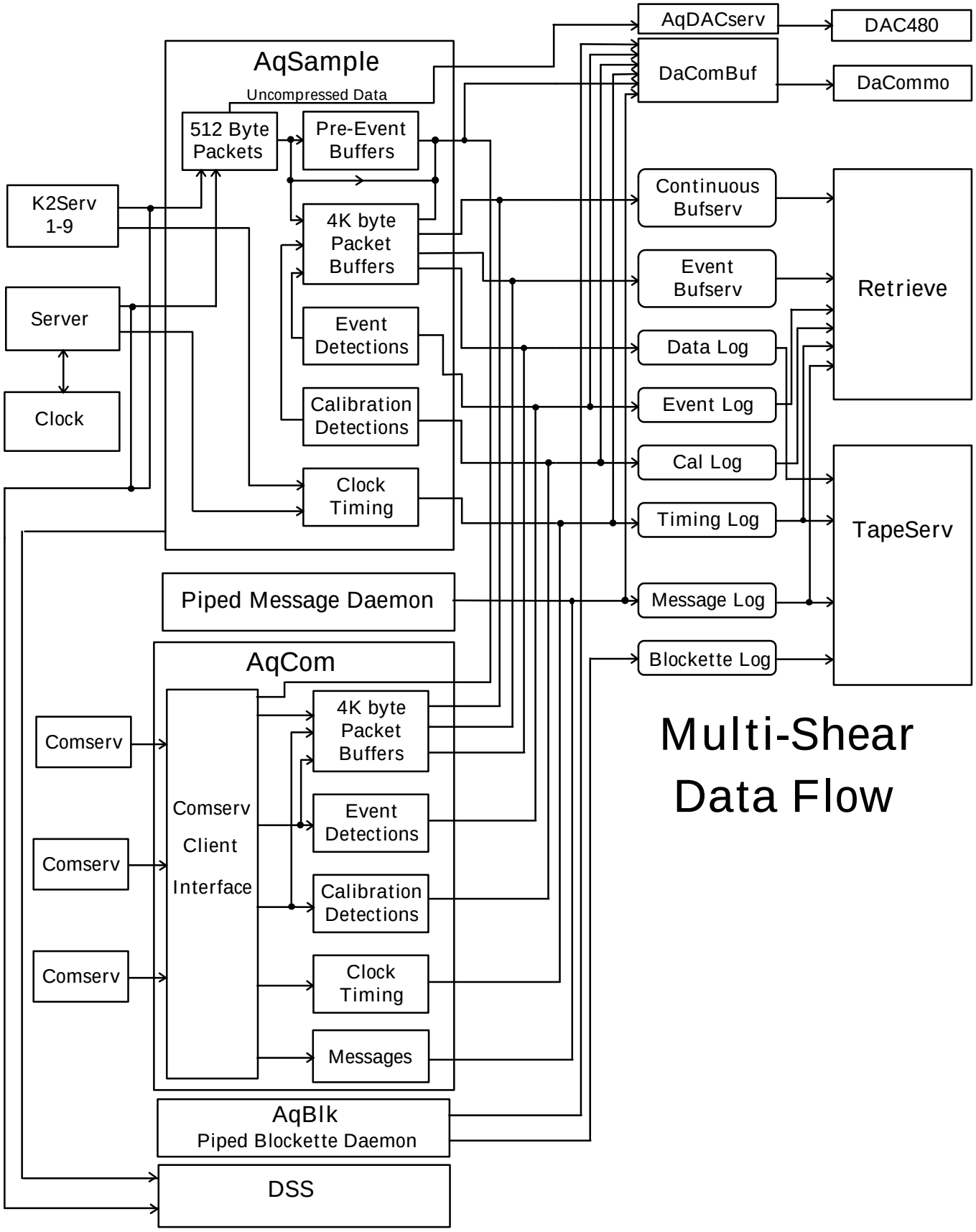
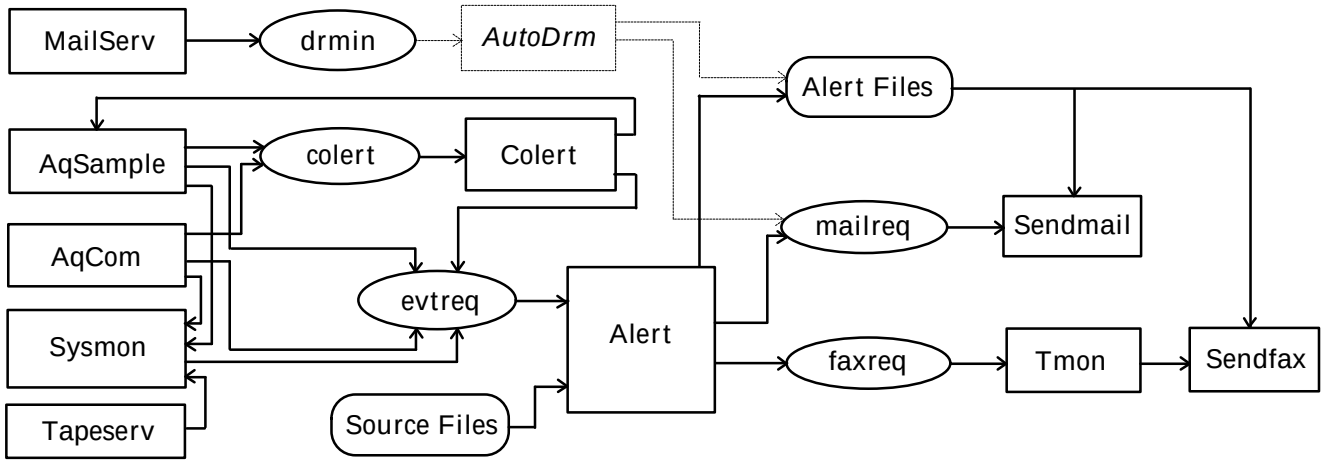
MULTI-SHEAR SOFTWARE PRELIMINARY CONFIGURATION
RELEASE 36/08-0131



This document describes Multi-Shear configuration. Additional background information can be obtained in the Quanterra SHEAR SOFTWARE CONFIGURATION GUIDE Revision B. Multi-Shear is a software package that is based on Ultra Shear and adds the capability to handle data for multiple stations. This product is still under development and capabilities and configuration are subject to change.

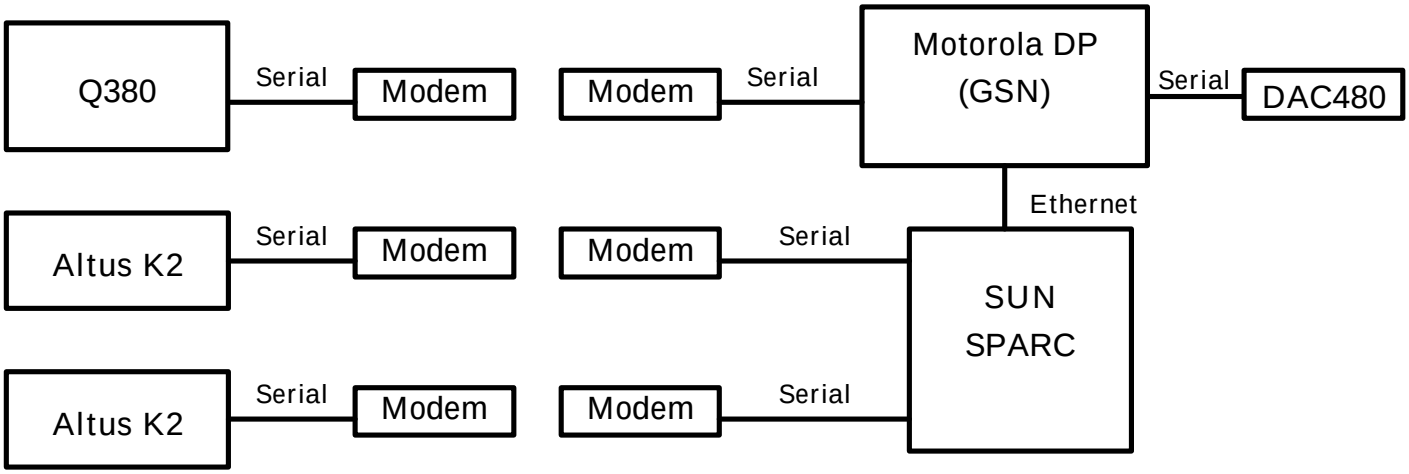


Multi-Shear Alarm & Event Flow

Configuration File

The configuration file (/r0/aqcfg) determines what data is to be recorded, how it is recorded, where it is sent, and where it comes from.

As an example of a config file, the following system is used :



The SUN is used as the final destination for the data from all three stations, as well as providing the serial port to Ethernet handling for the K2 digitizers. The DP provides tape storage and dial-up retrieval for the first station (provided by the Q380), in effect, the equivalant of a GSN DP. It also does the processing and compression of the K2 data and sends that data, and the data received from the Q380 to the SUN. The DAC480 is driven from the Z axis channels from the K2 digitizers. The ability to decompress the data received from the Q380 is not currently available.

The configuration file that runs on the DP is :

```
[aqdirs]
filtdir=/H0/MSHEAR/FILTERS tempdir=/R0 datadir=/H0/DAT aqdir=/H0/MSHEAR/LOGS alog=AQLog mlog=MsgLog
infodir=/r0
```

The ***filtdir*** must contain the file FIRFilters for use by AqSample. The combination of***tempdir***/***mlog*** is the file name of the message log used to refresh the status screen on AqShell. This is normally on Ramdisk for fast access. The combination of ***aqdir***/***alog*** is the file name of the message log that will be available for later access by retrieve.***Aqdir*** is also the directory where the timing, detection, calibration, and data logs are kept. Datadir will contain the data files created by bufserv that can be accessed by retrieve. ***Infodir*** has the location of the "vbb_system_description" file for the Retrieve "R" command, defaults to ***Aqdir***.

```
[aqstation]
Station=HRV0 lat=43.5392 lon=72.3758 elev=386 naz=0 eaz=90 source=0
Station=HRV1 lat=43.5392 lon=72.3754 eleve=399 naz=0 eaz=90 source=1
Station=HRV lat=43.5388 lon=72.3765 elev=403 naz=0 eaz=90 source=hrv ret=y to=360
```

This section is used to map a data source to a station name. There is one entry per line. More than one data source can be mapped to a station, however, this can cause problems with comlinks using the current comlink protocol due to lack of routing information. ***Station*** is the name that the data will carry, regardless of the source.***lat***, ***lon***, ***elev***, ***naz***, and ***eaz*** are only used for SAC output files from retrieve.

