Program

AlphaStation

Version

1.20

Description

AlphaStation is designed to forward seismic data from a Nanometrics data acquisition system (NaqsServer) to an IDC data center, using the CD-1 data format and protocol defined in IDC 3.4.2 (May 1998). Each instance of the AlphaStation program represents a single IDC "station", which may include up to 100 seismic data channels from different Nanometrics stations. Data are sent to the IDC as CD-1 data frames. Frames have a fixed time duration (typically 10 seconds) and contain data for all channels comprising the IDC station. The data for each channel within a frame is called a channel subframe.

Typical Operation

AlphaStation subscribes to real-time data from NaqsServer via TCP socket using the Naqs datastream interface, and maintains a second TCP connection to the IDC data center. During normal operation, CD-1 frames are built and sent to the IDC in near real time as data are received from Naqs. Occasionally data may be missing or delayed due to telemetry errors. In this case, AlphaStation will wait for missing data until a configurable timeout has expired, then send the incomplete frame (with missing samples set to -2^31). If the connection to IDC is lost, or if the bandwidth of the connection is (temporarily) insufficient to handle the required data rate, data frames are buffered, then sent in modified LIFO order (or optionally in FIFO order) when the connection is re-established. For short term outages, frames are buffered in memory. For longer term outages, frames are rebuilt using data from the Naqs ringbuffers.

To provide error recovery, AlphaStation maintains a history file which records the transmission status of each data frame as one of the following:

sent (complete) - complete frames which were successfully sent
sent (incomplete) - frames with missing data which were successfully sent
never sent - frames which were never built or never sent.

When AlphaStation station is started, it updates the history files to include frames which should have been built while it was shut down, and marks them as never sent. When AlphaStation has established a connection to the IDC receiver, it sends frames with the following priority within the available connection bandwidth:

1. current data (real-time data) in time order
2. old data frames which have never been sent,
3. old frames which were incomplete when originally sent (optional)

Gap recovery (transmission of old frames which have never been sent) may be done either in FIFO order (oldest first) or modified LIFO order (most recent first).

Retransmission of incomplete frames is carried out by a background process which runs at a configurable time delay behind real time. Frames which were incomplete when originally sent are rebuilt from the Naqs ringbuffer, and retransmitted if any of the missing samples have been recovered since the initial send. This feature provides the IDC receiver with the most complete data record available, while giving priority to real time data.

Data Compression

IDC 3.4.2 specifies that data for each channel may be sent in uncompressed format (as 4-byte IEEE integers) or in the Canadian compressed format (a second-difference compression format). AlphaStation supports both of these transmission modes.

Authentication

Starting with version 1.20, AlphaStation provides optional data authentication following the IDC 3.4.2 format. When authetication is enabled, each channel subframe is signed with a 40-byte

signature generated using the DSA signature algorithm. Data signing capability is provided by an external PCMCIA encryption token which implements the PKCS11 interface. This provides complete security, because the private key used to sign the data cannot be exported from the token. The token is required for data signing, but if authentication is not required, AlphaStation can be run without an encryption token.

State of Health

The CD-1 format as defined by IDC 3.4.2 provides a 32-bit field within each channel subframe defining the channel status for the current frame. AlphaStation defines these bits as follows:

1. status bit 31 (1 = dead channel): this bit is not used in v1.20; AlphaStation will not send an empty channel subframe (a subframe with no valid data samples)
2. status bit 30 (1 = zeroed data): not implemented
3. status bit 29 (1 = clipped data): not implemented
4. status bit 28 (1 = calibration in progress): this bit may be mapped to one of the external SOH channels of the digitizer; it will be set if the scaled SOH value exceeds 1.0 at any time during the frame. The selected SOH channel should be used to monitor the calibration-enable signal on the digitizer, and scaled appropriately (e.g. 0 = normal, 5 = calibration enabled).
5. status bit 24-27: undefined
6. status bit 23 (1 = vault door open): this bit may be mapped to an external SOH channels of the digitizer; it will be set if the scaled SOH value exceeds 1.0 at any time during the frame. The selected SOH channel should be used to monitor the vault door switch or transducer, and scaled appropriately (e.g. 0 = closed, 5 = open).
7. status bit 22 (1 = authentication box opened): this bit may be mapped to an external SOH channel of an instrument; it will be set if the scaled SOH value exceeds 1.0 at any time during the frame. The selected SOH channel should be used to monitor the vault door switch or transducer, and scaled appropriately (e.g. 0 = closed, 5 = open).
8. status bit 21 (1 = equipment moved): not implemented
9. status bit 20 (1 = clock differential too large): this is set if the GPS is unlocked and the estimated maximum clock differential exceeds the configured limit.
10. status bits 16-19: not defined
11. status bit 15 (1 = supply voltage out of range): this bit is set if the supply voltage to the digitizer falls outside the configured limits. This bit may be mapped to an external SOH channel; or, it may use the voltage level given in the digitizer's internal SOH report.
12. status bit 14 (1 = internal temperature out of range): this bit is set if the internal temperature of the digitizer falls outside the configured limits.
13. status bits 0-13: not defined

Summary of Inputs and Outputs

Required Input Files

Configuration File - AlphaStation.ini

This file defines all operating characteristics for the AlphaStation program.

Instructions for configuring AlphaStation are given in a later section.

Naqs Ringbuffers

AlphaStation requires read access to the Naqs ringbuffers (for outage recovery). These files must reside on the same machine or be accessible over a LAN on a shared drive.

Other Inputs

AlphaStation receives online data from NaqsServer Data Stream service. This requires a TCP connection to Naqs. See the reference manual section entitled "NaqsServer Online Data Streams."

Output Files Produced

Log File - Alpha.log

The log file contains diagnostic messages generated by AlphaStation and provides a summary of the program's operation. Each log message has an associated severity:

F	Fatal errors which require immediate system shutdown.
E	Errors - serious abnormal occurrences which may affect data integrity.
W	Warnings - less serious abnormal occurrences
I	Informational messages tracing the normal operation of the system.
V	Detailed informational messages tracing the normal operation of the system.
D	Debug - verbose trace messages generated during normal operation.

The log may be configured to show only messages at or above a specified severity level, by adjusting the verbosity setting.

History File - AlphaHistory.nmx

This file records the transmission status of each frame built by AlphaStation. This information is used to determine which frames need to be retransmitted to IDC. This file is a ringbuffer which keeps the transmission status for the most recent few days; the duration of the history file is set in the AlphaStation.ini file. This file is stored in binary format, but its contents may be listed using the ViewHistory program provided with AlphaStation.

Running AlphaStation

Starting AlphaStation

AlphaStation runs within a command window under the Microsoft Windows NT operating system. It may be started manually from the command line, by changing directory to c:\nmx\user, and typing:

AlphaStation

Alternatively,AlphaStation can be started automatically by the Nanometrics Watchdog program. The latter is more usual for an operating network. The syntax for the watchdog entry is given below.

Stopping AlphaStation

AlphaStation should always be shut down by typing "Quit <CR>" in the AlphaStation window. This ensures that all files are closed and all system resources are released.

Command Window Interface

The AlphaStation window displays all log messages generated by the AlphaStation acquisition program. AlphaStation also supports a limited keyboard interface. The following options are supported:

"D <CR>"	Change the log verbosity to DEBUG. This will display all log messages.
"V <CR>"	Change the log verbosity to VERBOSE. Debug messages are suppressed.
"I <CR>"	Change the log verbosity to INFO. Debug and verbose messages are suppressed.
"QUIT <CR>"	Stop AlphaStation and exit. This is the preferred way to stop AlphaStation, since it ensures that all files are closed and system resources are released.

Configuring AlphaStation

Purpose of Configuration

Before running AlphaStation , you must create a configuration file, AlphaStation.ini, to configure AlphaStation for your particular network. The main purposes of the configuration file are:

- } to provide connection information (addresses and ports) for IDC and Naqs,
- } to provide a mapping between IDC channels and Nanometrics channels,
- } to provide detailed operating parameters for each channel.

