Analyzer Interface Specification

CD TOC EXIT

Click on Item Below:

ABBOTT AD$_x$ Version 2

ABBOTT DIAGNOSTICS
A Division Of Abbott Laboratories
Abbott Park, IL 60064 U.S.A.

LN 9684-49 45632-102

AD$_x$® Analyzer Interface Specification

PROPRIETARY INFORMATION

Table of Contents

AD_x® Analyzer Interface Specification

Table of Contents

Page

Section 1 Overview

Starting with software revision V2.3, the AD_x® analyzer has the ability to connect to a remote computer. The remote computer can be a Laboratory Information System (LIS) or a simple personal computer (PC).

Once connected, the AD_x analyzer can send the results of the latest tests to the computer for further analysis. The computer can also send sample identification information to the AD_x analyzer—eliminating the need to enter the IDs by hand.

The intent of this document is to provide the technical details of the connection.While the “Overview” will be directed at the casual reader, the rest of the document is intended for a computer literate audience.

CONNECTIONS

There’s more to making the connection work than just stringing a wire between the AD_x analyzer and the computer. In this document we’ll break the connection into two parts: the “physical” connection and the “logical” connection.

The physical connection is the easiest to understand. It simply entails the details required to connect a cable from your computer to the back of the AD_x analyzer. The AD_x analyzer uses this cable to send data to and receive data from the computer. The section “Serial Port Specification” will give you all the details on the physical connection.

The logical connection means that it is not enough merely to connect the AD_x analyzer and the computer with a cable. Software for your computer will have to be purchased or written so your computer can interpret the data it receives from the AD_x analyzer.

Even the logical connection can be broken into two separate topics: collecting the data and interpreting the data.Your computer will need special software to be able to receive the data from the AD_x analyzer. The AD_x analyzer packages the data in a special way to ensure that it is error free. Even then, once you’ve collected the data, more software will be required to interpret the data. Your software will need to know how to extract the test results, the sample IDs and other important information from the data.

More information on collecting the data will be found in the “Protocols” and the “Abbott Standard Protocol” sections. Instructions on interpreting the AD_x analyzer data will be in the section “Catching Result Data.”

Overview

YOUR COMPUTER

Throughout the document, we'll make references to a "remote computer," or a "host computer" or a "host" or even just a "computer". They all mean the same thing: the computer system that you connect to the AD_x analyzer.

Your laboratory may have a large, expensive computer system that's tucked away in the basement. Or it may have a personal computer in a corner—it doesn't matter.

What kind of computer you have or where it is or how large it is, is not important. Whatever your system is, the AD_x analyzer has several features that should make the connection easy. The first feature is the "serial port" and the second is the "protocol."

SERIAL PORTS

You'll use a special cable to connect the AD_x analyzer to your computer system. One end of the cable is plugged into your system and the other to the serial port on the back of the AD_x analyzer. The AD_x analyzer uses a standard type of serial port that should make it easy to connect the two together.

This standard is known as the RS232-C standard and is almost universally used on personal computers and most of the larger systems.

PROTOCOLS

In order for the AD_x analyzer to exchange information with your computer, the two systems must agree to use a common convention. The convention governs the exchange of data with rules like: "only one side sends information at a time" and "send the data in small chunks." This convention, when put into computer terms, is called a "protocol."

When the AD_x analyzer sends the test results to your computer, it will use a protocol and expect your system to respond in kind. By using a protocol, both sides can ensure that the data is transferred without error.

Your computer system will have to have software on it that knows how to use the protocol to communicate with the AD_x analyzer. The protocol that the AD_x analyzer uses is known as the Kermit File Transfer Protocol.

The serial port is the portion of the AD_x analyzer hardware that lets it transmit and receive data. This is the most common interface used in data communications.

This next section requires the reader to be familiar with serial ports, interfacing techniques and telecommunication basics. You should also be familiar with the operation of the AD_x analyzer.

THE AD_X ANALYZER

The AD_x analyzer is an automated system which can perform toxicology or abused drug assays. The AD_x analyzer is designed for use by trained personnel in the hospital laboratory, private laboratory, physicians office laboratory, employee health facility, drug rehabilitation center or other facilities that perform abused drug testing.

The analyzer is typically placed on a bench top that can accommodate its height of 11 inches, length of 25.5 inches and depth of 18 inches.The serial port connection is on the rear panel of the analyzer. Refer to figures below.

insert illustration

ADDITIONAL INFORMATION

You'll also want to reference the AD_x Operator's Guide as you read on. To receive more information about the operation of the AD_x analyzer, write to: Abbott Diagnostics, AD_x Marketing, Abbott Park, Illinois 60064 U.S.A.

Section 2 Serial Port Specification

HARDWARE

The AD_X analyzer serial port supports the RS232-C standard, the most common interface found in data communications. Refer to the EIA Standard RS232-C for more information.

The AD_X analyzer port is a DB-25 female connector and is configured as a DCE device. The AD_X analyzer will operate with the minimum number of pins connected: 1, 2, 3, and 7.

ANALYZER DCE	HOST DTE
1 GND	GND 1
2 TXD	RXD 2
3 RXD	TXD 3
7 SIG GND	SIG GND 7

ANALYZER DCE	HOST DCE
1 GND	GND 1
2 TXD	TXD 2
3 RXD	RXD 3
7 SIG GND	SIG GND 7

The AD_X analyzer will assert the CTS and DSR lines when it is powered up, fully configured and ready to accept data.

Do NOT send data to the AD_X analyzer before it asserts the CTS and DSR lines. The host system can either monitor the status of these control lines or simply wait several seconds after the AD_X analyzer is powered up.

DATA FORMAT

The serial interface is asynchronous and the data is sent using 8 data bits, 1 stop bit, 1 start bit and no parity. The host system must be configured in a like fashion. The AD_X analyzer sends and expects to receive data using the ASCII character set.

BAUD RATE

The AD_X analyzer baud rate may be set to one of the following: 300, 1200, 2400, 4800 or 9600 baud. Use the AD_X analyzer SYSTEM... SERIAL PORT... SET UP... menu to find the BAUD RATE parameter and edit it accordingly.

Note:

IT IS IMPORTANT THAT THE DATA FORMAT AND BAUD RATE MATCH EXACTLY BETWEEN THE TWO SYSTEMS. FAILURE TO DO SO WILL CAUSE THE 8274 UART IN THE AD_X ANALYZER TO CORRECT ITSELF CONSTANTLY AND PUT UNNECESSARY BURDEN ON THE AD_X ANALYZER.

Serial Port Specification

FLOW CONTROL

The AD_X analyzer does NOT monitor any of the hardware control lines, as in DTR, RTS, CTS. Therefore hardware flow control cannot be used.

The AD_X analyzer is set up to use XON/XOFF (control-Q, control-S) flow control but this is DISABLED during the Kermit File Transfer Protocol.

Flow control is required when the AD_X analyzer sends information at a rate that is too high for the host system. Because of the variable packet size, and half duplex nature of the Kermit protocol, XON/XOFF flow control is not required.

DATA BUFFERING

The AD_X analyzer buffers transmitted and received data. Excess data due to a buffer overflow is lost. However, the buffers are several times larger than the maximum Kermit packet size so the buffers will not overflow.

DATA FILTERING

The AD_X analyzer will filter the data that it receives from a host in an attempt to keep nonsense characters from reaching the printer. Certain control characters can affect the output of the AD_X analyzer printer. The AD_X analyzer will also convert lowercase letters to uppercase.

Other characters are filtered from the input stream. Consult the section "Nonsense Data/Data Filtering" for specific information.

Section 3 The Abbott Standard Protocol

The AD_x analyzer uses a file transfer protocol to send data to a remote computer. The AD_x analyzer expects the remote computer to use the same protocol when sending data to it. The protocol used is the *Abbott Standard Protocol* and is described in the document titled: *Abbott's Instrument Communication Standard.*

The Abbott Standard Protocol is, in turn, an implementation of the Kermit File Transfer Protocol developed at the University of Columbia. If the remote system has software that supports the Kermit protocol, then it can exchange data with the AD_x analyzer.

KERMIT

Kermit was chosen for several reasons. It runs on many different systems and it is easy to implement.

Versions of Kermit have been written for just about every viable computer system made.

The details of the Kermit protocol are not described here. For information on the Kermit Protocol, consult the book: *Kermit, A File Transfer Protocol,* written by Frank daCruz and published by Digital Press, or write to Kermit Distribution, Columbia University Center for Computing Activities, 612 West 115th St., New York, NY 10025 U.S.A.

KERMIT ON THE AD_X ANALYZER

To provide maximum flexibility, the AD_x analyzer interface can act as a Kermit Server, a Kermit Client or a Basic Kermit system. The details of a Kermit Server or Client are not discussed. For more information, see the Kermit reference manuals.

The Kermit mode (Basic, Client or Server) is controlled through two menu parameters on the AD_x analyzer: SERVER and CLIENT. These are 'Y/N' (Yes/No) parameters that must be edited before using a mode.

Below are lists of Kermit packet types generated by the AD_x analyzer and recognized by the AD_x analyzer. An asterisk implies one or more of these packet types, the vertical bar implies "or". For a description of the Kermit packet types see the Kermit reference manuals.

FILE NAMES IN GENERAL

The Kermit protocol is a “File Transfer” protocol and needs to work with file names. The AD_x analyzer does not use files in the same sense as a typical computer, so it must create file names for the transmissions. The AD_x analyzer file names follow the MS-DOS format of eight alphanumeric characters, a dot and then three more alphabetic characters. AD_x analyzer file names break down into two basic types: those that are associated with result data and those associated with sample ID data.

RESULT FILE NAMES

When the results are transmitted to the remote system, the AD_x analyzer creates a file name in this format:

The file name is created in an MS-DOS style format. There are eight alphanumeric characters followed by a dot (‘ . ’) and three more alphabetic characters. (Case insensitive.)

The format of the name is: Rnnnnnxx.ADX where ‘nnnnn’ will be the serial number of the AD_x analyzer and ‘xx’ is the current carousel number (or zero).

This type of file is created by the AD_x analyzer and sent to the host.

Example: R0061405.ADX would come from AD_x analyzer number 614 running carousel number 5.

SAMPLE ID FILE NAMES

When sample IDs are transmitted from the remote computer (host) to the AD_x analyzer, the remote system must use a file name in this format:

The file name is in an MS-DOS style format. There are eight alphanumeric characters followed by a dot (‘ . ’) and three more alphabetic characters. (Case insensitive.)

The format of the name is: Snnnnnxx.ADX where ‘nnnnn’ will be the serial number of the AD_x analyzer and ‘xx’ is the current carousel number (or zero).

This type of file is created by the host and sent to the AD_x analyzer.

Example: S3276710.ADX means that these IDs are destined for AD_x analyzer number 32767 running carousel number 10.

The maximum value for a serial number is 32767. Any incoming serial numbers greater than that will get an Error (E) packet for their response. Carousel numbers are two digits in length and have a maximum value of 99.

AD_X ANALYZER / KERMIT SPECIFICS

It’s possible to tailor an implementation of Kermit to a particular instrument system and the AD_x analyzer does just that. This section briefly reviews some of the major items that are specific to the AD_x analyzer implementation of Kermit.