Source is either a single digit from 0 to 9 in which case it indicates a local digitizer, or a station name indicating it is from a remote station received by comserv. The source station can be different from the*station* name, in which the aqcom program will change all incoming packets to conform to the*station* name. If ***ret*** is enabled for a station, then it indicates to retrieve that there might be data available for that station.***to*** is used to change the data timeout for a remote station. If no new data is received during this period (in seconds) then the synthetic clock will stop running, which results in the station time being highlighted in Aqshell, indicating a lack of data. It defaults to 120 seconds.

```
[aqsignon]
Quanterra Multi-Shear Seismic Processor - Example
```

This line is used for the Aqshell Banner.

```
[retsignon]
Quanterra VBB Data Retrieval System
```

This line is used for the Retrieve sign on banner.

```
[aqglobal]
pass=4.6687725272870111d+025 netid=EX clkqflt=1200
det=500 tim=100 cal=100 data=2000 blk=1000 verbosity=2
chartname=/pipe/chart chartsize=200 level=3a
license=12345678901234567890 maxretrieve=3
```

This section sets values that are used by one or more programs in the acquisition package :

- ***level=n*** where *n* is 1, 2, 3F, 3S, or 3A. This is the default compression level if not specified in a LCQ. If not specified it is 1. 3F is Steim level 3, first differences. 3S is level 3, second differences. 3A is level 3, adaptive, which means that the first frame of each com record (normally 7 data frames) is compressed using both first and second differences, and which is best is used for the rest of the frame.
- ***pass=n*** where *n* is encoded password (moved here from the old ctlch section).
- ***det=n*** where *n* is the number of records for the detection log file. If not specified, then no detection log is used.
- ***tim=n*** where *n* is the number of records for the timing log file. If not specified, then no timing log is used.
- ***cal=n*** where *n* is the number of records for the calibration log file. If not specified, then no calibration log is used.
- ***data=n*** where *n* is the number of records for the data log file (data to be written to tape, using tapeserv).
- ***blk=n*** where *n* is the number of records for the blockette log file. If not specified, then no blockette log is used.
- ***verbosity=n*** sets the default verbosity level for Aqsample, can be overridden on command line.
- ***blockettes=y*** inserts calibration and detection blockettes into 4K records written to disk and tape by aqsample.
- ***chartname=pipename*** sets the default chart pipe name. In an LCQ, if you specify "chart" then this name is substituted for the name.
- ***chartsize=size*** sets the size of the default chart pipe. Aqsample is designed to not write to the pipe if doing so would overflow the pipe, thereby hanging aqsample.
- ***timeout=n*** where *n* is the number of seconds to use as a timeout in Aqshell when determining whether Aqsample has stopped. The default is 5 and the maximum is 300. This value should be increased for digitizers that communicate through modems, such as possible with K2 digitizers.
- ***netid=xx*** where *xx* is the SEED network name. The network name is now required.

- ***clkqflt=n*** where *n* is the number of seconds to use as a clock quality filter. This value is used by Aqsample and Aqcom to avoid recording frequent clock messages when the quality is moving between 40-100%. It has no effect if the clock quality should drop below 40%. It is also used by the clock program to filter out clock status dumps from GPS engines when it drops in and out of lock. The default is no filtering.
- ***license=n*** where *n* is a 20 digit number used to terminate the function of Aqsample and Aqcom after a certain date.
- ***showall=y*** enables aqshell to show data values in the status screen for channels that are not written to comlink, tape, or disk by aqsample. This is a debugging feature to allow showing data for DSS source channels.
- ***maxretrieve=n*** where *n* is the maximum number of "Retrieve" programs that are allowed to operate at one time. This is normally used on network connected systems to limit total simultaneous users. Defaults to no limit.

```
[aqdetect]
ehon co1
```

This line is used to redefine the names of the 32 available comm events. The default names are COMM:1 through COMM:32. You can replace these with more descriptive names by just listing the names. The first name listed renames COMM:1, the second COMM:2, etc. An individual comm event is actually the logical or of 3 flags, local, remote, and coincidence. The local flag can be set using the "Local Events" menu option in Aqshell. The remote flag can be set over a comlink connection from the DP. The coincidence is set by the "colert" program.

```
[comlink]
levels=32 mprio=8 cprio=4 dprio=30 tprio=6 port=23400 ipaddr=128.103.105.230
resend=20 pkts=500 delay=10 rce=y ws=16 resendpkts=2 notify=y udp=y
station=hrv

[commo0]
levels=32 mprio=8 cprio=4 dprio=30 tprio=6 port=23401 ipaddr=128.103.105.230
resend=20 pkts=500 delay=10 rce=y ws=16 resendpkts=2 notify=y udp=y
station=hrv0

[commo1]
levels=32 mprio=8 cprio=4 dprio=30 tprio=6 port=23402 ipaddr=128.103.105.230
resend=20 pkts=500 delay=10 rce=y ws=16 resendpkts=2 notify=y udp=y
station=hrv1
```

These three sections describe the comlinks for the remote station (hrv), and the two local stations (hrv0 and hrv1). The are all identical except the station and IP port. These IP ports must match up with comservs running the receiving DP (whether it be a Multi-Shear system, or Comserv running on Unix).

Note that calibrations, messages, and timeing are given a fairly low priority, while detections are given a high priority. Detections are given high priority because downstream systems may be trying to correlate detections from various sources to generate alarms.

Since these connections are used over ethernet there are a few defaults that are overridden. Window size and the resend interval are increased to allow return packet latency caused by routers and busy computers. Udp is set indicating to use UDP instead of TCP packets. This is more efficient in terms of network traffic and avoids connections being lost. Only use TCP if UDP results in excessive sequence errors. The parameters available for comlinks are :

- ***STATION=name*** specifies which station this dacommo is attached to. Data cannot be mixed in comlinks except if the comlink uses the data module name "anycom" and the station is "any".
- ***LEVELS=n*** where *n* is the maximum number of priority levels to be used. Must be between 1 and 100 and there is no default.
- ***MPRIO=n*** where *n* is the default priority level of message packets, this entry must be specified and be in the range of 0 to the number of levels. If 0 is specified, messages will not be transmitted.
- ***DETPRIO=n*** where *n* is the priority level of detection packets, if not specified, defaults to message priority.
- ***TIMEPRIO=****n* where *n* is the priority level of time correction packets, if not specified, defaults to message priority.
- ***CALPRIO=n*** where *n* is the priority level of calibration packets, if not specified, defaults to message priority.
- ***BLKPRIO=n*** where n is the default priority level of blockette packets, if not specified, defaults to message priority.
- ***DELAY=n*** where *n* is the polling delay in ticks for the comlink program "dacommo". This value defaults to 20 which is probably fine for 9600 baud links, 10 is recommended for 19200 or faster links as well as TCP/IP links.
- ***RESEND=n*** sets a timeout in seconds used to resend packets in the transmission window if no valid acknowledgements are received in this amount of time since the last packet was sent. The default is 5 seconds. If this is set to zero, then resends are disabled, which is not recommended, use a large value instead to avoid link lockup.

- ***PORT=n***. If the *IPADDR* parameter is set then this parameter is the TCP or UDP port number and must be between 5000 and 65535. If the *IPADDR* parameter is not set this is the name of the serial port.
- ***PKTS=n*** is the number of packets allocated for use by this dacommo. Each packet is 528 bytes long, so allocating 2000 packets would require a memory buffer of a little over 1MB.
- ***TIMEOUT=n*** where *n* is the time-out in seconds for the serial port, purpose unknown.
- ***WS=n*** where *n* is the window size. The window size must be between 2 and 64, with a default if not specified of 8.
- ***RCE=y*** will echo remote commands (excluding the remote commands that have responses, such as ULTRA_REQ). *RCE=n* to disable echo, which is the default.
- ***IPADDR=address*** sets DACOMMO to communicate via TCP or UDP rather than a serial port.
- ***UDP=y*** will enable the use of UDP packets instead of TCP if using a network connection.
- ***MAXRESENDS=n*** limits how many times packets can be resent without getting a acknowledge when using a TCP/IP connection. If this limit is exceeded the network connection will be closed and then re-opened. This parameter is not used for serial port or UDP connections. This is required for TCP connections since each time a packet is sent it uses system memory (MBUF) until the packet is acknowledged or until the connection is closed. A recommended value is 3.
- ***SYNCTIME=n*** sets the sync packet interval. If there are no packets waiting to be sent after this number of seconds since the last packet was sent, a sync packet will be set (regardless of grouping). The default is 10 seconds. If zero, no sync packets are sent.
- ***RESENDPKTS=n*** sets how many packets are sent out a time when resending packets. The default is 7, or WS-1, whichever is lower. When packets are resent, they are grouped into smaller units and each unit must be completely acknowledged before the next unit. When all packets that were in the window at the time resending began are acknowledged normal streaming mode is continued.
- ***NETDLY=n*** sets the number of seconds that Dacommo waits in between closing the last network connection and opening a new one. The default is 120 seconds. Only used for TCP connections.
- ***NETTO=n*** sets the network connection timeout. If no connection is made during this timeout, the connection is aborted and a new connection attempt is made after *NETDLY* seconds. The default is 60 seconds. Only used for TCP connections.
- ***GRPSIZE=n*** sets the size of packet transmission grouping. The default is 1 in which case there is no grouping, packets are sent as soon as they are available. If n > 1 then that many packets must be available before the whole group is sent. This parameter must be no higher than the window size - 1.
- ***GRPTIME=n*** sets a timeout so that if *GRPSIZE* packets are not available, after this timeout, whatever packets are available are sent. The default is 0, so this parameter must be set if*GRPSIZE* is > 1.
- ***NOTIFY=y*** will enable DP notification of change in sampling status. If enabled the DACOMMO will enter a "link acknowledge" phase when starting or when Aqsample starts or stops. This phase will send a LINK_PKT to the DP and will continue to do this every RESEND seconds until the DP sends back a ULTRA_REQ packet. In order to use this option, the DP must support the notify option. This option is only valid when using QSL format.
- ***OFLOW=y*** will enable hardware output flow control. DCD must be active and CTS is used to enable/disable transmission by the DA.
- ***FMT=<format>*** will set the communications format, which defaults to***QSL*** (Quanterra Smart Link). This format is based on the standard dacommo format in use for a number of years (now called***Q512***) in that it uses 512 byte records. QSL expands on this by only sending the amount of data required, rather than always sending 512 bytes of data. This reduces comlink loading as well as CPU loading on both ends. The packet sequence will also have a larger modulus than the window size and will be the largest multiple of the window size less than or equal to 256. This is to provide more secure link aknowledgments, should some old packets get stuck in a modem somewhere, you won't be acknowledging data from a previous window. In order to use this format the dpcommo must be modified to :
    1) Check for the "ETX" byte anywhere within the record transmission, indicating end of record. It must not interpret an ETX preceeded by a "DLE" escape character as end of record however. The end of the record can be found by subtracting the size of the error control packet (6 bytes) from the location of the ETX.
    2) Must stop decompressing frames when the number of samples is used up, rather than decompressing all possible frames within a record.
    3) Before processing any records, must send a "LINK_REQ" packet, and then wait for the "LINK_PKT" for details on format and sequence modulus. See the "Ultra Shear DP↔DA Data Structures documentation for details on these packets.