Inifile Structure

The AlphaStation configuration file is structured as an ***inifile***, a text file format which is designed to be readable and easy to edit. An inifile consists of a number of ***sections***, each containing several ***parameters***. Sections are identified by a name enclosed in square brackets (e.g. [Interface]). Each parameter is given on a separate line following the section identifier, in the format: ParameterName = Value. For example, the section defining the network connections for AlphaStation is typically:

```
[ Connections ]
  NaqsAddress = 199.71.138.13
  NaqsPort = 28000
  IdcAddress = 193.81.205.6
  IdcPort = 9050
```

Data order and default values

All parameters for a given section must appear after the section identifier for that section, and before any other section identifier. AlphaStation does not provide any default settings. Therefore, it is necessary to fully define every parameter in each section. Parameters must be defined in the order specified in the detailed configuration instructions below.

White space and comments

The inifile reader ignores white space and blank lines, so white space can be added anywhere within an inifile if desired to improve readability. Also, the inifile reader recognizes the double slash ("//") as a comment delimiter, so comments may be added anywhere in a file. The following are example comments:

```
// This is a full line comment
[Interface]          // a comment may follow a section header
  NaqsAddress = 199.71.138.13 // a comment may follow a parameter definition
```

Comments are useful for adding descriptive information to the inifile, or for temporarily removing parameters or sections from the file.

Inifile Error Detection

AlphaStation parses the ini files on startup. If it detects any errors (unrecognized fields or illegal values), it will print an error message and stop. To resume, fix the file using a text editor, then restart AlphaStation. The most common cause of unrecognized fields are:

Misspelled parameter names. Check the spelling carefully, and note that parameter names are case-sensitive.

Missing names. If a parameter appears out of order or in the wrong section, it will not be recognized.

Duplicated names. If a parameter name appears twice, the second instance will not be recognized.

Illegal values are values which are undefined or out of range for a particular parameter. The permitted values for each parameter are given in the detailed configuration instructions below.

Definition of Inifile Parameters

The AlphaStation configuration file (AlphaStation.ini) contains the following sections:

[AlphaStation]
[Connections]
[TxParameters]
[Authentication]
[AlphaLog]
[Station]
[Channel]

There must be exactly one section of each type, except [Channel]; there may be up to 100 channel sections. All sections and all parameters within sections must be in the specified order. The contents of each section are outlined below.

[AlphaStation]

This section serves as a header for the configuration file and defines the IDC station name for this installation. This must be the first section in the file.

StationCode

Definition: The registered station name or code for this AlphaStation, provided by IDC.

Permitted Values: An 8-character string.

[Connections]

This section defines the IP address and port for connecting to Naqs and the IDC Connection Manager. It contains the following parameters:

NaqsAddress

Definition: The IP address or host name of the NaqsServer machine.

Permitted Values: A valid host name or dotted decimal IP address.

NaqsPort

Definition: The port number of the Naqs datastream service (typically 28000).

Permitted Values: Any positive integer.

IdcAddress

Definition: The IP address or host name of the IDC Connection Manager.

Permitted Values: A valid host name or dotted decimal IP address.

IdcPort

Definition: The port number of the IDC Connection Manager.

Permitted Values: Any positive integer.

[TxParameters]

This section defines the CD-1 transmission characteristics of the station.

FrameTimeLength

Definition: The duration of each CD-1 data frame, in seconds.

Permitted Values: Integer values from 5 to 60 (typically 10 seconds).

MaxFrameDelay

Definition: This is the maximum time in seconds that a data frame will be delayed waiting for missing data. If this time is exceeded, the incomplete data frame will be sent with missing samples set to -2^31, and marked in the history file as incomplete. Incomplete data are retransmitted later if missing data are recovered.

Permitted Values: Integer values from 30 to 300 (typically 120 seconds).

Contiguity

Definition: The minimum number of frames that should be sent in forward order. This is the distance between marks in the modified LIFO ordering scheme (see GSE paper 243 dated 20 July 1995).

Permitted Values: Integer values from 1 to 10 (typically 3).

StackSize

Definition: The maximum number of unsent frames that will be buffered in memory during a transmission outage to IDC.

