2272	2695	45	0	93	92	102	255	51
1	2272	3595	60	0	93	92	102	255	51
1	2272	4495	75	0	92	92	102	255	51
1	2272	5395	90	0	92	92	102	255	51
1	2272	6295	105	0	92	92	102	255	51
1	2272	7195	120	0	91	91	102	255	51
1	2272	8095	135	0	91	91	101	255	51
1	2272	8995	150	0	91	91	101	255	51
1	2272	9895	165	0	91	91	101	255	51
1	2272	10795	180	0	91	91	101	255	51
1	2272	11695	195	0	90	91	100	255	51
1	2272	12595	210	0	90	91	100	255	51
1	2272	13495	225	0	90	90	100	255	51
1	2272	14395	240	0	90	90	100	255	51
1	2272	15295	255	0	90	90	100	255	51
1	2272	16195	270	0	90	90	100	255	51
1	2272	17095	285	0	89	90	99	255	51
1	2272	17995	300	0	89	90	99	255	51
1	2272	18895	315	0	89	89	99	255	51
1	2272	19795	330	0	89	89	99	255	51
1	2272	20695	345	0	89	89	99	255	51
1	2272	21595	360	0	89	89	99	255	51
1	2272	22495	375	0	89	89	99	255	51
1	2272	23395	390	0	89	89	99	255	51
1	2272	24295	405	0	89	89	99	255	51
1	2272	25195	420	0	89	89	99	255	51
1	2272	26095	435	0	89	89	99	255	51
1	2272	26995	450	0	89	89	98	255	51
1	2272	27895	465	0	89	89	98	255	51
1	2272	28795	480	0	89	89	98	255	51
1	2272	29695	495	0	89	89	98	255	51
1	2272	30595	510	0	89	89	98	255	51
1	2272	31495	525	0	89	89	98	255	51
1	2272	32395	540	0	88	89	98	255	51
1	2272	33295	555	0	88	89	98	255	51
1	2272	34195	570	0	88	89	98	255	51
1	2272	35095	585	0	88	88	98	255	51
1	2272	35995	600	0	88	88	98	255	51
1	2272	36895	615	0	88	88	98	255	51
1	2272	37795	630	0	88	88	98	255	51
1	2272	38695	645	0	88	88	98	255	51

APPENDIX G

SAMPLE CKSUM PROGRAM OUTPUT

APPENDIX G

SAMPLE CKSUM PROGRAM OUTPUT

```
Check sum equation checker.  Version 1.0
02-05-1987   22:25:25

Channel data file :2WID2052.DAT
Check sum equation file :SITEDAT\BSITE002.SUM
Output file name :2WID2052.CKS
```

1	5VREF	C08	COUNTS	1.1
2	PNLC&DMAINA	C09	WATTS	430.8
3	PNLC&DMAINB	C10	WATTS	433.4
4	PNLC&DMAINC	C11	WATTS	430.8
5	EXHAUSTFANA	C12	WATTS	75.3
6	EXHAUSTFANB	C13	WATTS	74.7
7	EXHAUSTFANC	C14	WATTS	75.0
8	REFRIGA	C15	WATTS	43.4
9	REFRIGB	C16	WATTS	43.5
10	REFRIG-C	C17	WATTS	43.1
11	KITPLUGS-A	C18	WATTS	106.4
12	KITPLUGS-B	C19	WATTS	107.1
13	KITPLUGS-C	C20	WATTS	107.0
14	STMCOOKEQ-A	C21	WATTS	251.7
15	STMCOOKEQ-B	C22	WATTS	251.7
16	STMCOOKEQ-C	C23	WATTS	252.9
17	RT FRYER-A	C24	WATTS	39.4
18	RT FRYER-B	C25	WATTS	39.9
19	TILTGRILL-A	C26	WATTS	74.8
20	TILTGRILL-B	C27	WATTS	74.7
21	H&V UNIT-A	C28	WATTS	40.0
22	H&V UNIT-B	C29	WATTS	53.1
23	H&V UNIT-C	C30	WATTS	53.1
24	STACKOVEN-A	C31	WATTS	74.3
25	STACKOVEN-B	C32	WATTS	75.0
26	STACKOVEN-C	C33	WATTS	74.0
27	MEALLINE-C	C34	WATTS	22.3
28	MEALLINE-A	C35	WATTS	22.2
29	PNLFIJMAINA	C36	WATTS	424.6
30	PNLFIJMAINB	C37	WATTS	429.8
31	PNLFIJMAINC	C38	WATTS	424.3
32	TILTGRILL-A	C39	WATTS	40.0
33	TILTGRILL-B	C40	WATTS	40.0
34	TILTGRILL-C	C41	WATTS	40.3
35	LFTFRYER-B	C42	WATTS	39.9
36	LFTFRYER-C	C43	WATTS	39.9
37	DSHWSH-A	C44	WATTS	213.6
38	DSHWSH-B	C45	WATTS	213.6