Q512 is provided for backwards compatibility with older Quanterra systems, those systems should be updated as soon as possible. This format sends fixed length 512 byte packets, and with a sequence modulus the same as the window size.

- ***KEEPNEW=y*** will remove oldest packets from a queue before newest packets when packet buffers are required by a higher priority queue. Default operation is to remove newest packets before oldest.
- ***FLOOD=y*** enables flooding by default. This can be turned on and off over the comlink and by using the comset routine.

```
[logcommo]
levels=2 mprio=1 port=23403 ipaddr=128.103.105.230
resend=20 pkts=500 delay=10 rce=y ws=16 resendpkts=2 notify=y udp=y
station=log
```

This comlink is only used by the piped message daemon program. "Pmsgd" will send messages from all stations into this comlink, indicating from which station it originated in the message if it knows.

```
[anycom]
levels=32 port=23402 ipaddr=128.103.105.230
resend=20 pkts=500 delay=10 rce=y ws=16 resendpkts=2 notify=y udp=y
station=any
```

This is a special comlink used to combine data from multiple stations onto the same comlink, which is normally forbidden. The data module name must be "anycom" and the station name must be "any" to avoid aborting aqsample and aqcom during configuration. The actual data will retain its station name, it will not be changed to "any".

```
[tape]
data=25 tim=5 msg=25 blk=25
changesec=120 mindays=60 logdir=/h0/mshear/logs
scsilog=/r0/scsilog
verbosity=n
bot=/r0/aqcfg
eot=/r0/aqcfg
path=/mt0
warnat=130 fullat=140 retry=5 throttle=5 timeout=10
append=y verify=y report=y retension=y
```

This section provides information for the "tapeserv" program. This configuration is for a single 2150 cartridge tape. The tape server is a background process that monitors tape activity, log activity, and writes the new data in all logs to the tape when any of the logs reaches the threshold for that log. The first part is the parameters global to all tapes :

- ***LOGDIR=path*** - specifies the directory where to search for log files to archived to disk. Any valid log file must have the string "log" somewhere in it's name, such as aqlog, data_log, etc.
- ***TIM=percent*** - specified at what percentage of the timing log being full where the logs will be written to tape. If not specified or *percent* is zero then the timing log will not be written to tape.
- ***MSG=percent*** - specified at what percentage of the message log being full where the logs will be written to tape. If not specified or *percent* is zero then the message log will not be written to tape.
- ***DATA=percent*** - specified at what percentage of the data log being full where the logs will be written to tape. If not specified or *percent* is zero then the data log will not be written to tape.
- ***BLK=percent*** - specifies at what percentage of the blockette log being full where the logs will be written to tape. If not specified or *percent* is zero then the blockette log will not be written to tape.
- ***MINDAYS=days*** - specifies how old a tape volume header must be (previous to the current day) for it to be automatically considered a blank tape.
- ***CHANGESEC=seconds*** - specifies how many seconds tapeserv will wait for you to change a tape or wait for a command confirmation.
- ***CACHEFILE=path*** - this optional parameter is the name of a disk cachefile created using the "makecache" command. To use the cachefile, "cache=y" must be specified for each drive to use the file. Note that cacheing is only available for Archive Viper 2150S tape drives.
- ***VERBOSITY=yes*** - enables more detailed messages when writing logs to disk.
- ***SHOWLOG=filename*** - creates a file that keeps a record of raw SCSI commands to the tape drives. This file can be read using the "showlog" program.
- ***BOT=filename{,filename}*** - Defines a list of files that will be written to tape as log records at the beginning of the tape, immediately following the volume header record.
- ***EOT=filename{,filename}*** - Defines a list of files that will be written to tape as log records at the end of the tape, followed by an "END OF TAPE" log message, and then the 40 end of file marks.
- ***PATH=path*** - specifies a new tape drive. All of the following commands reference only to the last tape drive specified using this parameter. Up to 4 path parameter may be specified.

These commands only apply to the last "path" specified :

- ***WARNAT=size*** - specifies when tape changing warning messages will start appearing. The size is in megabytes. If not specified then 130MB will be used for SCSI-1 drives and 1100MB for SCSI-2 drives.
- ***FULLAT=size*** - specifies when a tape change will be forced. If not specified then 140MB will be used for SCSI-1 drives and 1200MB for SCSI-2 drives..
- ***RETRY=count*** - the number of times tapeserv will try to write to a tape if there is an error when starting a write. This is intended for Archive DAT drives which sometimes think they are offline after a week or two. The default is one.
- ***THROTTLE=ticks*** - this inserts sleeps (for the specified number of ticks) between writing SEED records to the tape. If this value is not set, it may cause the system to lose digitizer data during tape writes.
- ***AUTOUNLOAD=y*** - indicates that you want to do an unload command when a tape is full. This has no effect on a cartridge tape, but will eject a DAT tape.
- ***RETENSION=y*** - indicates that the drive has a retension command available. Cartridge tapes do, DAT tapes don't.
- ***OVERWRITE=y*** - indicates that the program may overwrite data on this tape if there are not other available tapes and this tape has the oldest volume header.
- ***TIMEOUT=count*** - specifies how many times to retry a tape SCSI command if the drive returns a "busy" status. There is a one second delay between each attempt.
- ***VERIFY=y*** - indicates that the data written to a tape should be verified.
- ***CACHE=y*** - indicates that data should be first written to the cachefile on disk (or ramdisk) and then copied to the tape directly from the disk. Only Archive 2150S drives support this feature.
- ***APPEND=y*** - indicates that if a valid SEED volume header is found, that the drive should seek to end of data and check for filemarks. If no filemarks are found, then tapeserv will append data to this tape. If filemarks are found, or the APPEND flag is off, then the tape will be considered full.
- ***STREAM=y*** - indicates that the drive does not reliably support the normal mode of operation and that "dumb mode" should be used. Required for Archive 2525S drives.
- ***SEMILOAD=y*** - indicates that the drive should be put into a "semi-loaded" state between writing to logs. This is only valid on Hewlett Packard DAT drives.
- ***PREVENT=y*** - indicates that removal of the tape should be prevented on an active drive. Has no effect on cartridge drives.
- ***COMPRESS=y*** - enables compression on Hewlett Packard DAT drives. To allow this option to take effect, SW1 must be OFF and SW2 must be ON.
- ***REPORT=y*** - enables writing extended status information after an error condition.
- ***FATALTICKS=ticks*** - where *ticks* is the maximum number of systems ticks (normally 10ms) before a SCSI command is considered hopeless. The default is 360000 ticks (or about 1 hour). If this timeout occurs it is due to a drive firmware error and the drive will be hidden from tapeserv. To use that drive again the system must be reset.

```
[bufserv]
seg_c=2000 seg_e=10000 req_c=30 req_e=30 verb_c=2 verb_e=2 map_c=100 map_e=50
file=hrv_20hz rec_c=5000 rec_e=2000 mask=hrv.bh?
file=hrv_1hz rec_c=5000 mask=hrv.lh?
file=hrv_80hz rec_e=5000 mask=hrv.e??
file=hrv_0.1hz rec_c=1000 mask=hrv.v??
```

This section configures both the continuous and event bufservs. Parameters that are specific to the continuous or event bufserv end in "_c" or "_e". ***Seg_***x sets the number of segment descriptors, default is 1000. ***Req_***x sets the number of queued service requests, default is 20. ***Verb_***x sets the verbosity, default is 0. ***Map_***x sets the number of seed channels allowed in the map, default is 200.

A file description starts with ***file*** and ends when it gets to the next ***file*** or the end of the configuration section. The file parameter is the root file name. For instance, using the first file line as an example, the continuous bufserv will construct files with the names HRV_20HZ_C and HRV_20HZ_C_DIR as the data and directory files. The event bufserv will construct HRV_20HZ_E and HRV_20HZ_E_DIR. For online files a ***Rec_***x parameter is required, which is the number of 4K records for that file. There must be at least one ***mask*** entry for each file and multiple ***mask*** entries are allowed per file. The masks are normally used to group data with the same sampling rate (and from the same station) into the same file. You can specify a buffer file you have obtained from another system by using a ***Desc_***x parameter instead of the ***Rec_***x and ***mask*** parameters. The value for the ***Desc_***x parameter would normally be a one word indication of it's contents.