Permitted Values: Integer values from 5 to 30 (typically 15).

MostRecentFirst

Definition: This defines the order in which unsent frames will be sent to IDC after a transmission outage.

Permitted Values: 1 = send most recent frames first (LIFO order)
0 = send oldest frames first (FIFO order)

RetxEnabled

Definition: This indicates if incomplete frames should be retransmitted if more data becomes available.

Permitted Values: 1 = retransmission of incomplete frames is enabled.)
0 = retransmission of incomplete frames is disabled.)

RetxDelayMinutes

Definition: If retransmission is enabled (see above), incomplete frames are retransmitted later if the missing data have been recovered. This parameter defines the time delay from the nominal start time of the frame until it is retransmitted. This should be set roughly equal to the memory capacity of the digitizers (in minutes), to allow the maximum time for recovery of missing data before retransmitting incomplete frames.

Permitted Values: Positive integer.

TxHistoryHours

Definition: The duration in hours of the transmission history file. This defines the data storage capacity of the AlphaStation. This should be set roughly equal to the ringbuffer storage capacity of the data-acquisition system.

Permitted Values: Positive floating point value, typically 12 to 168.

[Authentication]

This section defines parameters for the authentication token used to sign channel data. To sign data, AlphaStation must log on to the token as a user, then create a signature using a private key stored on the token. This signature may be verified by the receiver using the corresponding public key. Public/private key pairs may be generated on the token using the Nanometrics TokenUtility program. There may be any number of keys stored on the token; each is identified by a unique ID. Use the TokenUtility program to manage keys on the token.

TokenID

Definition: The serial number of the PKCS11 token to be used for authentication.

Permitted Values: The serial number as an ASCII string, or "any" to use any token.

PIN

Definition: The user PIN code required to log on to the PKCS11 token.

Permitted Values: The valid PIN code as a string.

KeyID

Definition: The ID of the private key stored on the token to be used for signing data.

Permitted Values: A positive integer (typically 1 to 5) representing the private key ID.

[AlphaLog]

This section defines the name and location of the AlphaStation log file, and its verbosity setting at startup.

LogFilename

Definition: The base name for the AlphaStation log file. AlphaStation creates a new log file every day; the log file name is determined by appending the date (yyyymmdd) to the base name.

Permitted Values: Any valid file name.

LogDirectory

Definition: The directory in which the AlphaStation log file should be stored.

Permitted Values: Any valid directory name.

Verbosity

Definition: The startup verbosity of the AlphaStation log file.

Permitted Values: DEBUG, VERBOSE or INFO. Normal value is INFO.

[Station]

This section defines parameters such as state-of-health threasholds which apply to the entire Alpha station (i.e. all channels).

MaxSampleRate

Definition: This is the maximum sample rate for all channels of this station. It is used (together with number of channels and frame time length) to compute the maximum frame size in bytes.

Permitted Values: Positive integer, typically 40 or 100.

SohChannelName

Definition: The name of the Nanometrics state of health channel which contains status information which applies to all channels. This is typically SOH from an instrument located at the central acquisition site. This SOH channel name must be defined in the Naqs.stn file. In version 1.20, this includes only the state of health channel monitoring the authentication door status.

Permitted Values: A string in dotted format (e.g. STN01.SOH).

SohDirectory

Definition: The absolute pathname for the disk directory containing the Naqs ringbuffer for the above SOH channel. AlphaStation needs read access to the Naqs ringbuffers in order to transmit or retransmit old data.

Permitted Values: A valid pathname (e.g. E:\sohbuffers).

AuthenticationDoorSoh

Definition: This defines which state of health channel is mapped to the authentication door status bit for this station. The appropriate SOH channel should be used to monitor a transducer or microswitch on the authentication door.

Permitted Values: slow1, slow2 or slow3: use slow external SOH channel 1, 2 or 3
fast1, fast2 or fast3: use fast external SOH channel 1, 2 or 3
anything else: authentication door is not monitored

MinPower

Definition: This defines the lower limit of the acceptable range for digitizer supply voltage (in volts). If the supply voltage drops below this value at any time during a frame, status bit 15 will be set for the corresponding channel(s).