- The AD_x analyzer requests the maximum packet length of 94 characters.
- The AD_x analyzer requires no null padding.
- The AD_x analyzer uses CR (ASCII 13) character as its end-of-line mark.
- Control characters are quoted with the ‘#’ character.
- No eighth-bit prefixing is done, nor RLE compression.
- Only the simple, type 1, block check is supported.
- No “advanced capabilities” are supported except SERVER and CLIENT.
- Packet specifics are discussed below.

The maximum number of NAKs (Negative Acknowledgements) tolerated for one packet is editable but must be between 1 and 50. This is edited via the MAXNAKS parameter on the AD_x analyzer.

The maximum number of seconds tolerated between packets before logging a time-out is editable but must be between 1 and 300. This is changed through the TIMEOUT parameter on the AD_x analyzer.

The Mark Character on the AD_x analyzer may be changed to any control character between ASCII 1 and ASCII 31 inclusive. The AD_x SYSTEM... SERIAL PORT... SET UP... menu parameter, MARK CHAR, controls this value.

There are more minor items that are specific to Kermit on the AD_x analyzer. These are listed in the "Comments" section for each of the three modes covered in the sections below.

BASIC KERMIT

The fundamental Kermit level is the "basic" Kermit level. To transmit a result file to the host, use the AD_x analyzer Transmit procedure. To receive data from the host, use the AD_x analyzer Receive procedure. This Basic mode is selected by editing the SYSTEM... SERIAL PORT... SET UP... menu parameters, SERVER and CLIENT to 'N' (No).

Analyzer Procedure	Analyzer Generates	Analyzer Recognizes
TRANSMIT	S (F D* Z)* B | E	Y | N | E
RECEIVE	Y | N | E	S (F D* Z)* B | E

BASIC KERMIT: COMMENTS

The AD_x analyzer does not recognize data in the Y, Z or E packets. The AD_x analyzer does not put data in the E packet.

BASIC KERMIT: FILE NAMES

This is the format of the file names used by the AD_x analyzer under Basic Kermit transfers.

Rnnnnnxx.ADX

Results of a run (RUN, CAL or PANEL) transmitted to host, where 'nnnnn' is the zero-padded serial number of the AD_x analyzer and 'xx' is the carousel number (00 for no carousel number).

Snnnnnxx.ADX

Sample ID information received from host, where 'nnnn' is the zero-padded serial number of the AD_x analyzer and 'xx' is the carousel number (00 for no carousel number).

KERMIT CLIENT

When the AD_x analyzer is a Kermit Client, it acts as the "basic" Kermit with one addition: the Kermit R packet is generated by the Receive procedure. This mode is selected by editing SERVER to 'N' (No) and CLIENT to 'Y' (Yes).

Analyzer Procedure	Analyzer Generates	Analyzer Recognizes
TRANSMIT (SEND)	S (F D* Z)* B | E	Y | N | E
RECEIVE (GET)	R (Y | N | E)	S (F D* Z)* B | E

KERMIT CLIENT: COMMENTS

The AD_x analyzer does not generate the I, G or C packets.
The AD_x analyzer does not recognize the X packet.
The AD_x analyzer does not recognize data in the Y, Z or E packet.
The AD_x analyzer does not put data in the E packet.

The GET command is supported by the addition of the R packet. The SEND command is supported with the basic S (F D* Z)* B packets. The BYE and FINISH commands are not supported.

The R packet is sent at the start of the Receive procedure. The AD_x analyzer then falls into the "basic" RECEIVE loop.

KERMIT CLIENT: FILE NAMES

This is the format of the file names used by the AD_x analyzer under Kermit Client transfers.

Rnnnnnxx.ADX

Results of a run (RUN, CAL or PANEL) transmitted to host, where 'nnnnn' is the zero-padded serial number of the AD_x analyzer and 'xx' is the carousel number (00 for no carousel number).

Snnnnnxx.ADX

Sample ID information received from host where 'nnnnn' is the zero-padded serial number of the AD_x analyzer and 'xx' is the carousel number.*

* The AD_x analyzer Receive procedure asks the operator for a carousel number before creating the R packet, if the analyzer is not running. The operator will be allowed to enter a carousel number of zero (not all carousels will have numbers). How the host deals with a carousel number of zero is up to it.

KERMIT SERVER

To put the AD_x analyzer into "Kermit Server Mode," scroll to the menu option SERVER and edit it to 'Y' (Yes). When acting as a Kermit Server, the AD_x analyzer is completely passive and responds to commands from the host. The host must act as a Kermit Client.

When the AD_x analyzer is a Kermit Server, the operator can not use the Transmit or Receive procedures on the AD_x analyzer. All data transfer is controlled by the host computer. Error messages from the transmission of results are still displayed, requiring operator intervention. Data transfer may continue while waiting for the operator to respond. To avoid the need for operator intervention, the error messages can be suppressed. See the SHOW ERRORS parameter, described in Section 4, "Human Interface Changes", under "SET UP... Menu".

Analyzer as Server Recognizes	Analyzer Responds With
R	S (F D* Z)* B | E
S (F D* Z)* B | E	Y | N | E
G	Y | E

KERMIT SERVER: COMMENTS

The AD_x analyzer does not recognize the I or C packets (Sends E packet).
The AD_x analyzer does not recognize data in the Y, Z or E packets.
The AD_x analyzer does not put data in the E packet.

KERMIT SERVER: TRANSMIT

To transmit the results of a run from the AD_x analyzer to the host computer, the host must send the Kermit R packet. The R packet holds the name of the file to transfer. The format of the file name in the R packet is:

Rnnnnnxx.ADX

Results of a run transmitted to host where 'nnnnn' is the zero-padded serial number of the AD_x analyzer and 'xx' is treated as below.

'xx'	Action
00	AD_x analyzer sends results of the most recent run.
01...90	AD_x analyzer searches for results with this carousel number. First match found sends those results (*)
91	AD_x analyzer sends results stored under "slot" 1
92	AD_x analyzer sends results stored under "slot" 2
93	AD_x analyzer sends results stored under "slot" 3
94	AD_x analyzer sends results stored under "slot" 4
95	AD_x analyzer sends results stored under "slot" 5 (**)
96...99	Reserved for future interpretation by the AD_x analyzer

* The criteria for determining the "first" match found is simply that the AD_x analyzer starts with "slot" one and looks for the first matching carousel number. If all five runs have the same carousel number, you may not get the desired run.

**The "slot" matches the number under the "NUM" column printed during the AD_x analyzer Transmit procedure.

KERMIT SERVER: RECEIVE

To receive sample IDs from the host computer, the host simply starts sending the S (F D* Z)* B packet sequence. The file name in the F packet must be in the form of:

Snnnnnxx.ADX

Sample ID information received from host where 'nnnnn' is the zero-padded serial number of the AD_x analyzer and 'xx' is the carousel number (00 for no carousel number).

KERMIT SERVER: GENERIC COMMAND

The G packet lets the Client (host) retrieve special status information from the Server (AD_X analyzer). The following Generic Commands are supported:

Cmd	Description / Action taken by AD_X analyzer
F	Finish / Leaves Server mode
L	Logout / Leaves Server mode
D	Directory / Sends information about a stored result list
P	Program / Responds to commands from Client
Q	Status Query / Sends a status message to Client
VQ	Variable Query / Sends variable datum to Client

GENERIC COMMAND: FINISH AND LOGOUT

Upon the receipt of this command the AD_X analyzer sends an ACK packet and then exits its Server mode. The transmission of an ACK packet after receiving a 'G' packet may or may not follow the Kermit protocol. The Kermit manual is ambiguous on this point.

After the host issues the Finish or Logout command, the operator must edit the AD_X analyzer SERVER parameter again to restart the Server mode. This will be a bit confusing for the AD_X analyzer operator because the Finish/Logout command did not reset the AD_X analyzer menu parameter SERVER back to 'N.' The menu parameter will still (wrongly) suggest that SERVER mode is active. It will read 'Y' and the operator is required to edit it to 'Y.'

If the host issues a Finish or Logout command, it must explain this restart procedure clearly to the AD_X analyzer operator.

GENERIC COMMAND: DIRECTORY

The host uses this command to retrieve information about the results stored in a slot (see Section 4, "The Transmit Procedure"). Immediately following the 'D' is the slot number '1' to '5.'

The AD_x analyzer will respond with the information tucked in an ACK packet. After the 'Y' (in the ACK packet) comes:

n;xx;mm/dd/yy;hh:mm:ss

'n' is the "status" of the run:
'0' – no results in this slot
'1' – results, but not sent to host yet
'2' – transmission in progress
'3' – successful transmission
'4' – error during transmission

'xx' is the carousel number associated with that run, 00 for none

'mm/dd/yy' is the date of the run
'hh:mm:ss' is the time of the run

If the slot number sent from the host is invalid, the AD_x analyzer responds with an 'E' packet.

Examples:

Command from host : GD0
Example ACK from AD_x analyzer: "\ x1# E* \ r" (no slot #0)

Command from host : GD1
Example ACK from AD_x analyzer: "\ x1: Y3;06;10/16/89;11:49:48_ \ r"

Note:

' \ x1' IS THE SOH CHARACTER AND ' \ r' IS THE CR CHARACTER

GENERIC COMMAND: STATUS QUERY

The AD_x analyzer responds to the Q command with this information tucked in the ACK packet: ADx;NNNNN;VX.X where 'NNNNN' is the AD_x analyzer serial number and 'VX.X' is the software version number. Use this command to identify which AD_x analyzer is on-line.

Examples:

Command from host : GQ
Example ACK from AD_x analyzer: "\ x11 YADx;00003;V2.1Z \ r"

Note:

' \ x1' IS THE SOH CHARACTER AND ' \ r' IS THE CR CHARACTER

GENERIC COMMAND: PROGRAM

With the P command, the remote system can start and stop the AD_x analyzer. This generic packet is ACKd by the AD_x analyzer if the command is OK.

Argument	Action
P	Starts the AD_x analyzer as if the PANEL key was pressed
C	Starts the AD_x analyzer as if the CAL key was pressed
R	Starts the AD_x analyzer as if the RUN key was pressed
S	Stops the AD_x analyzer as if the STOP key was pressed

Examples:

Commands from host : GPP, GPC, GPR, GPS

Example ACK from AD_x analyzer: <the simple Kermit 'Y' packet>

If the command is accepted, the AD_x analyzer sends an ACK packet. If the command is rejected, the AD_x analyzer sends the 'E' packet.

Note:

THE REMOTE SYSTEM CAN ONLY COMMAND THE AD_X ANALYZER TO START PANEL, CAL OR RUN WHEN THE ANALYZER IS IN THE READY STATE. IT CAN USE THE STOP COMMAND AT ANY TIME. (SENDING A STOP COMMAND ON A 'STOPPED' ANALYZER IS OF NO HARM.)

GENERIC COMMAND: VARIABLE QUERY

The AD_x analyzer has several system variables with simple names. The Client can inquire the status of these variables by supplying the VQ command to the AD_x analyzer. The 'value' of the variable is returned in the ACK packet.

Name	Description	Values
C	Carousel Number	0 ... 99
S	System Status	"WARM UP," "READY," "RUNNING," "TRAINING," "ERROR PENDING"
D	AD_x Analyzer Display	<top line>;<bottom line>
T	Time Left in Run	<seconds (approximate)>

GENERIC COMMAND: VARIABLE QUERY: CAROUSEL NUMBER

If the AD_x analyzer is running and if the carousel in the instrument has a barcode, then the value is stored in this variable. Otherwise the value stored is zero.