When aqsample or aqcom connect to a bufserv they perform a series of requests to map seednames for the channels that will be written to a 1 byte component number and 1 byte file number. Bufserv will search to find the first file that will accept the seedname and still has available components. If no match is found aqsample/aqcom will generate a warning message and that data cannot be written to bufserv. There are a maximum of 32 components (individual seed channels) allowable in each file. If you are making frequent changes in your configuration file during installation you may want to stop and restart the bufserv's to get rid of any components in files that were requested by aqsample/aqcom but where no data was ever written. If there is actually data written then bufserv will find these components when it starts up, the only way to get rid of them would be to delete the data files so bufserv starts with a clean slate.

If you need to access data files from previous version of mshear then you should use file names in the form of online_<stream> such as online_vbb.

```
[pmsgd]
printinit=d;1b;14;d log=hrv default=hrv comlink=logcommo
```

This section defines parameters for the piped message daemon (pmsgd). This program collects messages from various pipes created by most programs and routes them to comlinks and the message log.***printinit*** is used if pmsgd logging a message to a printer. ***Log*** is a list of the stations for which messages are to be stored in the message log. Note that there are really two message logs, the temporary log which is only used by Aqshell, and the permanent log, which is used by retrieve and tapeserv. This option refers to the permanent log only, messages from all stations are written to the temporary log. Some programs put out messages that are not specific for one station, in which case the***default*** station is used for that message. If ***comlink*** is specified, then messages from all stations are sent to that comlink.

```
[server]
auxerr=y
autostart=y
verbosity=y
port=23300
ipaddr=128.103.105.230
rate=100
channels=3
status=hourly

[server1]
auxerr=y
autostart=y
verbosity=y
port=23301
ipaddr=128.103.105.230
rate=100
channels=3
status=hourly
```

These two sections are used by the K2 servers. Configuration of the various digitizer servers supported is documented in a later section. The [server] section is used by digitizer number 0, and the [server1] section is used by digitizer number 1.

```
[alert]
evtsize=8192
faxid=QUANTERRA
faxport=/s1b
dir=/h0/alerts
templates=/h0/alerts/sources
mailid=steim@quanterra.harvard.edu
mailhost=128.103.105.194
```

This section configuration of the alert system. The alert system allows faxes and/or emails to be sent when it receives a detection message through it's "evtreq" pipe. The operation of this system is described later. The parameters are :

- ***faxqsize*** is the number of queued faxes allowed at any one time. If there are further faxes to be sent they are held in the "faxreq" pipe until there is room in the queue for them. The default is 25, and the maximum allowable is 1024.
- ***faxsize*** is the size of the fax request pipe, the default is 1024.
- ***faxid*** is the default sending FAX name if not overridden for a particular recipient. Many older FAX machines will only accept the numbers 0-9, new models usually support upper case letters as well as some punctuation. This string is limited to 20 characters and will normally appear on the display and log of the recipients fax machine as the sender of the fax.

- ***faxretry*** is the default specification for how often and how many times to try to send a fax, if not overridden for a particular recipient. The format of this string is discussed later. The default is "120TO960".
- ***faxport*** is only required in the unlikely case of two or more "tmon" programs to indicate which tmon will be handling sending the faxes.
- ***deadcnt*** specifies that a delay be added in tmon after the indicated number of faxes have been sent so that someone can dial-in. The default is 5.
- ***deaddly*** specifies how many seconds to wait after "deadcnt" faxes have been sent before starting to send faxes again. The default is 120 seconds.
- ***dir*** indicates where the fax and Email files are to be found. It also contains any ".qub" bit mapped files that might be referenced by faxes.
- ***templates*** is the directory that contains one or more files for each recipient. Any file with a suffix of ".htm" or ".html" in this directory is considered a template file. The format of these files is discussed later.
- ***evtsize*** is the size of the event request pipe "evtreq" created by the alert program. The default is 2048.
- ***mailqsize*** is the number of queued Email files allowed at any one time. If there are further Email files to be sent they are held in the "mailreq" pipe until there is room in the queue for them. The default is 25, and the maximum allowable is 1024.
- ***mailsize*** is the size of the mail request pipe, the default is 1024.
- ***mailretry*** is the specification for often to try to contact a SMTP server when there are Email files to send, the format of this string is discussed later. The default is "120TO960".
- ***mailtimeout*** is the maximum number of seconds between each step in the SMTP negotiation before the attempt is aborted. The default is 90.
- ***mailid*** is the default "sender" of an Email, unless overridden in the template for a recipient. This would normally be the acquisition system administrator.
- ***mailhost*** can be repeated up to 10 times and is the IP (Internet Protocol) address of a SMTP server. If there is a time-out reaching a server, the next server in the list will be tried, circling through the list. Once a server is found that works, it is used until it stops working. The format is xxx.xxx.xxx.xxx[:yy] where each xxx is a number between 1 and 255 and yy is the telnet port number for the SMTP server. The port number is normally 25, and this is the port used if not specified in the mailhost address. The address would normally be in the same domain as the acquisition systems, lacking a SMTP on your domain, you can probably arrange with someone else to use their server. No account seems to be needed to use a SMTP server, but you should arrange the use of a server in case someone notices "unauthorized usage" of their server and decides to ban your domain.
- ***verbose***, if y or yes, enables status messages on the "alert" program. The default is no.

```
[colert]
pipesize=2048
co1=2,10,240,At least 2 Z:SPWW Co-Detections on Stations HRV, HRV0, and HRV1
co1=hrv.bhz:spww,hrv0.hhz:spww,hrv1.hhz:spww
```

This section is for the "colert" program which is used to detect the coincidence of one or more detections within a certain time limit from one or more sources. The following entries are allowed :

- ***PIPESIZE=size*** of the "coreq" pipe created by colert, the default is 1024 bytes.
- ***DEPTH=n*** where *n* is a number between 1 and 10 indicating how many detection onset times to keep queued for each detector. The default is 3. If smaller, a new detection could replace an older one that actually matches up with other detectors and therefore defeat a coincidence detection. Larger values require more processing time.
- ***name=minimum,window,duration[,message]*** where *minumum* is fewest detectors in the group that are triggered within the *window* to constitute a coincidence detection. When detected, if there is a *message* specified, and there is a [alert] section, the *name* of the detector and the *message* will be piped to the alert program. If the *name* is listed in the [aqdetect] section, the flag in aqsample will also be set for the *duration* of the detection. Note that these detectors can be retriggered, if a new set of events occurs in the window while the *duration* time has not expired, the duration counter is reset, but new alert is sent. This is always the first line of a detector specification, a new detector is defined when the *name* changes.
- ***name=detector{,detector}*** where *name* must be the same as the preceeding line and each detector has the standard detector name form of STATION.[LOCATION-]SEEDNAME:DETECTOR-NAME. If all the detectors do not fit onto one line, you may have multiple lines.

```
[sysmon]
tapes=10,60,60,60,60
hrv=5,1,60,60,60,60
hrv0=5,1,60,60,60,60
hrv1=5,1,60,60,60,60
clocks=hrv,hrv0,hrv1
verbosity=1
tolerance=5
```

This section sets values for use by the sysmon program :

- ***tapes=nodrive,tapeok,tapeerr,tapenod,tapechg*** - *nodrive* specifies the number of minutes that can elapse with no tape in the drive before the TAPENOD alert is generated. This is required since there will be no drive selected when the system first starts up, and when changing drives. The number for *tapeok* is the minimum number of minutes between successive generations of the TAPEOK alert signifying that the tape is running OK. Likewise *tapeerr* is the filter value for the TAPEERR alert (error on drive), *tapenod* is the filter value for the TAPENOD alert (no drive selected), and *tapechg* is the filter value for the TAPECHG alert (drive changed).
- ***station=aqstart,timeout,ok,to,poor,deg*** - *aqstart* specifies the number of minutes that can elapse when sysmon first starts up before it will generate an alert indicating that acquisition is not operating (TO). This is required since sysmon is started before either the Aqsample or Aqcom programs. *Timeout* is used to determine how often to make sure new data has been registered by Aqsample or Aqcom, in minutes. The *ok, to, poor,* and *deg* parameters are the filter values in minutes to avoid generating alerts that flop back and forth between clock values. The alerts generated will all be in the format of <station>.alert where alert is OK (acquisition OK, clock quality >= 60%), TO (acquisition timeout, no new data within the timeout period), POOR (acquisition running, but with quality < 40%), and DEG (acquisition running, but with quality = 40%). An example would be "HRV.OK".
- ***clocks=station{,station}*** - determines which station's data reception times can be used to reset the OS9 system clock.
- ***verbosity=n*** - sets the verbosity level, default is zero. Setting to one will enable writing a message for each station in the "clocks" list indicating whether the clock was good during the sample period, and what the deviation was.
- ***tolerance=n*** - sets the clock tolerance level between the calculated data clock value and the OS9 clock. If the difference is above this value (in seconds), the OS9 clock will be reset. The default is 10 seconds.
- ***interval=n*** - sets the interval in minutes between checking clocks, the default is 60 (1 hour).
- ***threshold=n*** - sets the threshold above which a station is not considered stable enough to use for clock calculations, the default is 20.
- ***aqstatus=station{,station}*** - uses the listed stations to determine what status to display on the 4120 acquisition light. The WORST status of the stations listed is used.

```
[dacserv]
device=dac480
port=/t1
speedcor=5
```

This section is required if running the "aqdacserv" process to produce analog output from aqsample. Generating analog output from remote station is not currently supported. This will likely be a process that obtains the data directly from one or more comserv processes, based on work done at Berkeley.