Permitted Values: Any floating point value.

MaxPower

Definition: This defines the upper limit of the acceptable range for digitizer supply voltage (in volts). If the supply voltage rises above this value at any time during a frame, status bit 15 will be set for the corresponding channel(s).

Permitted Values: Any floating point value.

MinTemp

Definition: This defines the lower limit of the acceptable temperature range for the digitizers (in degrees C). If the digitizer temperature drops below this level at any time during a frame, status bit 14 will be set for the corresponding channel(s).

Permitted Values: Any floating point value.

MaxTemp

Definition: This defines the upper limit of the acceptable temperature range for the digitizers (in degrees C). If the digitizer temperature rises above this level at any time during a frame, status bit 14 will be set for the corresponding channel(s).

Permitted Values: Any floating point value.

ClockDiffThreshold

Definition: This defines the maximum acceptable differential in microseconds between the digitizer clock and Universal Time. If the actual or estimated clock differential exceeds this threshold during a frame, status bit 20 will be set for this channel

Permitted Values: A positive integer (typically 4000 microseconds = 4 ms).

[Channel]

This section defines parameters which apply to a single data channel. There should be one [Channel] section for each channel of the IDC station.

NmxChannelName

Definition: The channel name defined by NaqsServer. This defines the data source for this channel. This channel must be defined in the Naqs.stn file.

Permitted Values: A string in dotted format (e.g. STN01.BHZ).

IdcChannelName

Definition: The registered IDC name for this channel, as provided by IDC. Both the NmxChannelName and the IdcChannelName must be unique, to ensure that each Nanometrics channel maps to a single IDC channel.

Permitted Values: A string (up to 16 characters).

SohChannelName

Definition: The name of the Naqs state of health channel containing status information for this data channel. This SOH channel must be defined in the Naqs.stn file. One SOH channel will contain status information for all data channels on a given digitizer.

Permitted Values: A string in dotted format (e.g. STN01.SOH).

RbfDirectory

Definition: The absolute pathname for the disk directory containing the Naqs ringbuffer for this channel. AlphaStation needs read access to the Naqs ringbuffers in order to transmit or retransmit old data.

Permitted Values: A valid pathname (e.g. E:\ringbuffers).

SohDirectory

Definition: The absolute pathname for the disk directory containing the Naqs ringbuffer for the above SOH channel. AlphaStation needs read access to the Naqs ringbuffers in order to transmit or retransmit old data.

Permitted Values: A valid pathname (e.g. E:\sohbuffers).

CalibrationFactor

Definition: The calibration factor for this channel in nm/count.

Permitted Values: A positive floating point value.

CalibrationPeriod

Definition: The period in seconds for which the above calibration factor is valid.

Permitted Values: A positive floating point value.

Authentication

Definition: Indicates whether data for this channel is authenticated.

Permitted Values: 0 = authentication OFF
1 = authentication ON

Compression

Definition: Indicates whether data for this channel should be sent in compressed format using the Canadian compression scheme.

Permitted Values: 0 = compression OFF (data sent as 4-byte IEE integers)
1 = compression ON

CalibrationChannel

Definition: This defines which slow state of health channel is mapped to the calibration status bit for this channel. The appropriate SOH channel should be used to monitor the calibration enable signal for the digitizer.

Permitted Values: slow1, slow2 or slow3: use slow external SOH channels 1, 2 or 3
fast1, fast2 or fast3: use fast external SOH channels 1, 2 or 3
anything else: calibration is not monitored

VaultDoorChannel

Definition: This defines which slow state of health channel is mapped to the vault door status bit for this channel. The appropriate SOH channel should be used to monitor a transducer or microswitch on the vault door.

Permitted Values: slow1, slow2 or slow3: use slow external SOH channel 1, 2 or 3
fast1, fast2 or fast3: use fast external SOH channel 1, 2 or 3
anything else: vault door is not monitored

PowerChannel

Definition: This defines which slow state of health channel is mapped to the supply voltage status bit for this channel. The appropriate SOH channel should be used to monitor the supply voltage for the digitizer.