39	DSHWSH-C	C46	WATTS	212.7
40	EXHFANS-A	C47	WATTS	42.8
41	EXHFANS-B	C48	WATTS	33.7
42	EXHFANS-C	C49	WATTS	52.7
43	SNAKLINE-A	C50	WATTS	135.7
44	SNAKLINE-B	C51	WATTS	135.2
45	SNAKLINE-C	C52	WATTS	106.8
46	SNAKGRID-A	C53	WATTS	52.7
47	SNAKGRID-B	C54	WATTS	53.4
48	SNAKGRID-C	C55	WATTS	52.9
49	MISC FRIG-B	C56	WATTS	26.8
50	MISC FRIG-C	C57	WATTS	26.8
51	MEALLINE-A	C58	WATTS	160.3
52	MEALLINE-B	C59	WATTS	159.3
53	MEALLINE-C	C60	WATTS	160.4
54	MEALGRID-A	C61	WATTS	52.7
55	MEALGRID-B	C62	WATTS	53.0
56	MEALGRID-C	C63	WATTS	52.5
57	MISC FRIG-A	C64	WATTS	26.4

Check Sum equations and resolutions :

C: C11 = C14 + C17 + C20 + C23 + C30 + C33 + C57   res: 430.8

General max/min information

No records have been excluded due to suspected reversed CTs.
No channel rescaling was selected
Negative channel data has been treated as zeros.

Channel	Minima	Min Res.	Maxima	Max Res.
8	255	232	255	232
9	9048	21	30592	71
10	8236	19	28177	65
11	7756	18	23267	54
12	603	8	678	9
13	448	6	523	7
14	675	9	750	10
15	1868	43	3433	79
16	1958	45	3394	78
17	1769	41	3106	72
18	106	1	958	9
19	107	1	214	2
20	0	0	857	8
21	2014	8	15607	62
22	1510	6	11831	47
23	1770	7	12393	49
24	0	0	3353	85
25	0	0	3279	82
26	0	D	8905	119

27	0	0	8891	119
28	1242	31	1522	38
29	1276	24	1595	30
30	1648	31	1967	37
31	0	0	5797	78
32	0	0	4278	57
33	0	0	6441	87
34	1205	54	1540	69
35	534	24	1001	45
36	849	2	22930	54
37	1290	3	27943	65
38	849	2	23761	56
39	0	0	921	23
40	0	0	3561	89
41	0	0	3669	91
42	0	0	4399	110
43	0	0	4559	114
44	0	0	15812	74
45	0	0	15385	72
46	0	0	12762	60
47	214	5	471	11
48	169	5	438	13
49	370	7	634	12
50	407	3	4344	32
51	271	2	4329	32
52	0	0	1603	15
53	0	0	1848	35
54	0	0	1817	34
55	0	0	2965	56
56	54	2	1154	43
57	885	33	1905	71
58	0	0	2565	16
59	478	3	5417	34
60	321	2	4973	31
61	0	0	3533	67
62	0	0	3025	57
63	0	0	2732	52
64	874	33	1933	73

General Data Analysis

2.75 Days of data
There were 0 dead channels

1 bad records / 66 good records 1.52 %

1 records excluded due to bad reference voltage readings
0 records excluded due to suspected reversed CTs

DIFFERENCE RECOROS FOR EQUATION C

Oay	Min	Main	Feed	Oiff	% Oiff	Channels
2052	120	12495	12972	-477	4	14 17 23 30 33 57
2052	240	8187	8733	-546	7	14 17 23 30 33 57
2052	480	20682	21337	-655	3	14 17 20 23 30 33 57
2052	720	15942	16479	-537	3	14 17 20 23 30 33 57
2052	900	10341	10931	-590	6	14 17 23 30 33 57
2052	1020	11634	12112	-478	4	14 17 23 30 33 57
2052	1080	10772	11377	-605	6	14 17 23 30 33 57
2052	1140	10772	11288	-516	5	14 17 23 30 33 57
2052	1200	8187	8710	-523	6	14 17 23 30 33 57
2052	1260	10772	11234	-462	4	14 17 23 30 33 57
2052	1320	13788	14280	-492	4	14 17 23 30 33 57
2052	1380	10772	11448	-676	6	14 17 23 30 57
2052	1440	8618	9315	-697	8	14 17 23 30 33 57
2053	120	8618	9156	-538	6	14 17 23 30 33 57
2053	540	12065	12644	-579	5	14 17 20 23 30 33 57
2053	600	19820	20308	-488	2	14 17 20 23 30 33 57
2053	720	12065	12700	-635	5	14 17 20 23 30 33 57
2053	780	10772	11414	-642	6	14 17 20 23 30 33 57
2053	840	14650	15138	-488	3	14 17 20 23 30 33 57
2053	900	16804	17340	-536	3	14 17 20 23 30 33 57
2053	1020	10772	11429	-657	6	14 17 20 23 30 33 57
2053	1200	11634	12157	-523	4	14 17 20 23 30 33 57
2053	1260	11203	11912	-709	6	14 17 20 23 30 33 57
2053	1380	7756	8245	-489	6	14 17 20 23 30 57
2054	300	8187	8663	-476	6	14 17 20 23 30 33 57
2054	480	19389	19830	-441	2	14 17 20 23 30 33 57
2054	540	21975	22723	-748	3	14 17 20 23 30 33 57
2054	600	21544	22127	-583	3	14 17 20 23 30 33 57
2054	720	17235	17808	-573	3	14 17 20 23 30 33 57
2054	900	15942	16538	-596	4	14 17 20 23 30 33 57
2054	960	11634	12294	-660	6	14 17 20 23 30 33 57
2054	1080	9910	10553	-643	6	14 17 20 23 30 33 57