Available options are :

- ***device=DAC480*** | ***MZ8610*** | ***XVME-505*** this option is required and specifies the Digital to Analog convertor board or unit to use.
- ***port=path*** is required only for the DAC480 and indicates which serial port it is connected to.
- ***blind=yes*** does not expect any response from a DAC480, assuming it is connected via a oneway connection.
- ***speedcor=n*** is used with the DAC480 to adjust how fast data is transferred to the DAC480.

A word of explanation is in order about the *speedcor* parameter. The system "tick" counter is used to generate an alarm at 1Hz rate. Should this clock be fast then the DAC480's buffer would fill up and the internal buffer would empty. Likewise, if too slow, the DAC480 will run out of data and the internal buffer will overflow. This can only be set by watching the Status display on the DAC480. There are 2560 samples for each channel on a DAC480, so each change of the display represents 256 samples. For instance, if, after 7 days the display change from averaging a "5" to a "3" that indicates 512 samples change. If the output rate is 80Hz this indicates that the incoming data rate was (512/80) 6.4 seconds slow over the 7 days, or (6.4/7) 0.914 seconds per day. The *speedcor* parameter should then be set to 0.914, or to avoid having to stop the aqdacserv program, you can use the setcorr utility to change it in real time :

```
$ setcorr 0.914 ↵
```

```
[dss]
high=2.5220277614434208D+035
middle=3.0412526690162778D+035
lowest=2.9398707340601414D+035
verbosity=1
maxcpu=10
maxbps=300
maxmem=20
port=5678
recon=60
```

This section configures the "Data Subscription Server". This program allows data to be sent to clients using UDP packets based on requests from the clients. See the DSS section later in this document.

- ***high=n*** is the encoded password for a highest priority client registration. There is no default.
- ***middle=n*** is the encoded password for a middle priority client registration. There is no default.
- ***low=n*** is the encoded password for a lowest priority client registration. There is no default.
- ***verbosity=n*** is the verbosity level. Level 1 reports new clients. Level 2 reports new report requests. Level 3 reports memory allocations. The default is 1.
- ***timeout=n*** is the number of seconds without getting a timeout reset packet from a client before the client is removed, the default is 600.
- ***maxcpu=n*** is the maximum amount of CPU time DSS is allowed to use in percent, the default is 20. If this amount is exceeded then reports are removed, lowest priority first.
- ***maxbps=n*** is the maximum amount of bytes per second to be transmitted, the default is 500. If this amount is exceeded then reports are removed, lowest priority first.
- ***maxmem=n*** is the maximum amount of memory in KB allowed to be used by DSS, exceeding this value results in DSS refusing new report requests. The default is 65 (K).
- ***port=n*** is the UDP port number that DSS will accept packets from, this should be between 5000 and 65000.
- ***recon=n*** is the number of seconds between trying to re-establish a data source connection, the default is 30.

Logical Channel Queues (LCQ)

Logical channel queues are what really control what data, on a channel by channel basis, goes to what storage or output device. They also used to define event detectors. LCQs are very similar between local stations (aqsample) and remote stations (aqcom), and unless otherwise noted, comments apply to both. No forward definitions are allowed. If you are deriving data from another LCQ, it must already be defined.

```
[aqlcqs]
*
* Altus K2 channels
*
seed=HHZ source=hrv0.100.2
mkmhh=spww fhi=2 flo=40 iw=160 nht=4 x1=33 x2=23 x3=8 xx=13 tc=500 wa=507 av=8
detector=k2z
comlink=commo0 epri=26 wecom=y
*
seed=HHN source=hrv0.100.1
detector=k2z
comlink=commo0 epri=26 wecom=y
*
seed=HHE source=hrv0.100.3
detector=k2z
comlink=commo0 epri=26 wecom=y
*
seed=ACE source=hrv0.clk
comlink=commo0 cpri=DEF wccom=y
*
seed=HHZ source=hrv1.100.2
mkmhh=spww fhi=2 flo=40 iw=160 nht=4 x1=33 x2=23 x3=8 xx=13 tc=500 wa=507 av=8
detector=k2z2
comlink=commo1 epri=26 wecom=y
*
seed=HHN source=hrv1.100.1
detector=k2z2
comlink=commo1 epri=26 wecom=y
*
seed=HHE source=hrv1.100.3
detector=k2z2
comlink=commo1 epri=26 wecom=y
*
seed=ACE source=hrv1.clk
comlink=commo1 cpri=DEF wccom=y
*
```

Yes/No entries can be shortened to Y and N. Some parameters also allow the Available option which can be shortened to A. The first line of a LCQ is the identification line and can have the following parameters :

- ***SEED=[location-]seedname SOURCE=data-source***. ***SEED*** specifies the seedname and optional location for this data. ***data-source*** has different formats based on the kind of data :
  - ***station.frequency.channel*** - data from a digitizer. station can be the station name, or the digitizer number.
  - ***station.aux.channel*** - digitizer auxiliary channels, these are considered 1Hz data.
  - ***station.status.channel*** - status channels, considered 1Hz data, these are :
    1 through ***10*** will be the number of free packets in the first ten dacommo links attached to this station.
    11 is the clock drift in microseconds for this station.
    12 is the clock frequency control value for this station, if relevant.
    13 is the percentage (times 100) fullness of the timing log, -1 if tapeserv not running.
    14 is the percentage (times 100) fullness of the message log, -1 if tapeserv not running.
    15 is the percentage (times 100) fullness of the data log, -1 if tapeserv not running.
    16 is the percentage (times 100) fullness of the first tape drive, -1 if tapeserv not running or "fullat" parameter not specified for this drive.
    17 through ***19*** are the fullness of the second through fourth tape drives, if configured.
    20 is the clock quality percentage (0 - 100).
  - ***station.clk*** - digitizer clock timing events.
  - ***station.seedname/filter*** - is used to take data recorded at a higher rate and filter it down to a lower frequency. The station and seedname must already be defined in the lcqs before this one. Filter is the name of the FIR filter to use. The resulting recording frequency will be the frequency of the source lcq divided by the decimation of the filter.

Subsequent lines may be one of three types :

- Detectors, either Murdock-Hutt, or Threshold. The entire line is used for parameters of the detector.
- Comlinks, the entire line is used for parameters.
- Charts, the entire line is used for parameters.
- General Options, options are separated by spaces.

Detector lines start with either :

- ***mkmhh=name*** where *name* is a new Murdock-Hutt detector. Description of parameters is covered started on page 101 of the SHEAR configuration manual.
- ***mkthr=name*** where *name* is a new Threshold detector.

The detector can now be referenced as station.[location-].seedname:name in the [aqcontrol] and [colert] sections. Parameters for the detectors are :

- ***detf=name*** specifies that the data is to be run through a IIR filter before running the detector, the filter must have already been loaded into memory using the loadiir program. If*name* is "remote" then the detector is loaded into the digitizer on a Q380 digitizer instead of being run in aqsample.
- ***off*** will initially disable the detector from running. It may later be enabled using the detector edit command from aqshell, or by remote command over a comlink.
- ***nolog*** will disable generation of Event Detection messages and log entries. It also disables writing to the alert and colert request pipes. If on, the only affect a detection will have is if it is used in a control detector equation.
- ***fhi=n*** is the Filhi parameter for Murdock-Hutt detectors or the high limit for Threshold detectors.
- ***flo=n*** is the Fillo parameter for Murdock-Hutt detectors or the low limit for Threshold detectors.
- ***iw=n*** is the Iwin parameter for Murdock-Hutt detectors or the hysterisis for Threshold detectors.
- ***nht=n*** is the N_hits parameter for Murdock-Hutt detectors or the number of sequential samples over a limit required for Threshold detectors.
- ***wa=n*** determines how long the detector will stay on after triggering. This is 1/4 of the actual count for Murdock-Hutt detectors and the actual count for Threshold detectors.
- ***x1=n*** is the Xth1 parameter for Murdock-Hutt detectors and not used for Threshold Detectors.
- ***x2=n*** is the Xth2 parameter for Murdock-Hutt detectors and not used for Threshold Detectors.
- ***x3=n*** is the Xth3 parameter for Murdock-Hutt detectors and not used for Threshold Detectors.
- ***xx*** is the Xthx parameter for Murdock-Hutt detectors and not used for Threshold Detectors.
- ***tc*** is the Def_tc parameter for Murdock-Hutt detectors and not used for Threshold Detectors.
- ***av*** is the Val_avg parameter for Murdock-Hutt detectors and not used for Threshold Detectors.

Comlink lines start with either :

- ***comlink*** indicates to use the default comlink, which uses a data module called "dacombuf".
- ***comlink=modname*** indicates to use a comlink that uses a data module called "*modname*".

The rest of the line contains the options :

- ***wccom=yes*** enables writing data or other packets from this LCQ to the comlink continuously. Instead of (y)es you can use (A)vailable, in which case the comlink output will be setup, but not enabled. It can be enabled using comlink commands.
- ***wecom=yes*** enables writing data packets to the comlink only when the control detector is on. The same available option is available as described above.
- ***cpri=n*** specifies the priority level to use for continuous data or other packets, 0 is off (default). If this LCQ is on a clock channel (ACE), message channel (LOG), or blockettes then using the keyword of "DEF" instead of a number will use the default timing, message, or blockette priority for this comlink.
- ***epri=n*** specifies the priority level to use for data when the control detector is on, 0 is off (default).
- ***detp=n*** specifies the priority level to use for detection packets, 0 is off. The default is to use the default detection priority for this comlink.
- ***calp=n*** specifies the priority level to use for calibration packets, 0 is off. The default is to use the default detection priority for this comlink.
- ***detcom=control-detector*** can be used to use a different control detector for this comlink. If not specified then the general control detector is used (described later).

Automatic dispatching of chart commands can be accomplished by specifying the parameters in the LCQ description if desired. Note that specifying minutes in addition to hours is not supported for automatic dispatch. All of the required parameters must be on the same line in the config file and***chart=*** must be the first one :

- ***chart[=pipename]*** where *pipename* is the pipe used to write the commands to. If the pipename is not specified, then the default pipename defined in the aqglobal section is used.
- ***ch_filt=iirname*** where *iirname* is the name of the optional iir filter to use.
- ***ch_gain=system-gain*** where *system-gain* represents the total gain including seismometer, digitizer, and any filter gains.
- ***ch_mag=magnification*** where *magnification* is at the specified frequency.
- ***ch_freq=hz*** where *hz* is the frequency at which magnification is valid.