Permitted Values: slow1, slow2 or slow3: use slow external SOH channel 1, 2 or 3
fast1, fast2 or fast3: use fast external SOH channel 1, 2 or 3
0: use the supply voltage reading in the digitizer internal SOH
anything else: calibration is not monitored

Example Configuration File

```
[ AlphaStation ]

StationCode = STN001

[ Connections ]

NaqsAddress = 199.71.138.13  // Naqs server IP address
NaqsPort    = 28000          // Naqs server port number

IdcAddress  = 199.71.138.13  // IP address  of IDC Connection Manager
IdcPort     = 2000           // port number of IDC Connection Manager

[ TxParameters ]

FrameTimeLength = 10   // in seconds
MaxFrameDelay = 60     // in seconds  (not less than 10 Secs.)
Contiguity    = 3      // in frames
StackSize = 6          // in frames
MostRecentFirst = 1    // Send order for delayed frames:
                       // 1 = most recent first; 0 = oldest first
RetxEnabled = 0        // Retransmission of imcomplete frames
                       // 1 = enabled, 0 = disabled
RetxDelayMinutes = 20  // time in minutes to wait
                       // before retransmitting incomplete frames
TxHistoryHours = 1     // duration of the tx history file in hours

[Authentication]

TokenID = any          // tokenID for the authentication token to use
PIN = tokenbpin        // the pin code to login token
KeyID = 1              // the ID of the key used for signing data

[ AlphaLog ]

LogFilename  = Alpha.log
LogDirectory = d:\user\AlphaStation
Verbosity    = VERBOSE

[ Station ]

MaxSampleRate = 200                 // Max sample rate for all channels
SohChannelName = STN01.SOH          // soh channel containing station status
SohDirectory = e:\nmx\user\RingBuff  // directory containing soh ringbuffer
AuthenticationDoorSoh = fast1       // fast or slow 1, 2, or 3
                                    // otherwise not monitored

MinPower = 10.                      // in volts
MaxPower = 15.                      // in volts
MinTemp = 0                         // in degree Celsius
MaxTemp = 40                        // in degree Celsius

ClockDiffThreshold = 2000           // in microSeconds

[Channel]

NmxChannelName = STN01.BHZ          // Nanometrics dotted station-channel name
IdcChannelName = IDC01/ch1          // IDC station-channel name
SohChannelName = STN01.SOH          // soh channel containing channel status
RbfDirectory = e:\nmx\user\RingBuff  // directory containing data ringbuffer
SohDirectory = e:\nmx\user\RingBuff  // directory containing soh ringbuffer

CalibrationFactor = 1       // nm/count
CalibrationPeriod = 5       // seconds
Authentication = 0          // 1 = authenticated, 0 = not authenticated
Compression    = 0          // 1 = compressed, 0 = uncompressed
CalibrationChannel = slow1  // fast or slow 1, 2, or 3; otherwise not monitored
VaultDoorChannel   = fast2  // fast or slow 1, 2, or 3; otherwise not monitored
PowerChannel       = slow3  // fast or slow 1, 2, or 3;
                            // 0 means use internal instrument SOH

[Channel]

NmxChannelName = STN01.BHN           // Nanometrics dotted station-channel name
```

```
IdcChannelName = IDC01/ch2            // IDC station-channel name
SohChannelName = STN01.SOH            // soh channel containing channel status
RbfDirectory = e:\nmx\user\RingBuff   // directory containing data ringbuffer
SohDirectory = e:\nmx\user\RingBuff   // directory containing soh ringbuffer

CalibrationFactor = 1        // nm/count
CalibrationPeriod = 5        // seconds
Authentication = 0           // 1 = authenticated, 0 = not authenticated
Compression    = 0           // 1 = compressed, 0 = uncompressed
CalibrationChannel = slow1   // fast or slow 1, 2, or 3; otherwise not monitored
VaultDoorChannel   = fast2   // fast or slow 1, 2, or 3; otherwise not monitored
PowerChannel       = slow3   // fast or slow 1, 2, or 3;
                             // 0 means use internal instrument SOH

[Channel]

NmxChannelName = STN02.BHZ            // Nanometrics dotted station-channel name
IdcChannelName = IDC02/ch1            // IDC station-channel name
SohChannelName = STN02.SOH            // soh channel containing channel status
RbfDirectory = e:\nmx\user\RingBuff   // directory containing data ringbuffer
SohDirectory = e:\nmx\user\RingBuff   // directory containing soh ringbuffer

CalibrationFactor = 1        // nm/count
CalibrationPeriod = 5        // seconds
Authentication = 0           // 1 = authenticated, 0 = not authenticated
Compression    = 0           // 1 = compressed, 0 = uncompressed
CalibrationChannel = fast1   // fast or slow 1, 2, or 3; otherwise not monitored
VaultDoorChannel   = fast2   // fast or slow 1, 2, or 3; otherwise not monitored
PowerChannel       = 0       // fast or slow 1, 2, or 3;
                             // 0 means use internal instrument SOH
```