GENERIC COMMAND: VARIABLE QUERY: SYSTEM STATUS

The status of the AD_x analyzer is reflected in this variable. "WARM UP" is the phase that the AD_x analyzer goes through when it is first powered on. "READY" means that the AD_x analyzer is ready to run. "RUNNING" means that the AD_x analyzer is performing a sample run, calibration run or panel run. "TRAINING" means that the operator is performing diagnostics procedures on the instrument. "ERROR PENDING" means that the AD_x analyzer has detected an error that could abort the run but is waiting for the operator to take the appropriate action.

Some errors only suspend the run until the operator takes the corrective action. Then the run continues normally.

Other errors will abort the run as soon as the operator presses a key. The AD_x analyzer will return to the "READY" state.

If the host computer receives an "ERROR PENDING" message, it is recommended that the host increases the frequency of the polling so it can determine the action taken by the operator to resolve the error.

GENERIC COMMAND: VARIABLE QUERY: AD_X ANALYZER DISPLAY

The ACK packet gathers data from the AD_x analyzer two-line display. Each line is 16 characters wide and padded with blanks. A semicolon separates the top line data from the bottom line data.

GENERIC COMMAND: VARIABLE QUERY: TIME LEFT IN RUN

When the AD_x analyzer is running, this variable will hold the number of seconds that remain before the run completes. Even though the resolution is seconds, the value should be considered approximate. It will be accurate for sample and panel runs, but the iteration attempts at curve fitting during a calibration run can take longer than expected and vary from run to run. When the computations take longer than expected, the number of seconds will continue to count down past zero and will be a negative number. Beginning with AD_x analyzer revision 3.2 software, however, the number of seconds will stop counting down at zero.

Examples:

Command from host : GVQC
Example ACK from AD_x analyzer: " \ x1$ Y55 \ r"

Command from host : GVQS
Example ACK from AD_x analyzer: " \ x1(YREADY6 \ r"

Command from host : GVQD
Example ACK from AD_x analyzer: " \ x1D YREADY ;SYSTEM... < \ r"

Command from host : GVQT
Example ACK from AD_x analyzer: " \ x1& Y819" \ r"

Note:

' \ x1' IS THE SOH CHARACTER AND ' \ r' IS THE CR CHARACTER

GENERIC COMMAND: VARIABLE QUERY: COMMENTS

AD_x analyzer variables cannot be 'set' to a value.

A variable name must be supplied. The AD_x analyzer will not display all active variables when sent a null name.

KERMIT SERVER: PERIODIC NAKs

While a Kermit Server is active but idle (neither sending nor receiving Kermit packets) it can send out periodic NAKs to the clients. The AD_x analyzer, when in the Server mode, does this as well.

The AD_x analyzer uses the two SYSTEM... SERIAL PORT... SET UP... menu parameters, TIMEOUT and MAXNAKS, to determine when to send an "idle NAK" packet. When the AD_x analyzer is a Server and idle, it waits TIMEOUT seconds for a packet from the host. After MAXNAK time-outs, the AD_x analyzer sends the NAK packet.

ABORTING THE TRANSFER

Regardless of the mode, Basic, Client or Server, both sides need the ability to abort an exchange of data in progress. To do so, one side will send a special Kermit packet to the other.

Both sides use the "Error (E)" packet to communicate fatal errors. If the AD_x analyzer receives an Error packet during the Transmit procedure, then the transfer is aborted and the operator is interrupted with this display (unless SHOW ERRORS is set to 'N' (No)):

```
AICS:HOST ERROR
PRESS ENTER
```

Errors during the Receive procedure are handled in a similar manner except they are printed instead of displayed.

The AICS protocol implies that the error (E) packet contains an error message. The AD_x analyzer (E) packets do not have any additional data in them. If the host sends an error message in its (E) packet, it is ignored. Either side can abort the transfer. If the AD_x analyzer wants to abort the transfer, it uses an Error packet as well.

Section 4 Human Interface Changes

This section briefly discusses what has changed, from the operator's perspective, in the AD_x analyzer software.

To accommodate the new file transfer protocol, two new procedures were added to transmit and receive files from the host and parameters were added to accommodate different Kermit settings. All the changes are within the SYSTEM... menu.

SYSTEM... MENU CHANGES

Direct Address	AD_x Analyzer Menu Item
1.0	SYSTEM...
1.1	DATE: x
1.2	TIME; x
1.3	VX.X SN: x
1.4	CONFIGURE...
1.4.1	NEED OP ID: x
1.4.2	NEED QC/IDS: x
**1.4.3	NEED CARSL ID: x
1.4.4	THRESH/CONC: x
1.4.5	DOOR LOCK: x
1.4.6	RESET VOLS: x
1.4.7	COLLATE: x
1.4.8	NUM COPIES: x
1.4.9	LANGUAGE: x
1.5	SERIAL PORT...
*1.5.1	RECEIVE
*1.5.2	TRANSMIT
*1.5.3	BAUD RATE: x
*1.5.4	AUTO SEND: x
*1.5.5	SET UP...
*1.5.5.1	TIMEOUT: x
*1.5.5.2	MAXNAKS: x
*1.5.5.3	SHOW ERRORS: x
*1.5.5.4	MARK CHAR: x
*1.5.5.5	SERVER: x
*1.5.5.6	CLIENT: x

*Menu changes with revision 2.3 software
**Menu changes new with revision 3.0 software

Human Interface Changes

CONFIGURE... MENU

The NEED CARSL ID parameter determines whether a carousel number is required for each run. If set to 'Y' (Yes), the operator is prompted for the carousel number at the beginning of the run if the carousel barcode label is missing or cannot be read. A number in the range of 1 to 26, inclusive, must be entered before the run will continue. This number is treated as if it were read from the carousel. When the parameter is set to 'N' (No), the run will continue if the carousel barcode label is missing or cannot be read. When no carousel barcode label is read, the carousel number is 0.

It is strongly recommended that NEED CARSL ID be set to 'Y' (Yes) when connecting the AD_x analyzer to a host computer.

SERIAL PORT... MENU

Most of the software modifications to implement the AICS protocol are found in the SERIAL PORT... Menu. The RECEIVE procedure tells the AD_X analyzer to wait for incoming data from the host computer. The TRANSMIT procedure sends data from the AD_X analyzer to a host computer using the AICS protocol. The AUTO SEND parameter, if set to 'Y' (Yes), transmits the results of the last run as soon as it is completed. The last item under this menu is SET UP..., a Secondary menu.

SET UP... MENU

The TIMEOUT parameter sets the number of seconds that the AD_x analyzer will wait for a packet from the host computer to arrive. If time expires before a packet arrives, then that is considered an error ("NAK"). This is primarily used for the initial Send-Init / ACK connection. This number can be edited to a value in the range of 1 to 300, inclusive.

The MAXNAKS parameter holds the value for the number of consecutive NAKs (errors) tolerated. Once this limit is exceeded, the transfer is aborted. This parameter can be edited from 1 to 50, inclusive.

SHOW ERRORS controls whether or not error messages interrupt the AD_x operator when transmitting results (even when in Kermit Server mode). If set to 'Y' (Yes), then the operator will be interrupted with error messages when they occur. If set to 'N' (No), the error messages are not displayed on the AD_x analyzer.

MARK CHAR allows the operator to redefine the character that starts each protocol packet. It is normally set to '1' indicating a Control-A (SOH) character, as defined by the protocol. It is editable, with a range of 1 to 31, inclusive.

SERVER, if edited to 'Y' (Yes), starts the AD_x analyzer acting as a Kermit Server. Refer to the five "Kermit Server" topics in "The Abbott Standard Protocol", Section 3, for more information. CLIENT, if edited to 'Y' (Yes), allows the AD_x analyzer to act as a Kermit Client when receiving sample IDs from the host. To act as a Kermit Client the AD_x analyzer simply sends the Kermit (R) packet before waiting for the (S) packet from the remote.

Human Interface Changes

TRANSMISSION OF RESULTS TO HOST

Results are transmitted to the host system using the "Basic" Kermit protocol technique. Kermit is not described here—see the Kermit reference manuals for more information.

Results may be transmitted manually (via the AD_x analyzer Transmit procedure) or the transmission can occur automatically at the end of the run. The order of Kermit packets during transmission are:

AD_x Analyzer Sends		Host Sends	
(S)	Send-Init	(Y)	ACK
(F)	File Header	(Y)	ACK
(D)	Data	(Y)	ACK
(Z)	End of File	(Y)	ACK
(B)	End of Session	(Y)	ACK

After each packet is transmitted by the AD_x analyzer, Kermit requires the AD_x analyzer to stop and wait for the acknowledgement from the host. If an acknowledgment (ACK) does not arrive or the remote sends a negative acknowledgement (NAK) packet, the AD_x analyzer resends the packet. If the host never sends an ACK or keeps sending NAKs, then the AD_x analyzer eventually gives up trying and stops the transmission. This is determined by two parameters on the AD_x analyzer: TIMEOUT and MAXNAKS. The former is the number of seconds that the AD_x analyzer will wait for a response from the host. If nothing comes from the host, the AD_x analyzer counts it as one error. MAXNAKS is set to the number of consecutive errors tolerated before the transmission is aborted. Typical values are 10 seconds for TIMEOUT and 10 MAXNAKS.

RESULT TRANSMISSION ERRORS

If the transmission is aborted because of an error, the AD_x analyzer can be programmed to interrupt the operator with an error on the display. The interruption can be suppressed as well. See the SHOW ERRORS parameter under “SET UP... Menu”, earlier in this section.

The displayed errors will use the current AD_x analyzer error scheme where the error message appears on the top line, the corrective action appears on the bottom line and the system beeps until the correct key is pressed.

For AICS errors, the bottom line will always be “PRESS ENTER”. The operator must press the ENTER key to regain normal instrument operation. The transmission error should be diagnosed and corrected before the session is restarted.

The error message on the top line will vary according to where in the transmission the error happened. The top lines all start with the “AICS:” prefix. This way the operator can quickly distinguish these errors from other AD_x analyzer errors.

For example:

Error occurs on packet	Top Line is:
(S)	AICS:NO HOST
(F)	AICS:NAK ON F
(D)	AICS:NAK ON D
(Z)	AICS:NAK ON Z
(B)	AICS:NAK ON B

Human Interface Changes

THE TRANSMIT PROCEDURE

This procedure sends the results of a run to the host computer using the AICS protocol. This is the “manual” way to send the results. To use it:

The operator scrolls to TRANSMIT and presses ENTER. The AD_X analyzer prints:

pasteup illustration

The AD_X display will prompt the operator for the run number.

The operator enters the number (NUM) of the run to be transmitted and presses ENTER. The AD_X analyzer then starts to transmit the results of that run.

Once the transmission is started, it continues in the background. When the Transmit procedure finishes, the display reads READY and the printer prints “DONE: xx:xx:xx”.

To cancel a transmission in progress, the operator must restart the Transmit procedure and then enter the number of the run that’s being transmitted.

This procedure can be invoked while the instrument is running, ready or warming up.

To fit all of the information into one line, the date and time of the stored run is abbreviated.

The various status messages include:

NOT TRANSMITTED	the results have not been transmitted yet
TRANSMISSION COMPLETE	transmission was successful
ABORTED TRANSMISSION	an error occurred during the transmission
TRANSMITTING NOW	the results are being transmitted

AUTOMATICALLY SENDING RESULTS

The operator can edit the menu parameter AUTO SEND to 'Y' (Yes) and activate the automated transmission option. If activated, the transmit code starts immediately after the carousel is removed from the instrument. The results transmitted are those from the run that just finished.