- ***ch_hours=hours*** where *hours* is the number of hours (and therefor the interval) to be printed on one chart.
- ***ch_traces=count*** where *count* is the number traces to print on one chart.

General options available are :

- ***OFFSET=n*** will subtract *n* counts from digitizers before processing the data.
- ***REVERSED=yes*** will negate the data from a digitizer before processing the data.
- ***DETECTOR=<control-detector>*** specifies the "general detector". This "general detector" is used if a output destination does not define it's own detector.
- ***DETBUF=<control-detector>*** specifies the control detector to be used for writing to disk using bufserv.
- ***DETSEQ=<control-detector>*** specifies the control detector for use only on sequential (tape) output
- ***PEBUFS=n*** sets the number of pre-event buffers for this LCQ. When an event triggers, up to this many 512 byte compressed data packets will be output, containing the data before the event.
- ***WCSEQ=yes*** enables writing continuous data to the data log, messages to the message log, or blockette packets to the blockette log. This parameter supports the "available" option.
- ***WESEQ=yes*** enables writing event data to the data log. This parameter supports the "available" option.
- ***WCBUF=yes*** enables writing continuous data to the disk, and inserting detection and calibration packets into the 4K records written to disk. This parameter supports the "available" option.
- ***WEBUF=yes*** enables writing event data to the disk, and inserting detection and calibration packets into the 4K records written to disk. This parameter supports the "available" option.
- ***NODET=yes*** disables writing detection packets to the detection log.
- ***NOCAL=yes*** disables writing calibration packets to the calibration log.
- ***NOTIM=yes*** disables writing timing packets to the timing log.
- ***CLEV=n*** overrides the default compression level for this LCQ.
- ***GAP=samples*** will set the number of samples that will cause a flushing of the 4K records on the tape or disk. The default is 0.5 samples.
- ***CALDLY=seconds*** will set the amount of time after a calibration ends and the detection flag goes off. The default is 60 seconds.
- ***CAL0DB=value*** will define the voltage or current that corresponds to a 0dB calibration. This information is passed in the Calibration Blockettes.
- ***COMFR=count*** where *count* is the total number of frames (one is always used by the header) and must be between 2 and 8. If one of the dacommos is running in fixed length mode, then this option will be ignored and the count dictated by the dacommo will be used instead. This valud defaults to 8. (8 • 64 = 512 bytes).
- ***4KFR=count*** sets the maximum number of frames in a 4K disk or tape record.*Count* must be between 2 and 64 (64 • 64 = 4096 bytes). The actual number of data frames will always be an integral multiple of the number of data frames in a 512 byte dacommo record. Typical values are for Q512 or QSL dacommo format would be 7 data frames per dacommo record times 9 dacommo records is 63 data frames in a 4K record (plus one header frame). For MM256 dacommo format there are 3 data frames per dacommo record times 21 dacommo records is 63 data frames per 4K record. If you are running dacommo in QSL mode (or not running dacommo on this LCQ at all) and set*COMFR=4* and *4KFR=20*, then there would be 3 data frames per dacommo record and 6 dacommo records for a total of 18 data frames and one header frame in a 4K record.
- ***FLUSH=count*** sets the maximum number of outstanding frames in a 4K record before writing to disk. By default, if the rate is greather than or equal to 80hz, the flush threshold is 60 frames. Between 40 and 79Hz it is 30, between 20 and 39Hz it is 15, and below 20Hz it is 5 frames.*count* must be between 2 and 64.
- ***AVGF=filter*** is the name of an option filter to be used before calculating the RMS and standard deviation within the bandwidth of the filter.
- ***AVGLEN=n*** sets the interval for reporting RMS and standard deviation statistics. n is the number of samples and defaults to zero (off).
- ***USCALE=n*** specifies the scaling factor to use when displaying data in aqshell, defaults to 1.0.
- ***DACF=filter*** sets the optional IIR filter to use when sending data to analog output.
- ***DACNUM=n*** sets the analog output channel and must be between 1 and 4 or 1 and 8 depending on hardware configuration.
- ***DACDEC=n*** is an optional decimation to use for analog output. Normally every sample is sent to the analog output, setting this value to 2, for instance, would skip every other sample.
- ***SCALE=n*** sets a multiplier to adjust the full scale output of the analog output, the default is 1.0.
- ***MAMP=n*** sets the amplitude in DAC counts for timemarks and defaults to 40. This must be specified and set to 0 for use with DAC480.
- ***DACON=yes*** enables analog output on this channel, this option should be last in the series of analog output parameters to guarantee they are all processed.

- ***FIRFIX=n*** sets a multiplier that is used when FIR filtering down data from a higher frequency to a lower frequency. It defaults to 1.0.

```
[aqcontrol]
k2z=hrv0.hhz:spww
k2z2=hrv1.hhz:spww
```

The section defines the control detectors that are referenced in the [aqlcqs] section. The control detector allow logical combinations of previously defined Murdock-Hutt, Threshold, and Comm detectors. The format is either the actual Comm detector name or in the form : station.[location-].seedname:detector_name. Detector_name may also be "CALON" indicating the current calibration status on that digitizer.

Detectors may be combined using the following logical operators :

- ***&*** Logical AND - Precedence Level 2
- **+** Logical OR - Precedence Level 3 (lowest)
- ***!*** Logical NOT - Precedence Level 1 (highest)
- ******* Logical Exclusive OR - Precedence Level 3 (lowest)
- ***(*** .... ***)*** Parenthesis can be used to modify the precedence ordering.

For instance, "xyz=*(hrv.bhz:spww+hrv.hhz:vspww)&hrv.bhn:spww*" would only trigger event recording if either the first two detectors are on, and the third detector is on. "xyz" is the control detector name. All of the different types combined, so that "*xyz=hrv.bhz:remote+hrv.bhz:calon+comm:5+hrv.bhz:spww"* is valid. The control detector specification can also be following by an optional comma and a Y or N. If Y is specified, such as "xyz=hrv.bhz:spww,y" then a log message will be generated when the control detector goes on or off.

AQCOM Configuration

```
[aqcom]
source=hrv.bhz
wcbuf=y wcseq=y
comlink wccom=y wecom=y cpri=25 epri=27
*
source=hrv.bhn
wcbuf=y wcseq=y
comlink wccom=y wecom=y cpri=25 epri=27
*
source=hrv.bhe
wcbuf=y wcseq=y
comlink wccom=y wecom=y cpri=25 epri=27
*
source=hrv.ehz
webuf=y weseq=y
comlink wccom=y wecom=y cpri=14 epri=15
*
source=hrv.ehn
webuf=y weseq=y
comlink wccom=y wecom=y cpri=14 epri=15
*
source=hrv.ehe
webuf=y weseq=y
comlink wccom=y wecom=y cpri=14 epri=15
*
source=hrv.log
wcseq=y
comlink wccom=y cpri=DEF
*
source=hrv.ace
comlink wccom=y cpri=DEF
*
```

The [aqcom] section is very similar to [aqlcqs] :

- ***SEED*** is not normally required, if not specified the location and seedname is obtained from the source parameter.
- ***SOURCE=station.[location-seedname]*** determines which comserv to get the data from (possibly remapped in the [aqstation] section. If remapping occurs (STATION <> SOURCE in [aqstation]) then aqcom is the one that changes station names.
- ***COMLINK, CPRI, EPRI, WCCOM, WECOM, DETP, CALP, WCSEQ, WESEQ, WCBUF, WEBUF, NODET, NOCAL, NOTIM, GAP, 4KFR***, and ***FLUSH*** are also supported.

Note that aqcom is handling messages instead of pmsgd, so we need a LCQ for the message packets (LOG).

AQBLK Configuration

```
[aqblk]
source=hrv.gps
wcseq=y
comlink wccom=y cpri=8
```

The [aqblk] section is a very small subset of the [aqlcqs] section :

- ***source=station.[location-]seedname*** allows received piped packets with the designated station/location/seedname to be processed.
- ***wcseq=y*** enables combining received blockettes into 4K records and to be written to the blockette log. The blockette log can be written to tape and downloaded using the retrieve program. You can also use (A)vailable to setup, but not enable recording.
- ***Comlink[=modname]*** indicates to allow received blockettes into 512 byte records for transmission by dacommo. wccom and cpri have their usual meanings.

Server Configuration Section

The available parameters depend on the digitizer used. In all digitizers except the K2, the configuration file is ignored until the keyword **[server]**, and parameters are processed until a new keyword is found. If the server is not for digitizer number 0 (K2 servers only), then the configuration keyword is **[servern]** where n is the digitizer number (1 - 9).

Q380 DIGITIZER

The available parameters are :

- ***AUXERR=y*** - write error messages to standard error path.
- ***AUTOSTART=y*** - release reset automatically (so doesn't require serv_start or servtest) after 5 seconds.
- ***GPSON=y*** - enables polling of GPS through SPI connection.
- ***GPSSERIAL=y*** - GPS is connected through serial port.
- ***AUXCAL=y*** - write calibration status messages to standard error path.
- ***INTCAL=y*** - use internal timemarks during calibration.
- ***DATABUF=nn*** - where *nn* is the number of seconds of data to buffer.
- ***ERRBUF=nn*** - where *nn* is the number of error messages to buffer.
- ***SVCQ=nn*** - where *nn* is the number of service queues to allocate.
- ***VCO=nn*** - where *nn* is the initial VCO control value.
- ***SOH=path,low-high*** - where *path* is the serial port for the remote SOH board. This parameter may be repeated two times. *Low* is the aux channel number where the first channel is mapped to. *High* is the highest aux channel to write to.
- ***QPWR=low-high*** - *Low* is the aux channel number where the first A/D on the power board is mapped to. *High* is the highest aux channel to write to.
- ***Q380=slot,low-high,freq[,caltype,alow-ahigh{,options}]*** - Defines a QDP380 board as installed as board "*slot*". Slot must be between 1 (1st board on QVI number 1) and 4 (2nd board on QVI number 2). "*Low-high*" defines the channels where the data from the QDP will be reported. "*freq*" is 0 for 80/40/20/10/1 Hz output, and 1 for 160/20/1Hz output. If a calibrator board is installed then caltype is either "*QAPCAL*" or "*SCAL*". "*Alow-ahigh*" are the aux channel number range where the aux data from that calibrator board will be mapped to. Available "options" for **QAPCAL** are :
  - ***C*** - Common enable output line (STS-2).
  - ***Ennn*** - Calibration enabled for channels *n*, *n*, and *n*. Default is E123.
  - ***R[ddd[/lll[/hhh[/iii]]]]*** - Recentering available, optional override of default, low, high, and increment in ms.