Data Analysis for equation C

Distribution of failed records

% Negative:	48.48
% Positive:	0.00
Total :	48.48

Average # of resolutions difference ALL records

Negative:	1.003
Positive:	0.000
Both :	1.003

Average # of resolutions difference for FAILED records

Negative:	1.324
Positive:	0.000
Both :	1.324

Maximum # of resolutions difference for ALL records

	Res	Last occurred at Day	Minute	# of occur
Negative:	1.74	2054	540	1
Positive:	0.00	0	0	1

```
   Percent time channel is on when | diff | > resolution

   %            10   20   30   40   50   60   70   80   90   100
              |--------------------------------------------------
100.0 C11     |**************************************************
100.0 C14     |**************************************************
100.0 C17     |**************************************************
 62.5 C20     |*******************************
100.0 C23     |**************************************************
100.0 C30     |**************************************************
 93.8 C33     |***********************************************
100.0 C57     |**************************************************


   Percent time | diff | > resolution when channel is on

   %            10   20   30   40   50   60   70   80   90   100
              |--------------------------------------------------
 48.5 C11     |************************
 48.5 C14     |************************
 48.5 C17     |************************
 42.6 C20     |*********************
 48.5 C23     |************************
 48.5 C30     |************************
 50.8 C33     |*************************
 48.5 C57     |************************


   Distribution of errors given in # or resolutions, Average:  -1.003

   %      bin              10   20   30   40   50   60   70   80   90
100
                        |--------------------------------------------------
  0.0 [ -2.5,  -2.0]    |
 10.6 [ -2.0,  -1.5]    |*****
 37.9 [ -1.5,  -1.0]    |*******************
 42.4 [ -1.0,  -0.5]    |*********************
  9.1 [ -0.5,   0.0]    |****
  0.0 [  0.0,   0.5]    |
  0.0 [  0.5,   1.0]    |
  0.0 [  1.0,   1.5]    |
  0.0 [  1.5,   2.0]    |
  0.0 [  2.0,   2.5]    |
```

Most Extreme Differences for equation C
(based on % time |diff| > resolution when channel is on)

Fewer than 5 positive differences

Most negative differences (40 % criterion)

Ratios: (power + difference/power)

Records	1	2	3	4	5
Day	2054	2053	2052	2052	2054
Min	540	1260	1440	1380	960
Diff	-748	-709	-697	-676	-660
C11	1.034	1.063	1.081	1.063	1.057
C14	0.003	0.055	0.071	0.099	0.120
C17	0.729	0.707	0.712	0.725	0.778
C20	0.127	-0.325	0.000	0.000	0.230
C23	0.940	0.852	0.606	0.852	0.739
C30	0.574	0.619	0.614	0.626	0.612
C33	0.703	-0.597	-0.047	0.000	0.595
C57	0.557	0.355	0.634	0.640	0.643

Power: (channel values)

Records	1	2	3	4	5
Day	2054	2053	2052	2052	2054
Min	540	1260	1440	1380	960
Diff	-748	-709	-697	-676	-660
C11	21975	11203	8618	10772	11634
C14	750	750	750	750	750
C17	2761	2416	2416	2459	2977
C20	857	535	0	0	857
C23	12393	4806	1770	4553	2529
C30	1755	1861	1808	1808	1701
C33	2517	444	666	0	1629
C57	1690	1100	1905	1878	1851

DISTRIBUTION

No. of
Copies

OFFSITE

2 DOE Technical Information Center

Mashuri Warren
Lawrence Berkeley Laboratory
Building Energy Systems Program
Building 90-2024
Berkeley, CA 94720

Dean Devine
Federal Energy Management Program
Forrestal Bld.
Washington, DC 20545

ONSITE

DOE Richland Operations Office

J.J. Sutey

5 Westinghouse Hanford Company

J.E. Uecker (3)
W.W. Leonard
R.S. Poplin

31 Pacific Northwest Laboratory

S.A. Bailey
P.J. Cowley
M.J. Fishbaugher (10)
S.G. Hauser
D.D. Hostetler
R.P. Mazzucchi (5)
R.G. Pratt
R.W. Reilly
G.M. Stokes
R.A. Stokes
S.A. Weakly
Publishing Coordination (2)
Technical Report Files (5)