RECEPTION OF SAMPLE IDs FROM HOST

The AD_X analyzer can accept sample ID (SID) information from the host computer. The Receive procedure performs the task.

When receiving data from the host, the general flow of Kermit packets is the same as described in "Transmission of Results To Host", in this section, but the roles of sender and receiver are reversed. For example, the AD_X analyzer expects to receive the (F) packet and then it sends the (Y) packet.

THE RECEIVE PROCEDURE

Beginning with AD_X analyzer revision V3.0 software, this procedure can only be invoked while the instrument is running. Previous software revisions allowed it to be invoked while the instrument is ready or warming. This is not recommended.

Briefly, the operator scrolls to the procedure and presses ENTER. Since sample IDs are tied to carousels and carousels are identified by numbers, the Receive procedure will prompt the operator for a carousel number if the current carousel does not have a barcode label, or if the analyzer is not running.

After the carousel number is entered, the display changes to reflect the status of the Kermit reception process.

Display	Description
RECEIVE WAITING FOR HOST	AD_X analyzer is waiting for (S) packet
RECEIVE GETTING A FILENAME	AD_X analyzer is decoding the (F) packet
RECEIVE GETTING DATA	AD_X analyzer is decoding a (D) packet
RECEIVE RECEIVE COMPLETE	AD_X analyzer is decoding a (Z) or (B) packet

Human Interface Changes

Once the data has been received, the data is parsed, presented to the AD_X analyzer and loadlist information is printed:

pasteup illustration

"Current" data is data that applies to the carousel in progress. (The AD_X analyzer QC/ID INFO... menu always displays "current" SID data.) The QC/ID data is transferred to the QC/ID INFO... menu where it will apply to the run in progress.

Several things could go wrong and cause the Receive procedure to reject the received IDs. The data must parse correctly (see 'Receive Data – Errors') and the data must match the current carousel. More details can be found in Section 6 of this document.

Section 5 Catching Result Data

This section covers, in detail, the format of the data sent from the AD_x analyzer to the host computer. The data contains the test results from a previous run.

Unlike other Abbott instruments, the format of the data is not the same as the printout. Rather, the format is based on the idea of “records”. These records are sent to the host system one record at a time. The host should collect the records, then, when the transmission is complete, parse them to extract the relevant information.

The records will be described in more detail; however, be aware that the size of a record is larger than that of a Kermit packet. This means that during a transmission, the AD_x analyzer will:

- format the test results of a run into records
- break the records into Kermit packets
- transmit the packets to the host

In turn, the host system should:

- accept the Kermit packets from the AD_x analyzer
- reassemble the records from the packets
- parse the test results from the records

Catching Result Data

DATA WITHIN A RECORD

When data is inserted into the records:

The format and precision of concentrations matches exactly what is printed on the paper tape.

Other intermediate result values, such as blank values or raw mP (millipolarization) values are not as important and the format and precision may be greater than what is printed on the tape.

NUMERIC DATA

May have leading zeros and/or leading white-space.
May have a leading sign (‘+’ or ‘–’).
May be in scientific notation (3.140E-02).

ALPHABETIC DATA

Is case sensitive unless otherwise noted.
May be preceded by white-space.

A limited set of special characters is transmitted as well. These include:

Date information has slashes (05/02/89).
Time information has colons (11:03:45).

If the information in a field is “not applicable”, a special character is inserted into the field. Refer to “Missing Fields” in this section for additional information.

The “units” string that is embedded in reagent cartridge records usually has a slash in it (MG/DL, MOLES/LITER).

Fields within a record are separated from one another by a special character. Refer to “Fields Within Records” in this section for details.

TEST RESULTS INTO RECORDS

At the end of a run on the AD_x analyzer, the software creates a list of records relevant to that run. The result data are inserted into these records. In general, one record is created for each position on the AD_x analyzer carousel. All of the result data specific to that carousel location are inserted into that record. At the end of the run, there is also some information created that isn't related to a carousel location. Additional records are created for this special information.

After all of the records have been created for a run, they are linked together in a list that describes the entire run. The AD_x analyzer has enough memory to store the results for five runs at once. The results from the sixth run overwrite the first; the results of the seventh overwrite the second and so on. The AD_x analyzer stores the records in volatile RAM storage, so they are lost if the power fails.

Again, the results are stored in volatile memory on the AD_x analyzer. The host system should be set up to retrieve the results from the AD_x analyzer as soon as they are ready!

Catching Result Data

RECORD IDs

Records are distinguished from one another by their “type”. Most records are associated with specific carousel positions and the type of the record indicates information relevant to that carousel position. Around the carousel, each position, or “location”, could hold one of the following:

- a reagent cartridge
- a sample
- a control
- a calibrator
- nothing (empty)

There are three other record types and these are used to hold information that is not specific to a position on the carousel. The first is the “carousel record” type which contains information about the entire carousel used in the run. The second type is the “calibration curve information” record which is used during calibration runs to hold supplemental data about the curve. The third record is the “AICS Header” record type which is described in a later section.

The type of each record is signified by its “Record ID” field. This is the ID that is inserted into the record and used by the host system to determine the contents of the record. The Abbott Communication Standard dictates a general format for record IDs: 8 bytes in length, followed by a one character delimiter, followed by the record specific data. There will always be an 8 character record ID, but the total length of the record will vary from one ID to another.

The AD_x analyzer specification imposes more structure on record IDs. The first three characters of every AD_x analyzer record ID are alphabetic and provide a mnemonic nature for the ID. The next four characters are numeric. The remaining character is an ASCII space character. A delimiter character follows the space and serves to mark the start of the data in the record.

The Record IDs are (excluding the quotation marks and the delimiter):

Record ID	Description
“CSL0100 ”	Record has carousel data
“CAL0200 ”	Record has calibrator data
“SAM0300 ”	Record has sample data
“CTL0400 ”	Record has control data
“RGT0500 ”	Record has reagent cartridge data
“CCI0600 ”	Record has additional calibration data
“EMP0700 ”	Record has empty position data
“SID1000 ”	Record has sample ID data
“00000000”	AICS Header Record (*)

*The last record ID, AICS Header Record, is discussed later in this section. Note that the AICS Header Record is not an AD_X analyzer record ID, and therefore, does not fit into the above description.

As stated before, the first three characters of an AD_X analyzer record ID provide a mnemonic nature for the ID. The numeric portion is used for “version control”, a method to communicate record changes to the host computer. The four digits are broken into two groups; one is the major change code and the other, the minor change code. For example:

CTL0400	
CTL	mnemonic portion
04	major change code
00	minor change code

The intent of the mnemonic portion of the code is to aid the host software engineers in debugging the software. In the event that the AD_X software must make a major change to the data in a record, it changes the mnemonic portion and the first two digits of the group code of the record ID. If a minor change is made to the record content, only the last two digits of the ID change.

Catching Result Data

SPECIFIC RECORD CONTENTS

Each record type will be reviewed, describing the data in that record. The data in the field will be characterized by its name, maximum size and data content.

Each field in the record is a variable length field. It may be smaller than the maximum size listed. Fields within a record are separated from one another by a delimiter character. The size number does not account for the delimiter character. More information on the delimiter character is provided under "Fields Within Records" in this section.

CAROUSEL RECORDS

This record is used to hold information about the entire carousel used in the run.

Record ID: CSL0100

The record holds this information in this order:

Field Name	Size	Content
Instrument	3	Contains the string “ADX”.
Serial Number	5	The serial number of this AD_x analyzer. Numeric in the range of 0 to 32767.
SW Version	4	The software revision level of the analyzer. In the format “Vx.x” (as in “V2.3” or “V3.0”)
Launch Cmd	5	The key used to start the run. Alphabetic, either: “RUN,” “CAL” or “PANEL”.
Start Digits	5	Numeric keys pressed before Launch Cmd. Typically indicates the operator is calibrating more than one assay per carousel.
Start Date	8	Date run was started. Numeric with slashes. Format is “mm/dd/yy” as in “05/02/89”.
Start Time	8	Time run was started. Numeric with colons. Format is “hh:mm:ss” and in 24 hour time. As in: “09:10:32”, “14:22:59”.
Operator ID	10	Operator identifier. Alphanumeric. Between 0 and 10 characters in length.
Carousel ID	5	Carousel identifier. Numeric: 0 to 99
Thresh Only	1	Value from AD_x analyzer menu item THRESH/CONC. Either “Y” or “N”. (*)
Nag Code	5	Value that indicates what warning messages appeared on the AD_x printout at the start of the run. Numeric in the range of 0 to 65535. See Section 5, “The Nag Code”, for interpretation.
Error String	40	A printable ASCII error message that occurred during the run. (**) One of the following: “ERROR: CAROUSEL ID MISMATCH” “ERROR: OPERATOR ID MISMATCH” “ERROR: QC/ID DATA MISMATCH” NOT APPLICABLE string

* The string “Y” means that the operator wants only qualitative results. Qualitative results will have the strings “ABOVE T” and “BELOW T” in the result field instead of a numerical value for the concentration. An “N” string means that the results field of a sample record should have a numeric value.

**If this field contains an error message, then the most likely cause is the host sending incorrect sample IDs to the AD_x analyzer. Refer to Section 6, “Receive Sample ID Data – Errors”, for more information. The field will contain the NOT APPLICABLE string if no errors occurred. The NOT APPLICABLE string is described in Section 5, “Missing Fields”.

Catching Result Data

CALIBRATOR RECORDS

This record is used to hold information about each calibrator used in the calibration.

Record ID: CAL0200

The record holds:

Field Name	Size	Content
Location	2	Position of this calibrator on the carousel. Numeric in the range of 1 to 20.
Reagent Location	2	Carousel location for the reagent cartridge associated with this calibrator. Numeric in the range of 0 to 19.
Error String	40	Contains printable ASCII information if an error occurred on this calibrator. If no error occurred, then this field contains the NOT APPLICABLE string described in Section 5, "Missing Fields."
Calib Level	1	Alphabetic character ("A", "B", "C", "D", "E" or "F") that indicates the calibrator concentration level.
Calib Conc	10	Numeric concentration value for this calibrator.
Blank Value	10	Numeric blank reading for this calibrator.
Net I Value	10	Numeric Net Intensity value (or Initial I value) for this calibrator. (*)
mP Value	10	Numeric millipolarization (or Percent) value for this calibrator. (*)

*Assays that use the FPIA technology report results with "Net Intensity" and "millipolarization (mP)" values. Assays that use the REA® technology report results with "Initial I" and "Percent" values. Also note that this record does NOT have the "Is Diluted" field.

SAMPLE RECORDS

This record is used to hold information about each sample on the carousel.

Record ID: SAM0300

The record holds:

Field Name	Size	Content
Location	2	Position of this sample on the carousel. Numeric in the range of 1 to 20.
Reagent Location	2	Carousel location for the reagent cartridge associated with this sample. Numeric in the range of 0 to 19.
Error String	40	Contains printable ASCII information if an error occurred on this sample. If no error occurred, this field contains the NOT APPLICABLE string described in Section 5, “Missing Fields”.
Sample ID	10	Alphanumeric ID for the sample. If no sample ID is present, this field contains the NOT APPLICABLE string described in Section 5, “Missing Fields”.
Modifier	18	Printable ASCII string that contains supplementary information for the result field. The contents will be one of the following: “HI” (above the therapeutic range) “LO” (below the therapeutic range) “>=T” (above the threshold) NOT APPLICABLE string (*)
High Blank	1	Either “Y” or “N”. If the contents are “Y”, then the blank reading on this sample was above the assay’s maximum allowable value.
Blank Value	10	Numeric blank reading for this sample.
NetP Value	10	Numeric Net Polarization (or Initial I) value for this sample. “Net P” for FPIA assays and “Initial I” for REA assays. (**)

* A NOT APPLICABLE modifier can be interpreted that the sample result was within the therapeutic range or below the threshold. Always check the sample’s Result field for validity before reporting the modifier field.