  Available options for **SCAL** are :
  - ***Ann*** - Maximum calibration amplitude (-96 to 0) in dB.
  - ***C*** - Coupling option available.
  - ***Dnn*** - Maximum calibration duration in seconds.
  - ***Ennn*** - Calibration enabled for channels *n*, *n*, and *n*. Default is E123.
  - ***Fn*** - Always use filter *n*.
  - ***Mnn*** - Calibrator output is monitored on channel *nn*
  - ***R[ddd[/lll[/hhh[/iii]]]]*** - Recentering available, optional override of default, low, high, and increment in ms.
- ***GAIN=channel,gain*** - *Channel* is the digitizer channel (1-n) and *gain* is a floating point gain, the default gain for each channel is 1.0.
- ***THROTTLE=value*** - Defaults to 20, lower values (down to 0) will reduce latency (desirable for real-time output such as DSS) but will increase CPU usage.
- ***RETRYSECS=seconds*** - How many seconds to wait before trying to put a digitizer channel back into the map. The default is 0 which means no retry.

Q4120 DIGITIZER

- ***PHYSICAL=n*** - where *n* is the maximum number of physical channels, normally either 4 or 8.
- ***AUXERR=y*** - write error messages to standard error path.
- ***AUTOSTART=y*** - release reset automatically (so doesn't require serv_start or servtest) after 5 seconds.
- ***AUTOEXT=nn*** - where *nn* is the number of seconds of data before switching to external timemarks. Default is zero, which disables this feature. If this is set non-zero and autostart=y then serv_start is not required.
- ***AUTORST=y*** will enable resetting the server and then releasing reset if a missing sync sample is received. Note that autoext=y is required if this option is used.
- ***AUXCAL=y*** - write calibration status messages to standard error path.
- ***DATABUF=nn*** - where *nn* is the number of seconds of data to buffer. The default is 50.
- ***ERRBUF=nn*** - where *nn* is the number of error messages to buffer. The default is 64.
- ***SVCQ=n****n* - where *nn* is the number of service queues to allocate. The default is 20.
- ***SAFETY=nn*** - where *nn* is the number of data buffers to reserve for the server to catch-up with the DSP. The default is 10.
- ***MSGS=nn*** - where *nn* is the number of DSP to host message buffers. The default is 64.
- ***VCO=nn*** - where *nn* is the initial VCO control value. The default is 2048.
- ***DELAY=n*** - where *n* is the number of seconds after the TMS320 starts before it starts any 2105s, defaults to 2 seconds.
- ***SOH=path,low-high*** - where *path* is the serial port for the remote SOH board. There may be up to 2 SOH boards.*Low* is the aux channel number where the first channel is mapped to.*High* is the highest aux channel to write to.
- ***PWR=low*** - where low is the first aux channel, corresponding to the voltage, temperature will be at low+1.
- ***2105=path*** - file name for 2105 object code. Different files may be available for different filters.
- ***TMS320=path*** - file name for TMS320 object code.
- ***FIRFILT=path*** - file name of file containing TMS320 FIR filters.
- ***MAPFILE=path*** - *path* to write optional channel/memory map.
- ***GPS=magnavox|motorola|mot3|gps1|gps2 -Magnavox*** indicates that there is a Magnavox GPS Engine attached to /x5. ***Motorola*** indicates that there is a Motorola GPS Engine that supports NMEA format attached to /x5.***Mot3*** indicates that there is Motorola GPS Engine that does not support NMEA format attached to /x5.***GPS1*** indicates that there is an external GPS1 connected to some serial port (controlled by clock). ***GPS2*** indicates that there is an external GPS2 connected to some serial port.
- ***GPSSTAT=daily|hourly|10min*** - Sets automatic status reporting interval.
- ***GPSAUTO=y*** - Enable reporting GPS status when changing clock states.
- ***GPSOPTS=[reset],[defon]*** - If *reset* is specified, the GPS Engine will be reset whenever the server starts, else the GPS Engine is only reset if the RF power was off when the server starts. If*defon* is specified it indicates that the 4120 is jumpered so that the default state is RF power on.
- ***SCAL=side,port,alow-ahigh{,options}*** - where side is 1 for channels 1-4 and 2 for channels 5-8. port is the serial port used to talk with the supercal. "Alow-ahigh" are the aux channel number range where the aux data from that calibrator board will be mapped to. Available options for SCAL are :
  - Ann Maximum calibration amplitude (-960 to 0)
  - C Coupling option available
  - Dnn Maximum calibration duration
  - Ennn Calibration enabled for channels n, n, and n. (All = 123)
  - Fn Always use filter n.
  - Mnn Calibrator output is monitored on channel nn
  - R[ddd[/lll[/hhh[/iii]]]] Recentering available, optional override of default, low, high, and increment
- ***QCAL=side,alow-ahigh[,options]*** - where side is 1 for channels 1-4 and 2 for channels 5-8. "Alow-ahigh" are the aux channel number range where the aux data from that calibrator will be mapped to. Available options for QCAL are :
  - Dnn Maximum calibration duration
  - Mnn Calibrator output is monitored on channel nn
  - R[ddd[/lll[/hhh[/iii]]]] Recentering available, optional override of default, low, high, and increment
  - F Signal is on channels 1-3 instead of 2-4 (default).
- ***CHECKGAP=y*** - Server checks for data gaps (this is always done in Aqsample).
- ***NOXFER=y*** - The DSP section transfers messages only to server, no data.
- ***BURSTSIZE=n*** - where n is the number of bytes to be transferred from shared DSP/System CPU memory per burst, in bytes. Will be made into a multiple of 4 bytes. Default is 512.
- ***BURSTINT=n*** - where n is the interval between bursts, the interval being equal to n * 500µs for a tickrate of 32000 and n * 450us for a tickrate of 20000. Default is 16.
- ***INBUFSIZE=n*** - where n is the size of the 2105 interrupt input buffers in the TMS320. Default is 125.

- ***AUTOGAIN=n,counts*** - Set the gain on channel *n* so that the output value of stage 1 is *counts* when the calibration voltage is applied. This is not supported on current Analog input boards. If this option is used, the GAIN option is ignored.
- ***GAIN=n,value*** - Set manual gain on channel *n* to floating point *value*.
- ***TICKRATE=n*** - Sets sampling rate, defaults to 20000. 32000 is the other supported rate.
- ***STAGE=1,frequency,delay[,>channels[*mul]][,>>channels]***
- ***STAGE=n,frequency,filter_name,<source[,>channels[*mul]][,>>channels]*** - where *n* is the TMS320 stage number, *delay* is the 2105 filter delay in seconds, *filter_name* is the name of the TMS320 filter, and *source* is the stage from which this stage derives it's input data. Each stage can have two outputs, one (defined by the >) that goes to the server as a fixed point value, and the other which is passed through to the input of other stages as a floating point value. *channels* are the channels to output or pass through and *mul* is an optional scaling multiplier to bring the fixed point output into the desired range. For instance "*>1234*0.25 >>5678*" will output to the server channels 1, 2, 3, and 4 with the value multiplied by 0.25 and allow channels 5, 6, 7, 8 and to be passed through to other stages. You cannot use a stage as a source for a channel unless that channel is defined as a pass through on another stage. You cannot pass data through unless there is another stage to receive it. The maximum number of available output frequencies is "**MAXFREQ**" (defined in servdata.i) and the maximum number of TMS320 filter stages is "**MAXSTAGE**" (defined in server.i). Output is not allowed for frequencies above 1Khz. If a stage defines a channel/frequency combination already defined, it is assumed to be a detector filter, and its channels are mapped into logical channels to avoid conflict. The server "get_logical" call can be used to determine the logical channel based on the physical channel number and filter name.

STAGE 1 DELAYS

- AD32 = 0.006234
- AD32M = 0.00163
- AD20 = 0.015975
- AD20M = 0.002663

DSP Data Flow

Raw A/D data and a quantagrator bit are input to an AD2105 fixed point DSP (one per channel) at 20Khz or 32Khz. The 2105 DSP does the quantagration processing and filters the result down to a 1Khz or 2Khz data stream, with an absolute maximum range of ±26 bits. Data from all of the 2105's is multiplexed into a single TMS320 floating point DSP. During startup the 320 calculates both a fixed point channel offset and floating point channel gain factor (for automatic calibration, or manual gain setting in config file). Before the 2105 data is processed the offset is subtracted, converted to floating point, and multiplied by the channel gain factor. Data is passed from filter stage to filter stage in floating point format. Data to output to the server is multiplied by a stage multiplier (*mul, or 1.0 if not specified) and then converted to fixed point.