Required Files

alphastation.bat, alphaStation.jar, alphastation.ini

CanadianCompression.dll, jpkcs11.dll

Nanometrics DLL release 1.5 for Win32

Installation

AlphaStation must be installed either on the NaqsServer machine, or on a machine which has TCP/IP access to the Naqs machine, and network access to the Naqs ringbuffers. It can be installed as follows:

1. Ensure that Java 1.2.2 or higher is installed on the machine.
2. Create directories c:\nmx\bin and c:\nmx\user if they do not exist.
3. Copy AlphaStation.bat, AlphaStation.jar, jpkcs11.dll and CanadianCompression.dll to c:\nmx\bin
4. Install Nanometrics DLL release 1.5 for Win32 to c:\nmx\bin
5. Copy AlphaStation.ini to c:\nmx\user and edit as appropriate for your installation
6. Add c:\nmx\bin to your default path.

Running AlphaStation via Watchdog

AlphaStation may be automatically started and monitored by the Nanometrics watchdog program by adding the following entry to your watchdog.ini file*:

[WatchEntry]

ProgramTitle = AlphaStation
ExitAction = Restart
PingsSemaphore = true
StartDelay = 6s
SessionType = Default
ScreenDisplay = Default
InitXPos = 0

```
InitYPos = 0
InitXSize = 0
InitYSize = 0
ProgramPathname = "java -cp c:\nmx\bin\AlphaStation.jar AlphaStation"
WorkingDirectory = "c:\nmx\user"
```

* Note: remember to increase the number of entries by 1 in the header of watchdog.ini.

Log Messages

AlphaStation produces the following log messages to indicate different error conditions:

Fatal Errors

Fatal errors are unrecoverable errors for which the application stops. These include:

AlphaStation configuration error, if the configuration file contains errors.

Thread died ... quitting ..., if a component thread stops running properly.

Errors

Error messages are reported by the following modules:

AlphaStation

Unable to open info file, if it cannot open the history file. The program will exit. Move or delete the history file, then retry.

MessageClient is not Running Ok, if thread MessageClient died;

RTFrameBuilder is not Running Ok, if thread RTFrameBuilder died;

FrameSender is not Running Ok, if thread FrameSender died;

RbfFrameBuilder is not Running Ok, if thread RbfFrameBuilder died;

ConnectionManager is not Running Ok, if thread ConnectionManager died.

InfoFile

InfoFile Constructor Error, if it fails to open or create history file alphaHistory.nmx.

MessageClient

Failed to connect to NaqsServer, if it fails to connect to Naqs Server;

Data channel not found in NaqsServer station file, if the specified channel name is not available from Naqs. Correct the channel name in the AlphaStation.ini file.

Soh channel not found in NaqsServer station file, if the specified channel name is not available from Naqs. Correct the channel name in the AlphaStation.ini file.

Warnings

Warning messages are reported by the following modules. These are less critical errors.

RbfFrameBuilder

Can't open RBF File, if the specified ringbuffer file cannot be opened

Can't open SOH File, if the specified SOH ringbuffer file cannot be opened

ConnectionManager

Error in sending..., indicating a write error on the TCP connection to IDC. The connection will be closed, then reopened again after 60 seconds.