** Assays that use the FPIA technology report results with “Net Intensity” and “millipolarization (mP)” values. Assays that use the REA® technology report results with “Initial I” and “Percent” values.

Catching Result Data

Field Name	Size	Content
Result	10	Usually numeric concentration value for this sample. (***) Non–numeric possibilities include: “ABOVE T” (qualitative above threshold) “BELOW T” (qualitative below threshold) “HIGH” (off-scale; above F calibrator) “LOW” (off-scale; below A calibrator)
Is Diluted	1	Either “Y” or “N”. “Y” means that the sample was run with the assay’s sample volume parameter different from the calibrator volume parameter.

***The AD_x analyzer can be programmed through the keypad to report either quantitative or semiquantitative sample results. This ability is controlled through the menu item THRESH/CONC. If THRESH/CONC is set to ‘Y’ (Yes), then results are reported numerically and the modifier indicates whether the sample is above the threshold, outside the therapeutic range or neither is applicable. If THRESH/CONC is set to ‘N’ (No), then the results for Abused Drug Assays are only reported as “ABOVE T” or “BELOW T”.

The “HIGH” and “LOW” strings can appear regardless of the THRESH/CONC setting. These two strings indicate that the result is outside of the range of the calibration curve. Typically this warrants that the sample be repeated before a result is reported.

CONTROL RECORDS

This record is used to hold information about each sample on the carousel.

Record ID: CTL0400

The record holds:

Field Name	Size	Content
Location	2	Position of this control on the carousel. Numeric in the range of 1 to 20.
Reagent Location	2	Carousel location for the reagent cartridge associated with this control. Numeric in the range of 0 to 19.
Error String	40	Contains printable ASCII information if an error occurred on this control. If no error occurred, this field contains the NOT APPLICABLE string described in Section 5, "Missing Fields".
Control Level	3	One of the following alpha strings: "QCL" for a low control "QCM" for a medium control "QCH" for a high control
Modifier	18	Printable ASCII string that contains supplementary information for the result field. The contents will either be: "IN" (control is within specifications) "OUT" (control is outside specifications)
High Blank	1	Either "Y" or "N". If the contents are "Y", the blank reading on this control was above the assay's maximum allowable value.
Blank Value	10	Numeric blank reading for this control.
NetP Value	10	Numeric Net Polarization (or Initial I) value for this control. "Net P" for FPIA assays and "Initial I" for REA® assays. (*)
Result	10	Numeric concentration value for this control or one of the following alpha results: "HIGH" (off-scale; above F calibrator) "LOW" (off-scale; below A calibrator)
Is Diluted	1	Either "Y" or "N". "Y" means that the control was run with the assay's sample volume parameter different from the calibrator volume parameter.

*Assays that use the FPIA technology report results with "Net Intensity" and "millipolarization (mP)" values. Assays that use the REA® technology report results with "Initial I" and "Percent" values.

Catching Result Data

REAGENT CARTRIDGE RECORDS

This record is used to hold information about each reagent cartridge.

Record ID: RGT0500

The record holds:

Field Name	Size	Content
Location	2	Position of the reagent cartridge on the carousel. Numeric in the range of 0 to 19.
Name	29	Name of the assay (i.e., “ACETAMINOPHEN”).
Assay Number	3	The unique ID number for the assay. Numeric ranging from 1 to 999. See the AD_X Operator’s Guide for the correlation between assay numbers and their names.
Assay Type	2	“Type” code for the assay. Numeric from 1 to 99. Refer to “Assay Type Codes” in this section for details.
Sample Reps	2	Sample replicate size selected for the reagent cartridge. Numeric from 1 to 99.
Calib Reps	2	Calibrator replicate size selected for the reagent cartridge. Numeric from 1 to 99.
QC Date	8	The date of the last run that all identified controls passed. Example: “05/02/89”. Note that not all levels of controls must be run, but all samples identified as controls must pass. If the date is “00/00/00” and the time is “00:00:00”, controls have not been run or have not all passed for this assay.
QC Time	8	The time of the last run that all identified controls passed. Example: “14:53:00”. Note that not all levels of controls must be run, but all samples identified as controls must pass. If the date is 00/00/00” and the time is “00:00:00”, controls have not been run or have not all passed for this assay.
Calib Date	8	The date that the last calibration passed. Example: “12/02/90”. If the date is “00/00/00” and the time is “00:00:00”, this assay has not been calibrated.
Calib Time	8	The time that the last calibration passed. Example: “22:03:01”. If the date is “00/00/00” and the time is “00:00:00”, this assay has not been calibrated.
Low Limit	10	The therapeutic low limit for the assay. Numeric. Set to zero for the Abused Drug Assays because these assays do not have a lower limit.
High Limit	10	The therapeutic high limit for the assay. Numeric. For Abused Drug Assays, this is the threshold value.

Field Name	Size	Content
Units	10	The selected units for the assay. Printable ASCII. Example “NG/ML”.
Dilution Factor	10	If Dilution Protocol is run on this assay, then this is the dilution factor. The value should be set to 1.0 if Dilution Protocol is not performed for the assay. Refer to Section 5, “Dilution Protocol”, for more information.
Cartridge Barcode	10	The 10 digit barcode number that appears in the upper right-hand corner of the reagent cartridge label. Example: “1234567890”.
Tests Left	6	The amount of tests remaining in the reagent cartridge. Numeric, for example “45”.

Catching Result Data

CALIBRATION CURVE INFO RECORDS

This record is used to hold information about each calibration curve in a calibration run. This is required because a calibration run creates new information that doesn't mesh nicely with the other structures. One of these records will accompany each calibration on the carousel (up to 3 with the current carousel). Presently, every AD_x assay uses six calibrators.

Record ID: CCI0600

This record holds:

Field Name	Size	Content
Reagent Location	2	Position of the reagent cartridge on the carousel that goes with the calibration curve. Numeric in the range of 0 to 19.
Calib Status	4	Either "PASS" or "FAIL" depending on the success of the calibration.
Average A Value	10	The average millipolarization (or Percent) value for the A calibrator. If no replicates were run, then this is the same as the actual value. Numeric. (*)
Fitted A Value	10	The predicted value for the A calibrator from the calibration curve. Numeric.
Average B Value	10	Average value for B calibrator.
Fitted B Value	10	Predicted value for B calibrator.
Average C Value	10	Average value for C calibrator.
Fitted C Value	10	Predicted value for C calibrator.
Average D Value	10	Average value for D calibrator.
Fitted D Value	10	Predicted value for D calibrator.
Average E Value	10	Average value for E calibrator.
Fitted E Value	10	Predicted value for E calibrator.
Average F Value	10	Average value for F calibrator.
Fitted F Value	10	Predicted value for F calibrator.

*Assays that use the FPIA technology report results with "Net Intensity" and "millipolarization (mP)" values. Assays that use the REA technology report results with "Initial I" and "Percent" values.

Field Name	Size	Content
Parameter A	10	First of four values that describe the equation of the calibration curve. Numeric (exponential). Example: “1.033E-01”.
Parameter B	10	Second of four values that describe the equation of the calibration curve. Numeric (exponential). Example: “1.033E-01”.
Parameter C	10	Third of four values that describe the equation of the calibration curve. Numeric (exponential). Example: “1.033E-01”.
Parameter D	10	Fourth of four values that describe the equation of the calibration curve. Numeric (exponential). Example: “1.033E-01”.
RMSE	10	The root mean squared error statistic for the calibration curve. Numeric.
Error String	40	Contains printable ASCII information if an error occurred on this curve. If no error occurred, then this field contains the NOT APPLICABLE string as described in Section 5, “Missing Fields”.

To calculate the “residuals”, subtract the “fitted” value from the “average” value for any calibrator A to F. This value will be the same as the value in the ERR column on the AD_x analyzer printout.

Catching Result Data

EMPTY LOCATION RECORDS

This record is used to hold information about each empty location on the carousel. An empty location has no cuvette and no reagent cartridge.

Record ID: EMP0700

This record holds:

Field Name	Size	Content
Location	2	Position of this empty location on the carousel. Numeric in the range of 1 to 20.

FIELDS WITHIN RECORDS

The AD_X analyzer uses variable length, delimiter bound fields within a record. (Only the AICS Header record is a fixed length field record.) Consequently, subsequent revisions of the AD_X analyzer software should not change the relative order of the fields in a record.

The current delimiter character is the semicolon ‘ ; ’ character. This one was chosen because it is not used in the AD_X analyzer printout and cannot be confused as anything else but a delimiter.

MISSING FIELDS

If a field is blank or not applicable, a special character is inserted to prevent the two delimiters from bumping one another.

Examples of a not applicable field are the Error String field in a cuvette with no errors or a missing sample ID on a sample cuvette. The character chosen is the question mark ‘?’ character. Like the delimiter character, this one does not appear on a normal printout.

The host system must be able to handle *any* field having the not applicable character. For normal AD_X analyzer operation, almost all fields in the records are applicable. When errors occur during a run, any field may be set to not applicable.

PARSING INDIVIDUAL RECORDS

From the host's perspective, each record begins with the record ID and ends with an END MARKER. Record IDs are described in this section.

The END MARKER is used to mark the end of a record and to separate it from the next record. The END MARKER is CRLF, the carriage return character (ASCII 13), followed by the line feed character (ASCII 10).

For example, a complete carousel record includes:

Date Transmitted	Description
CSL0100 ;	Record ID
ADX;	Instrument
614;	Serial number
V2.3;	Software revision (V2.3 or higher)
PANEL;	Launch command
0;	Start digits
05/02/89;	Start date
15:37:45;	Start time
061457;	Operator ID
2;	Carousel ID
N;	Thresholds Only
31231;	Nag Code
?;	Error String
<carriage-return>	ASCII 13
<line-feed>	ASCII 10
RGT0500 ;	start of next record
...	etc.

Note that any leading white-space was removed and that each field is on a separate line for readability only.

Catching Result Data

ERROR MESSAGES

Any normal AD_x analyzer error message that can appear on the AD_x analyzer printout appears in the Error String field in a record. Consult the AD_x Operator's Guide for a complete list of the printed and displayed error messages.

The exceptions are the printed error messages that appear as:

* CAL ERROR * <specific error>

on the AD_x analyzer printout. These messages, when moved into the transmitted data stream, have the prefix "* CAL ERROR *" removed. The host receives only the specific error message, as in "A CAL READING TOO LOW" or "RMSE TOO LARGE".

Important:

IF A RECORD HAS AN ERROR MESSAGE IN THE ERROR STRING FIELD, DO NOT ATTEMPT TO INTERPRET ANY RESULT DATA IN THAT RECORD. ERROR MESSAGES CANNOT BE IGNORED. THEREFORE, THE HOST SOFTWARE MUST ALWAYS CHECK THE ERROR STRING FIELD BEFORE PARSING ANY RESULT FIELD.

RESULT MODIFIERS

Characters that "modify" the interpretation of the results printed on the tape are given their own field. Example modifiers include:

Modifier	Interpretation
HI	Specimen result is above the therapeutic limit HI LIMIT(for Toxicology assays)
LO	Secimen result is below the therapeutic limit LO LIMIT (for Toxicology assays)
>=T	Abused Drug Assay result is greater than or equal to the threshold THRSHLD
IN	Quality Control is within specifications
OUT	Quality Control is outside specifications

Note:

ABUSED DRUG ASSAYS ARE HANDLED DIFFERENTLY THAN THE OTHER ASSAYS IN THAT THERE ARE NO THERAPEUTIC LIMITS. FOR THESE ASSAYS, THE RESULTS ARE EITHER "GREATER THAN OR EQUAL TO THE THRESHOLD" OR "LESS THAN THE THRESHOLD" VALUE.

CODED INFORMATION

Some records contain members that hold encoded information. The data must be decoded to be interpreted. For example, the Carousel Record member Nag Code compresses the nag messages into a coded value.

THE NAG CODE

The Nag Code field is located in the Carousel Record. This field holds a numeric value that is copied directly from an internal data structure. It must be decoded to be of any value.

Each bit in the value controls the appearance of a Nag Message. Once decoded, it can tell you what AD_x analyzer Nag Messages appeared on the analyzer printout at the start of a run. Example AD_x analyzer Nag Messages include:

"* WARNING: ADx NOT PHOTO CALIBRATED"
"* WARNING: ADx FAILED TEMP CHECK"
"* TIME FOR: PIPETTE CHECK"

The value in the Nag Code field is an integer that runs from 0 to 65535 inclusive. Each bit in the integer corresponds to a Nag Message. If the bit is SET (1), the Nag Message does NOT appear. If the bit is RESET (0), the Nag Message prints on the analyzer printout.

Catching Result Data

Three of the bits are not used and should be ignored. Bits 9, 10 and 15 (least significant bit is bit 0) should be ignored regardless of their value. The bit meanings are:

Bit	Reset Interpretation (i.e., bit is 0)
0	ADx NOT PHOTO CALIBRATED
1	ADx FAILED PIPETTE CHECK
2	ADx NOT BOOM CALIBRATED
3	ADx NOT TEMP CALIBRATED
4	ADx FAILED PHOTO CHECK
5	ADx FAILED TEMP CHECK
6	ADx NOT CAROUSEL CALIBRATED
7	WARNING: EXTERNAL THERMISTOR FAILURE
8	WARNING: ADx WARMUP OVERRIDDEN
9	<not used>
10	<not used>
11	TIME FOR: PIPETTE CHECK
12	TIME FOR: TEMP CHECK
13	TIME FOR: PHOTO CHECK
14	WARNING: ADx SERIAL NUMBER NOT SET
15	<not used>

ASSAY TYPE CODES

The AD_x analyzer is capable of running assays of different types on one carousel. Abbott groups their assays according to these type codes:

Group	Assay Type
AMINOGLYCOSIDE GROUP CODE	1
ANTIBIOTIC GROUP CODE	1
ANTIARRHYTHMIC GROUP CODE	2
ANTIASTHMATIC GROUP CODE	3
ANTICONVULSANT GROUP CODE	4
ANTINEOPLASTIC GROUP CODE	5
CARDIAC GLYCOSIDE GROUP CODE	6
TOXICOLOGY GROUP CODE	7
ABUSED DRUG ASSAYS GROUP CODE	8
ENDOCRINE GROUP CODE	9
THYROID GROUP CODE	10
CLINCHEM GROUP CODE	11
SERUM PROTEINS GROUP CODE	12

Note that Aminoglycosides and Antibiotics share the same code.

Catching Result Data

THE AICS HEADER RECORD

The Abbott Standard Protocol mandates that there be a special record sent at the beginning of every file transfer. This special record is known as the HEADER record and it is identified with the reserved record ID of "00000000".

This Header Record is to be common among all Abbott instruments that support the Standard. It is a FIXED FIELD record that contains the data for an AD_x analyzer:

Start Position	Field Type	Description	Actual Field Content
1	N	Record ID	"00000000"
9	A	Delimiter	" ; "
10	AN	Instrument ID	"ADX "
15	AN	Serial Number	<depends>
25	AN	Software Revision	"V3.0 " (or higher)
35	AN	Instrument Specific	<all ASCII spaces>
45		End of Record Marker	CRLF

The Header Record is always the first record (not packet) transmitted to the host. It is followed by the Carousel Record. Thus, an example transmission for a V2.3 AD_x instrument, serial number 614, becomes:

Date Transmitted	Description
00000000	Header Record ID
;	Delimiter
ADX^^	"^" means a space character (ASCII 32)
614^^^^^^^	"^" means a space character
V2.3^^^^^^	"^" means a space character
^^^^...^^	10 more spaces
<carriage-return>	ASCII 13
<line-feed>	ASCII 10 End of Record
CSL0100 ;	Record ID
ADX;	Instrument
614;	Serial Number
V2.3;	Software revision
PANEL;	Launch command
0;	Start digits
05/02/89;	Start date
15:37:45;	Start time
061457;	Operator ID
2;	Carousel ID
N;	Thresholds only
31231;	Nag code
?;	Error string
<carriage-return>	ASCII 13
<line-feed>	ASCII 10 End of Record
RGT0500 ;	start of next record

Catching Result Data

DILUTION PROTOCOL

The AD_X analyzer is capable of running samples with dilution protocol. This means that samples are run with a sample volume that is different from the volume used to calibrate the assay. The Dilution Protocol procedure is not currently used in Abused Drug Assays but is often used in therapeutics or toxicology samples with extraordinarily high concentrations.

Dilution Protocol is used on the AD_X analyzer by editing the SAMPLE VOL parameter for the assay. The new ratio of SAMPLE VOL to CAL VOL is the "dilution factor". The AD_X analyzer uses the dilution factor when calculating the results. The result is calculated by dividing the diluted result by the dilution factor.

The results that appear on the analyzer printout (and transmitted in the data stream) have had the dilution factor applied and, therefore, NO additional calculations are required.

The dilution factor is sent in every reagent cartridge record transmitted. A factor of "1.0" indicates that Dilution Protocol was not performed for that one assay. Given the flexibility of the AD_X analyzer, Dilution Protocol can be run on one or more assays on the carousel without affecting the others.

Each sample record has an "Is Diluted" field that is set to 'Y' if Dilution Protocol was used; otherwise, it is set to 'N'. If the result for that sample is not reportable, perhaps because of an error, the "Is Diluted" field is set to 'N'. In other words, if a sample is run with dilution protocol, but the calculated result is "LOW" or "HIGH", then an 'N' appears in the "Is Diluted" field for that sample.

The only data that can come from the host to the AD_x analyzer is sample ID (SID) data. To receive SID data from a host computer, the operator uses the Receive procedure on the AD_x analyzer. The Receive procedure starts the Kermit Receive routine. Once the reception is complete, the procedure scans the data then prints the information on the analyzer printout. The received IDs are then applied to the run in progress or the next run.

Note that the Receive procedure parses the SID data after the reception has finished. This complicates the handling of errors detected during the parsing phase.

RECEIVE SAMPLE ID DATA – ERRORS

Before discussing the received data, it is important to understand what the AD_x analyzer will do if it finds an error in the received Sample IDs (SIDs) during the parsing phase. These are not the errors that might occur during the transfer of Kermit packets from the host to the AD_x analyzer, rather what happens when the AD_x analyzer detects an error in the SID data *after* the file exchange is finished.

There are three errors that the AD_x analyzer might detect while parsing the received data. One, the received Carousel ID may not match the carousel in the AD_x analyzer. Two, the received Operator ID may not match the ID entered at the start of the run. Three, the received Sample IDs may not match the cuvettes on the running carousel. Note that these three errors can only happen if the AD_x analyzer is running when the data is received.

If one of these errors is detected, then the proper error message, either “ERROR: CAROUSEL ID MISMATCH”, “ERROR: OPERATOR ID MISMATCH”, or “ERROR: QC/ID DATA MISMATCH” (respectively) will be inserted into the Error String field of the Carousel Record created with the other result records at the end of the run.

The host, therefore, should check the Error String field in the Carousel Record when it receives the results from the AD_x analyzer. An error message in the field suggests that the AD_x analyzer detected a problem with the Sample IDs sent from the host.

The AD_x analyzer has to be running when the SID data arrives to create one of these error conditions. Beginning with software revision V3.0, this will be the case when the AD_x analyzer is not acting as a Kermit Server (parameter SERVER is ‘N’). The Receive procedure can only be used when the analyzer is running and has completed its carousel verification checks (loadlist has printed). However, if the AD_x analyzer is acting as a Kermit Server (parameter SERVER is ‘Y’), the host may transmit SID data before a run and bypass these checks. This is NOT recommended.

Sending Sample IDs

RECEIVE SAMPLE ID DATA – DETAILS

The AD_x analyzer Receive code expects the data to be transmitted in a specific format. The SID data from the host should contain the following information:

Description	Example
Record ID delimiter	Example “SID1000 ” ‘ ; ’
Operator ID delimiter	“061457” ‘ ; ’
Carousel ID delimiter	“10” ‘ ; ’
Contents ID for a location delimiter ... Contents ID for a location delimiter	see below ‘ ; ’ ... see below ‘ ; ’
End of Record Marker	CRLF (ASCII 13,10)

Note that the SIDs for the location need not be in any order. But the first field must be the operator ID, and the second is the carousel ID.

OPERATOR IDs

The operator ID string is numeric, less than or equal to 10 characters in length, separated from other fields with delimiters and set to '?' if missing. Some examples, with the delimiter already appended, are:

Operator ID	Transmitted Data
1234567890	1234567890;
061457	061457;
1	1;
<none>	?; (*)

*Use the special character described in "Catching Result Data", Section 5, under "Missing Fields", to tell the AD_X analyzer to ignore the operator ID.

If the AD_X analyzer is in operation and the host sends an operator ID, the host's operator ID is compared to the Operator ID entered from the AD_X analyzer keypad. If the two IDs differ, the error message "ERROR: OPERATOR ID MISMATCH" is printed.

If the AD_X analyzer is in operation and the host sends an operator ID but the operator did not enter an ID, that is considered an operator ID mismatch error. The error message "ERROR: OPERATOR ID MISMATCH" is printed.

If the AD_X analyzer is not in operation, any operator ID received from the host is ignored.

Note:

THE OPERATOR ID RECEIVED FROM THE HOST IS USED ONLY FOR VERIFICATION OF THE OPERATOR ID ENTERED INTO THE AD_X ANALYZER AT THE START OF THE RUN. THIS OPERATOR ID IS NOT INSERTED INTO THE INTERNAL AD_X ANALYZER DATA STRUCTURES NOR DOES IT APPEAR ON THE AD_X ANALYZER PRINTOUT. IT IS ONLY USED TO CROSS-CHECK THE ID ENTERED IN THE AD_X ANALYZER.

Sending Sample IDs

CAROUSEL IDs

The carousel ID string is numeric, less than or equal to 5 digits in length, separated from other fields with delimiters, and set to '?' if missing. Some examples with the delimiter already appended are:

Carousel ID	Transmitted Data
23	23;
1	1;
<none>	?; (*)

*Use the special character described in "Catching Result Data", Section 5, under "Missing Fields", to tell the AD_X analyzer to ignore the carousel ID.