Exception in getPAF(), if it can't connect to IDC connection manager;

Exception in connectToDLM(), if it can't connect to IDC disk loop manager;

Exception reading PAF, if there is a read error while reading the port assignment frame from the IDC connection manager.

For all of the above cases, AlphaStation will try to reconnect after 60 seconds.

Unknown IDC Alert type, if it receives an undefined alert. The alert will be ignored.

ConnectionManager

Trouble Shooting

Check the Connection to NaqsServer

When AlphaStation connects to NaqsServer, you will see the following log message:

MessageClient...(2) Connected to Naqs (thomass:28000)

"thomass:28000" is the IP address and port of the NaqsServer datastream service.

If AlphaStation cannot connect to NaqsServer, the following message will be reported:

Failed to connect to NaqsServer

plus a reason. If AlphaStation cannot connect to NaqsServer, check to make sure that NaqsServer is running and that the Naqs address and port are specified correctly in the AlphaStation.ini file.

Check the Connection to IDC

When AlphaStation connects to the IDC Connection Manager, you will see the following log message:

ConnectionManage(1) Connected to IDC CM (address:port)

where address:port is the IP address and port of the IDC connection manager.

When AlphaStation connects to the IDC DiskLoop Manager, you will see the following log message:

ConnectionManage(2) Connected to IDC DLM (address:port)

where address:port is the IP address and port of the IDC diskloop manager.

If AlphaStation cannot connect to IDC, one or more of the following messages will be reported:

Exception in getPAF()

Exception in connectToDLM()

If AlphaStation cannot connect to the Connection Manager, check that:

1. the IP address and port of the Connection Manager are specified correctly in AlphaStation.ini
2. there exists a network connection between the AlphaStation machine and IDC
3. TCP connections from AlphaStation to IDC are enabled by all intervening firewalls
4. the IDC Connection Manager and Diskloop Manager are both running.

Check that data are getting to Naqs

If data for one or more channels are not arriving in real time, you may see messages similar to the following:

Channel subframe for channel STN01.BHE has no data

If AlphaStation is receiving data for other channels, this probably indicates a telemetry problem between the specified station and Naqs.

If Everything is OK

If everything is OK, you will see the following messages:

MessageClient...(2) Connected to Naqs (address:port)

ConnectionManage(1) Connected to IDC CM (address:port)

ConnectionManage(2) Connected to IDC DLM (address:port)

and then AlphaStation will be fairly quiet with no error or warning messages printed at the console window. AlphaStation reports a hourly report of transmission result as an INFO message. Change the log verbosity to VERBOSE to display a log message each time AlphaStation sends out a data frame. A typical log is shown below (for FrameTimeLength of 10 seconds):

V 2000-05-04 09:52:41 RTFrameBuilder..(5) RTF built 2000-05-04 09:50:30

```
V 2000-05-04 09:52:41  FrameSender.....(9) sent RTF 09:50:30 (32644 bytes) in 71 + 0 ms
V 2000-05-04 09:52:51  RTFrameBuilder..(5) RTF  built 2000-05-04 09:50:40
V 2000-05-04 09:52:51  FrameSender.....(9) sent RTF 09:50:40 (32636 bytes) in 110 + 0 ms
V 2000-05-04 09:53:01  RTFrameBuilder..(5) RTF  built 2000-05-04 09:50:50
V 2000-05-04 09:53:01  FrameSender.....(9) sent RTF 09:50:50 (32708 bytes) in 100 + 0 ms
V 2000-05-04 09:53:11  RTFrameBuilder..(5) RTF  built 2000-05-04 09:51:00
V 2000-05-04 09:53:11  FrameSender.....(9) sent RTF 09:51:00 (32700 bytes) in 90 + 0 ms
```

Environment

AlphaStation is a Java program and should be run with Java 1.2.2 (or higher) under the Windows NT 4.0 (or higher) operating system.

See Also

General Utilities - Watchdog

NaqsServer Data Aquisition System

NaqsServer Online Data Streams

This document information

Date created: 3/23/98
Date last revised: 1/22/01