Note:

ALTHOUGH THE CAROUSEL ID STRING HAS RESERVED SEVERAL DIGITS FOR A VALUE, THE CURRENT AD_X CAROUSELS ARE NUMBERED FROM 1 TO 26.

If the host sends a carousel ID and there is a carousel ID barcode on the carousel in the instrument, the two carousel IDs are compared. If they differ, the error message "ERROR: CAROUSEL ID MISMATCH" is printed.

If the AD_X analyzer is in operation and the host sends a carousel ID but the carousel in the instrument does not have a carousel barcode number, then that is not considered an error. No error message is printed.

If the AD_X analyzer is not in operation, the carousel ID received from the host is ignored.

Note:

A NUMBERLESS CAROUSEL CANNOT BE ASSIGNED A NUMBER BY THE HOST. JUST LIKE THE OPERATOR ID, THE CAROUSEL ID FROM THE HOST IS USED TO VERIFY THE ID ON THE RUNNING CAROUSEL. IT IS NOT INSERTED INTO THE INTERNAL AD_X ANALYZER DATA STRUCTURES NOR DOES IT APPEAR ON THE PRINTOUT. IT IS ONLY USED TO CROSS-CHECK THE CAROUSEL NUMBER ALREADY IN THE AD_X ANALYZER.

THE PARAMETER **NEED CARSL ID**, DESCRIBED IN SECTION 4, "HUMAN INTERFACE CHANGES", UNDER "CONFIGURE... MENU", CAN FORCE THE OPERATOR TO INPUT A CAROUSEL BARCODE NUMBER IF ONE IS NOT READ FROM THE CAROUSEL DURING THE VERIFICATION CHECKS.

CONTENTS ID FOR A LOCATION

Remember that 20 of the 21 locations on the AD_x carousel hold one of the following: a reagent cartridge, a cuvette or nothing. (Location 'R,' the zero location, must hold a reagent cartridge.)

A location with a cuvette can be subdivided. The cuvette could be one of three quality control levels (Low, Medium or High), a sample with an ID, a sample without an ID, or one of the six calibrators. These six permutations (L, M, H, with ID, without ID, calibrator) should be communicated to the AD_x analyzer as well.

REAGENT CARTRIDGES

To identify a location as holding a reagent cartridge, send the data in this format: Lxx:RGNTnnn;

The "xx" is the location of this cartridge on the carousel (use "00" for location 'R') and "nnn" is the unique Abbott assay number assigned to this assay. (Use the assay numbers described in the AD_x Operator's Guide.) For example:

Reagent Location	Name	Number	Transmitted Data
R	PCP	61	L00:RGNT061;
7	ACETA	30	L07:RGNT030;

Sending Sample IDs

CUVETTES AS QUALITY CONTROLS

To identify a location as holding a quality control (QC) cuvette, send the data in this format: Lxx:QCn;

The "xx" is the location of the QC on the carousel. The "n" will be one of the following: "L", "M" or "H" depending on the level of the QC. For example:

Quality Control Location	Level	Transmitted Data
1	Low (L)	L01:QCL;
5	Medium (M)	L05:QCM;
17	High(H)	L17:QCH;

Note:

IF THE HOST CALLS A CUVETTE A CONTROL, THEN THE AD_X ANALYZER WILL APPLY ITS INTERNAL QC RANGE CHECKING ROUTINES. THE RESULT FOR THAT CONTROL WILL BE COMPARED WITH THE RANGES STORED IN THE AD_X ANALYZER. VALUES FOUND TO BE OUTSIDE OF SPECIFICATION WILL BE FLAGGED ON THE PRINTOUT AND MARKED AS "OUT" IN THE RESULT RECORD. VERIFY THAT THE CORRECT RANGES HAVE BEEN PROGRAMMED FOR EACH ASSAY. FOR DETAILS, CONSULT THE AD_X OPERATOR'S GUIDE.

CUVETTES AS SAMPLES WITH AN ID

To identify a location as holding a sample cuvette, send the data in this format: Lxx:nnnnnnnnnn;

The "xx" is the location of the sample on the carousel and "nnnnnnnnnn" is the sample ID associated with the cuvette. The sample ID may be up to 10 characters long. For example:

Samples Location	Identification	Transmitted Data
1	1234567890	L01:1234567890;
2	J.DOE	L02:J.DOE;
2	SMITHY57	L02:SMITHY57;
9	1	L09:01;
8	2	L08:02;
7	3	L07:03;
6	4	L06:4;

Note:

ALL INCOMING SIDS WITH ALPHABETIC INFORMATION ARE FORCED TO UPPERCASE. REFER TO "NONSENSE DATA/DATA FILTERING," IN THIS SECTION, FOR MORE INFORMATION.

CUVETTES AS SAMPLES WITHOUT AN ID

To identify a location as holding a sample cuvette without any ID, send the data in this format: Lxx:NONE;

The "xx" is the location of the sample on the carousel. The "NONE" informs the AD_x analyzer that there is not an SID associated with this sample; cuvette is present, but is not identified.

CUVETTES AS CALIBRATORS

To identify a location as holding a calibrator, send the data in this format: Lxx:CALIB;

The "xx" is the location of the calibrator on the carousel. The "CALIB" informs the AD_x analyzer that there is a calibrator in this location. No distinction is made between calibrator levels.

EMPTY LOCATIONS

Finally, to identify a location as empty (no cuvette, no reagent cartridge) send the data in this format: Lxx:EMPTY;

The "xx" is the location of the empty location on the carousel. The "EMPTY" informs the AD_x analyzer that there is no cuvette in the location.

Sending Sample IDs

ORDER/OMISSIONS

The order of the information in the SID record is important. The first field must be the record ID; the second, the operator ID and the third, the carousel ID. After those three fields, the data content for a location can come in any order.

The host need not send Content IDs for every location on the carousel (but it's advisable). An omitted Content ID is set to "Blank" for that location. The operator, when invoking the Receive procedure, will be presented with a printout similar to:

pasteup illustration

The '–' character denotes "Blank", meaning that the host system did not send information about the location. The asterisk is used to alert the operator to a difference between the RECEIVED and CURRENT columns.

EXAMPLE SID FILE FROM HOST

A Sample ID file sent from the host computer to the AD_x analyzer might contain this data:

Data Transmitted	Description
SID1000 ;	Record ID
061457;	Operator ID
7;	Carousel ID
L00:RGNT030;	Carousel position R is Acetaminophen
L01:12345;	Carousel position 1 is a sample, ID 12345
L02:QCL;	Carousel position 2 is a Low control
L03:NONE;	Carousel position 3 is a sample, no ID
L04:EMPTY;	Carousel position 4 is empty
L05:EMPTY;	Carousel position 5 is empty
L06:EMPTY;	Carousel position 6 is empty
L07:RGNT040;	Carousel position 7 is Ethanol
L08:12345;	Carousel position 8 is a sample, ID 12345
L09:EMPTY;	Carousel position 9 is empty
... ...	... ...
L20:EMPTY;	Carousel position 20 is empty
<carriage-return>	ASCII 13
<line-feed>	ASCII 10

Note:

ANY LEADING WHITE-SPACE WAS REMOVED AND EACH FIELD IS ON A SEPARATE LINE FOR READABILITY ONLY.

Sending Sample IDs

OMISSIONS REVISITED

As mentioned above, the host system does not have to send content information for each location on the carousel. The AD_x analyzer will handle omissions gracefully. However, the host computer SHOULD send content information for all locations on the carousel.

Omitting location information will only serve to confuse the AD_x analyzer operator and limit the ability of both systems to double check the data. ***So don't omit anything.*** Send 21 Content ID records.

INSERTING RECEIVED SAMPLE IDS: SUCCESS OR FAILURE

The rules for matching the received sample IDs to the running carousel are as follows:

Match the operator IDs

- a NOT APPLICABLE from the host system matches anything in the AD_x analyzer

Match the carousel number

- a NOT APPLICABLE from the host system matches anything in the AD_x analyzer

If the AD_x analyzer is not running then

- copy data from RECEIVED into CURRENT
- blanks are not copied, whatever was in Current, stays there
- No mismatch errors can happen in this condition

If the AD_x analyzer is running then

- for each Content ID record received
  - ☐ consult the "AD_x Analyzer Running Compatibility Matrix" on the Received data versus the current carousel contents (see Appendix One)
  - ☐ if the matrix indicates error, stop checking immediately and set the error flag
  - ☐ if the error flag is set after the attempted transfer, insert the appropriate error message into the record "CSL0100 "

The host computer should now be able to recognize whether or not the SID data sent was valid for the running carousel.

NONSENSE DATA / DATA FILTERING

Incoming data is filtered by the Kermit Task before it gets to the Receive procedure. The eighth bit of every byte is reset; all control characters except CR and LF are converted to spaces; the delete character and the backslash (\ ) are converted to spaces; and all lower case letters are forced to upper case.

After the data is received, the fields are separated out and checked. The Receive procedure will consider the received data to be nonsense if:

- the first field is not a record ID “SID1000 ” or
- the second field is not an operator ID or
- the third field is not a carousel ID or
- any location data embedded in an SID is not within zero and twenty or
- any assay number received (RGNTnnn) is not an active assay

Sending Sample IDs

APPENDIX 1 – RUNNING COMPATIBILITY MATRIX

If the AD_x analyzer is running when the Receive procedure is invoked, then error checking is done to make sure that the received data matches the current carousel contents. This matrix shows how the AD_x analyzer determines whether the Receive procedure is successful in matching the received data with the current carousel contents – or not.

The possibilities for the "Current" carousel contents are listed in the columns, and the "Received" possibilities are in the rows. For instance, the column "Rgnt" means that there is a reagent cartridge in this position on the carousel. "Rgnt" in the row means that the host system has sent reagent cartridge information for the location on the carousel.

Abbreviations:

Rgnt	Reagent cartridge
QC	Quality control
SID	Sample with an identifier
Smpl	Sample without an identifier
Calib	Calibrator

Appendices

To check for compatibility, pick the column below that matches the data in the CURRENT column of the Receive procedure printout. Then pick the row below that matches the data in the RECEIVED column of the Receive procedure printout. The box at the intersection will be either “ERR” or “OK”. ERR indicates that the Receive procedure will fail with a mismatch error.

		CURRENT					
		Rgnt	**QC**	**SID**	**Samp**	**Calib**	**Empty**
RECEIVED	**Rgnt**	OK-1	ERR	ERR	ERR	ERR	ERR
	QC	ERR	OK-2	ERR	OK-3	ERR	ERR
	SID	ERR	ERR	OK-4	OK-5	ERR	ERR
	Samp	ERR	ERR-6	ERR	OK	ERR	ERR
	Calib	ERR	ERR	ERR	ERR	OK	ERR
	Empty	ERR	ERR	ERR	ERR	ERR	OK
	Blank	OK-7	OK-7	OK-7	OK-7	OK-7	OK-7

Notes:

1 The assay numbers must match, if not ERR.

2 The levels (L, M, H) must match, if not ERR.

3 Received QC data can overwrite an unidentified sample.

4 The IDs must match exactly, if not ERR.

5 Identified samples can overwrite an unidentified sample.

6 A current control comes from either the presence of a QC barcode clip or operator entry using the QC/ID INFO... menu. Both imply operator intervention and yield ERR.

7 Received blanks get no error checking (see “Omissions Revisited” in Section 6).

APPENDIX 2 – DEFINITIONS

The glossary defines several computer terms and some of the AD_x analyzer terminology used in this specification.

ASCII

American Standard Code for Information Interchange; a 128 character code used almost universally by computers for representing and transmitting character data. The AD_x analyzer uses, and expects the host to use, the ASCII character set.

---

alphanumeric

This term is used to classify characters. A character is considered "alphanumeric" if it is in the range of '0' to '9', 'A' to'Z', or 'a' to 'z' in the ASCII collating sequence.

In context, the strict classification is often relaxed to include other ASCII characters such as the period, comma or colon. But within this text, it should be construed in the strictest sense.

---

alphabetic

This classification includes the upper and lower case letters of the alphabet.

---

assay

The qualitative or quantitative analysis of a substance or drug, the substances itself or the analysis itself.

---

calibration

The procedure used to establish a range for determining the concentration of an unknown sample. The AD_x analyzer uses 6 determined standards or "calibrators" as part of the calibration procedure. The useful result of a calibration is the "calibration curve", a plot of the calibrator concentrations versus the millipolarization or intensity values obtained during the run.

control

A carefully produced standard of a known concentration which is often used to validate a run.

---

control character

An ASCII character in the range 0 to 31 or 127. Produced on typical computers by holding the CTRL key and typing another character.

CRLF

The Carriage Return (CR) character (ASCII 13) followed by the Line Feed (LF) character (ASCII 10). Also denoted as “CR/LF”.

---

dilution

If an unknown sample is too concentrated, it is sometimes diluted in an attempt to bring the concentration into the range of the instrument.

FPIA

Fluorescence Polarization Immunoassay. An assay technology that uses the principle of fluorescence and rotational differences between large and small molecules to determine the concentration of the analyte.

---

high / low

Typically flags on the AD_x analyzer printout that indicate the measured concentration lies outside of the calibration curve. Results that come from “off curve” situations should be repeated.

---

millipolarization (mP)

A unit of measurement in FPIA assays that determines the printed concentration of analyte.

numeric

This character classification is relaxed in the specification to include the digits, ‘0’ to ‘9’ along with the period (‘ . ’), the exponential character (‘E’, ‘e’) and the sign characters (‘+’, ‘–’).

---

printable ASCII

A conjunct to “alphanumeric”, this character classification includes those characters from the space character (ASCII 32) to the tilde character (ASCII 126) inclusive.

---

qualitative

A result that does not give a numeric value. Typically “Yes” or “No”; “Pass” or “Fail”; “In” or “Out”; “Above T” or “Below T”.

---

quantitative

A result with a numeric quantity.

REA®

Radiative Energy Attenuation. An assay technology that uses the principles of fluorescence and absorbance to determine the concentration of an analyte.

---

semi-quantitative

A chemical analysis to determine a numeric approximation of the total quantity of the various members of the drug class and their metabolites.

therapeutic limits

Some analytes have a therapeutic range of concentrations; they are most effective within this range. Results outside of the range are usually flagged by the AD_x analyzer to draw the attention of the clinician. Not used for Abused Drug Assays.

threshold

A predetermined concentration which implies a presumptive presence or absence of a drug in the sample. All presumptive positive results should be confirmed.

APPENDIX 3 – SAMPLE DECODER C PROGRAM

```
/*****************************************************************
**
**              Nagit – Takes an ADx analyzer (base 10) Nag code
**                  and translates it into semi-meaningful text
**
**                  27–Jun–89       Patrick Conroy
**
*****************************************************************/
#include <stdio.h>

#ifdef          __STDC__                /* Function Prototypes */
int             main( void );
void            decode_nagcode( unsigned int );
#else
# define        const
#endif
#define         FALSE       0
#define         TRUE        (! FALSE)
/*   **
    These are the Nag Messages that appear at the start of Run, Cal or Panel.
    The order is important; each message in the array matches a bit in the nag code.
** */
static const char *nag_message[  ] = {
      "* WARNING: ADx NOT PHOTO CALIBRATED",
      "* WARNING: ADx NOT BOOM CALIBRATED",
      "* WARNING: ADx NOT TEMP CALIBRATED",
      "* WARNING: ADx FAILED PHOTO CHECK",
      "* WARNING: ADx FAILED PIPETTE CHECK",
      "* WARNING: ADx FAILED TEMP CHECK",
      "* WARNING: ADx NOT CAROUSEL CALIBRATED",
      "* WARNING: EXTERNAL THERMISTOR FAILURE",
      "* WARNING: ADx WARMUP OVERRIDDEN",
      "", /* bit 9 – not used */
      "", /* bit 10 – not used */
      "* TIME FOR: PIPETTE CHECK",
      "* TIME FOR: TEMP CHECK",
      "* TIME FOR: PHOTO CHECK",
      "* WARNING: ADx SERIAL NUMBER NOT SET",
      "" /* bit 15 – not used */
      };
      #define   MAX_NAGS ( sizeof(  nag_messages ) / sizeof( char * ) )
```

```
/*============================================================== */
main( )
{
        /* ** variables ** */
        char            user_input[ 80 ];
        unsigned int    nagcode;
        int             user_quits;

        /* ** code starts ** */
        user_quits = FALSE;
        while (!user_quits) {
                printf( "Enter nag code (0 to quit) : " );
                gets( user_input );             /* get code from user */

                nagcode = atoi( user_input ); /* string to number */
                if (nagcode == 0)
                        user_quits = TRUE;
                else
                        decode_nagcode( nagcode ); /* show 'em the messages */ }
} /* main */
/*============================================================== */
void decode_nagcode (nagcode)
        unsigned int nagcode;
{
        /* ** variables ** */
        int  bit;
        /* ** code starts ** */
        /* Display it in hexadecimal — just for grins */
        printf( ":: Nag Code [%0X] ::\n", nagcode );

        for (bit = 15; bit >= 0; bit –= 1) {            /* 16 bits/word */
              if ((nagcode & (1 << bit)) == 0)          /* bit is RESET */
                 if (strlen( nag_message[ bit ] ) > 0)  /* ignore msg 9,10,15*/
                        puts( nag_message[ bit ] );
        }
        puts( "" );
} /* decode_nagcode */
```

APPENDIX 4 – DETERMINING THE INSTRUMENT SERIAL NUMBER

The serial number of the AD_x analyzer must be known before the start of a Kermit session. The AD_x analyzer will reject a Kermit transfer if the serial number in the file name does not match the number on the instrument.

The AD_x analyzer supports the "ENQuire" control character.

For example, when sending an AD_x analyzer (with revision V2.3 software and a serial number of 430) a control-E (^E) followed by an end of line character (CR, CRLF, LF, or Escape), the instrument responds with this string:

ADx 00430 V2.30<CRLF>

The response string is "ADx <serial number> <software revision>". The serial number is 5 digits, zero padded. The revision is a capital 'V' followed by digit dot digit digit, as in V2.30 or V3.00.

Note:

IF THE USER HAS EDITED SERVER TO 'Y' (YES), THEN THE AD_X ANALYZER IGNORES ANY ^E RECEIVED. IT IS LOOKING FOR VALID KERMIT PACKETS AND ^E IS NOT A VALID KERMIT PACKET.

In summary, to determine the serial number of the AD_x analyzer, have the host computer send a "^E <EOL>" sequence and wait for the "AD_x nnnnn Vx.xx" response.

APPENDIX 5 – DEBUGGING THE CONNECTION

Several features are in the AD_X software to aid in debugging the serial port connection. They are all accessed via special "direct addresses" using the AD_X analyzer keypad. To use them, the AD_X analyzer must be in the WARM UP or READY state and at the SYSTEM... SERIAL PORT... menu. The AD_X analyzer display should read:

READY (or WARM UP)
SERIAL PORT...

Edit the AD_X analyzer SYSTEM... SERIAL PORT... BAUD RATE parameter to match the host system and make sure that the SYSTEM... SERIAL PORT... SET UP... SERVER parameter is set to 'N' (No).

Testing the Connection

The first debug aid will help determine if the host system is properly connected to the AD_X analyzer. Press 8.232.1 ENTER on the AD_X analyzer keypad and the analyzer will transmit 25 lines of ASCII text out of the port. The host should receive 25 lines of data:

0123456789.,!ABCDEFGHIJKLMNOPQRSTUVWXYZ<CRLF>

Testing Sample IDS

If the AD_X analyzer detects a problem with a sample ID record, it merely prints a "nonsense data" error. More specific information on the cause of the error can be helpful when testing the host software. Press 8.232.4.1 ENTER on the AD_X analyzer and "RS232: AICS QC/ID ERRORS ON" will print on the AD_X printer. Once this debug code is enabled, more specific error messages will appear on the printout during the Receive procedure.

When finished debugging the Sample ID files, be sure to turn this option off by pressing 8.232.4.0 ENTER. "RS232: AICS QC/ID ERRORS OFF" will print. Do not run the AD_X analyzer with this option on.

APPENDIX 6 – EXAMPLE OUTPUT AND FILES

This is an example of a successful calibration run on the AD_x analyzer. Upon completion, the results of the run were transmitted to the host computer and stored in the file R0322402.ADX.

pasteup illustration

Appendices

This is the contents of the file R0322402.ADX which held the data from the successful calibration run. Two records, the RGT0500 record and the CCI0600 record, are too long to fit within the margins and are shown wrapped.

Since the carriage return and line feed characters are not printable, their location is shown with “CRLF.”

This is an example of a unsuccessful calibration run on the AD_X analyzer. Upon completion, the results of the run were transmitted to the host computer and stored in the file R0322403.ADX.

pasteup illustration

Appendices

This is the contents of the file R0322403.ADX which held the data from the unsuccessful calibration attempt. Two records, the RGT0500 record and the CCI0600 record, are too long to fit within the margins and are shown wrapped.

Since the carriage return and line feed characters are not printable, their location is shown with “CRLF.”

This is an example of a sample run on the AD_X analyzer. Upon completion, the results of the run were transmitted to the host computer and stored in the file R1960101.ADX.

Quality control samples were in locations 4 and 10. The control in location 10 was within the allowable range and the control in location 4 was outside the range. The sample in location 14 is the only sample in this run without a sample ID. Samples 7, 8 and 9 were greater than the Cocaine assay threshold and sample 14 had an error that prevented the determination of the final result. The plus sign ('+') next to the blank value for sample 14 indicates that the value was above the allowable limit.

Appendices

pasteup illustration

This is the contents of the file R1960101.ADX which held the data from the sample run. The RGT0500 and CSL0100 records are too long to fit within the margins and are shown wrapped.

Again, the carriage return and line feed characters are not printable so their location is shown with “CRLF.”

Appendices

This is an example of a panel run on the AD_x analyzer. Upon completion, the results of the run were transmitted to the host computer and stored in the file R0075704.ADX.

This panel run ran three samples on three assays. The results for the sample in location 1 (SID of 041586) was less than the threshold for Cocaine Metabolite assay. A NET I SMALL error prevented the determination of the Methadone assay result. The results for sample 2 (SID of 041587) were above the thresholds for the Cocaine Metabolite and Methadone assays. Sample 3 (SID of 041588) had results similar to sample 2.

pasteup illustration

Appendices

This is the contents of the file R0075704.ADX which held the data from the panel run. The RGT0500 records are too long to fit within the margins and are shown wrapped.

Since the carriage return and line feed characters are not printable their location is shown with “CRLF.”