HRDCU/LRDCU DIGITIZERS

The available parameters are :

- ***AUXERR=y*** - write error messages to standard error path.
- ***AUTOSTART=y*** - release reset automatically (so doesn't require serv_start or servtest) after 5 seconds.
- ***VERBOSITY=y*** - Generates messages for HRDCU/LRDCU initialization.
- ***GPSSERIAL=y*** - GPS clock is connected to a serial port.
- ***DATABUF=nn*** - where *nn* is the number of seconds of data to buffer.
- ***ERRBUF=nn*** - where *nn* is the number of error messages to buffer.
- ***SVCQ=nn*** - where *nn* is the number of service queues to allocate.
- ***VCO=nn*** - where *nn* is the initial VCO control value.
- ***RETRIES=nn*** - where *n*n is the number of times the server will re-initialize the digitizers if a LRDCU starts without finding the external clock from the HRDCU.
- ***SOH=path,low-high*** - where *path* is the serial port for the remote SOH board. This parameter may be repeated two times. *Low* is the aux channel number where the first channel is mapped to. *High* is the highest aux channel to write to.
- ***HRDCU=path,delay*** - where *path* is the serial port for the HRDCU. *delay* is the HRDCU filter delay in seconds. The HRDCU must be defined before any LRDCUs. The HRDCU is assigned channels physical channels 1 through 4 (4 is the calibration monitor channel).
- ***LRDCU=path,delay,freq,channels*** - **where *path* is the serial port for a LRDCU. *delay* is the LRDCU filter delay in seconds. *freq* is the output sample rate in Hz.** *Channels* are the number of normally active channels (1-6). The server always adds one for the calibration channel.

Example :

```
HRDCU=/X1,0.045
LRDCU=/X2,0.03,100,3
LRDCU=/X3,4.537,1,6
```

Defines a HRDCU as channels 1 through 4 on port "/x1" with a filter delay of 45 milliseconds, a LRDCU as channels 5 through 8 on port "/x2" at 100Hz with a filter delay of 30 milliseconds, and another LRDCU as channels 9 through 15 on port "/x3" at 1Hz with a filter delay of 4.537 seconds.

K2 DIGITIZER

The available parameters are :

- ***AUXERR=y*** - write error messages to standard error path.
- ***AUTOSTART=y*** - release reset automatically (so doesn't require servtest) after 5 seconds.
- ***DATABUF=nn*** - where *nn* is the number of seconds of data to buffer. Default is 60.
- ***ERRBUF=nn*** - where *nn* is the number of error messages to buffer. Default is 20.
- ***SVCQ=nn*** - where *nn* is the number of service queues to allocate. Default is 20.
- ***VERBOSITY=y*** - enables writing of digitizer resetting messages as well as packet resend requests and resent packet messages to be written to the standard error path.
- ***PORT=n*** - sets the serial port path for the serial link to the K2 or the IP Port for a network connection if "IPADDR" is specified, in which case must be between 5000 and 65535.
- ***IPADDR=address*** sets the server to communicate via TCP/IP rather than a serial port (using the sernet program on a SUN).
- ***NETDLY=seconds*** sets the number of seconds between trying to connect the sernet program when using TCP/IP.
- ***RATE=frequency*** - 100Hz is the only rate supported by the K2 software at this time and is the default.
- ***CHANNELS=n*** - where n is the number of channels to be received from the K2, the default is 3.
- ***TIMEOUT=n*** - where n is the number of seconds to initially wait when waiting for an acknowledgement of a initialization packet sent to the K2, the default is 3. If no response is received within this time from the K2, the packet is sent again and the timeout is doubled, not to exceed*MAXTIMEOUT*.
- ***MAXTIMEOUT=n*** - where n is the maximum timeout for the above description, the default is 100 seconds.
- ***INTERVAL=n*** - where n is the number of seconds to wait for reception of a resent packet, between requests. The default is 5 seconds.
- ***LAG=n*** - where n is the maximum number of seconds between the earliest second of data still being waited for from the K2 to the highest second received before giving up. This must be below DATABUF - 2. The default is 30.
- ***GAP=n*** - where n is the maximum gap between the last packet received from the K2 and a new, higher non-contiguous packet. If there is more than n number of seconds between the two then no attempt is made to fill in the gap and the output data from the digitizer will jump. If below this value, then the server will request the missing packets. If it is unable to recover the missing packet before hitting the LAG parameter, the data that is not available will be set to zero. The default is 5.
- ***RESTART=n*** - where n is the number of seconds used as a timeout before giving up on packet transmission from the K2. If there are no valid packets from the K2 within this time, the K2 digitizer will be re-initialized. The default is 120.
- ***OFLOW=y*** will enable output flow control of packets to the K2. This would normally be used with modems.
- ***ONEWAY=y*** will setup the server to accept data from a K2 without requiring data to be sent to the K2. Note that retries are not available, missing data will be replaced with zeroes.
- ***QUALGOOD=n*** - where n is the clock quality to be reported when the K2 reports it is locked to GPS time, the default is 4.
- ***QUALBAD=n*** - where n is the clock quality to be reported when the K2 reports it is not locked to GPS time. the default is 2.
- ***STATUS=daily|hourly|10min*** - determines how often to get a status dump from the K2, the default is daily.
- ***RETRIES=n*** - where *n* is the maximum times the server will request a resend of a packet, the default is 1.
- ***K2BUG1=n*** - where *n* is the number of seconds between sending a "\\\" to try and get the K2's attention.
- ***K2BUG2=n*** - where *n* is the number of times to send the "\\\". The default is 2.
- ***K2BUG3=n*** - where *n* is the number of seconds to wait after resetting the K2 before trying to get it's attention again. The default is 120.
- ***K2BUG4=n*** - where *n* is the number of seconds between sending a "\\\" after resetting, this is repeated until acknowledged. The default is 5.

Clock Configuration Section

The configuration file is ignored until the keyword "**[clock]**" and parameters are processed until another keyword is found. The parameters available are :

- ***FORMAT=string_format*** - where string format can be :
  - ***QTS*** - Quanterra Time & Space format for GPS1 (Magnavox) clocks.
  - ***QTS2*** - Quanterra Time & Space format for GPS2 (Motorola) clocks.
  - ***QTS3*** - Quanterra Time & Space format for GPS3 (Mot3) clocks.
  - ***UA31S*** - Meinberg UA31S DCF77 Clock.
  - ***OMDC*** - Kinemetrics OMDC, 432DC, or 77DC.
  - ***OS9*** - operation using the OS9 system clock.
- ***YEAR=yrsrc*** - a parameter only used with an OMDC format clock, which doesn't know what year it is. yrsrc can either be the current year (such as 1994), or OS9 to use the year from the OS9 system clock.
- ***VERBOSITY=n*** - where n is 0 for no messages, 1 for errors and offset setting messages, and 2 for messages every timemark.
- ***PHASELOCK=y*** - enables phase locked frequency control.
- ***M0DULUS=n*** - where n is the fraction of a second to which digitization times are kphase locked. Default is 1/80 second, or 0.0125.
- ***PATH=port*** - sets the serial port from which the timestring is to be read. "SERVER" is used if the Quanterra GPS clock is connected to the system via the SPI port.
- ***MODEL=n*** where n specifies the clock model that will be reported in the timing logs. The following models are defined :
  0 = unknown clock.
  1 = OS9 internal clock.
  2 = Kinemetrics Goes clock.
  3 = Kinemetrics Omega clock.
  4 = Kinemetrics DCF clock.
  5 = Meinburg UA31S clock.
  6 = Quanterra GPS1 (or Magnavox) with QTS format.
  7 = Quanterra GPS1 with Goes emulation.
  8 = Quanterra GPS1 with UA31S emulation.
  9 = Quanterra GPS2 (or Motorola) with QTS2 format.
  10 = Kinemetrics Altus K2 GPS clock.
  11 = Quanterra GPS3 (or Mot3) with QTS3 format.
- ***LOCKLEVEL=n*** - allows adjusting the minimum reception quality for phase lock, at the default of 1, the highest quality is required.
- ***DELAYMIN=minutes*** - allows setting the delay in minutes after a clock loses the accurate time lock before changing quality from 0 to 2. This is designed to allow momentary losses of satellite to not interfere with phase lock operation while it operates on it's internal clock. Default is 60 minutes.
- ***PLLDELAYMIN=minutes*** - allows setting the delay in minutes after external lock is lost and when the PLL stops tracking.
- ***TIMEOUT=seconds*** - Sets the timeout, in seconds, after which loss of the clock string will generate an error. If the clock string is not received for this timeout, the quality level will be set to the following parameter until the clock string returns. A value of zero (default) disables this feature.
- ***LOSSLEVEL=n*** - Sets the quality level if the clock string is not received.
- ***LOSSINTERVAL=seconds*** - How often an error message will be generated while the clock string is missing, default is 3600 (1 hour).
- ***ZONEOFFSET=seconds*** - Sets an offset (plus or minus) from UTC for those people who wish to time data using local time rather than UTC.
- ***SYSTOLERANCE=seconds*** - Sets how close the difference between sucessive minute strings and the OS9 clock must be (plus or minus) when waiting for clock string acceptance, default is 10 seconds. If the difference is greater than this number of seconds, the acceptance time will be recalculated using the new difference.
- ***MINSYSDIFF=seconds*** - Determines how close the time string has to be to the OS9 clock to be accepted on the first time. If there is greater than this many seconds, it will take 2 or more minutes to accept the clock string as valid. Default is 180 seconds.
- ***SYSRATIO=n*** - Sets the ratio of time between how far apart the clock string and OS9 clocks to the number of minutes that will elapse with a consistant difference in those two clocks. The number of minutes is calculated as truncate((<clock difference> + (<sysratio> - <minsysdiff>) / <sysratio>) + 1. Initial clock differences less than MINSYSDIFF will result

as the first clock string being accepted, larger differences require more minutes for acceptance. The number of minutes for acceptance will be clipped at *MAXWAIT* minutes. Default ratio is 1000.

- ***MAXWAIT=minutes*** - Sets maximum time for clock acceptance. Default is 180 minutes (3 hours).
- ***PFRAC=value*** - Overrides the default proportional feedback factor for the Proportional-Integral phaselock system. The defaults for this value range from 6.0 for the 4120 to 20.0 for the HRDCU.
- ***IFRAC=value*** - Overrides the default integral feedback factor for the Proportional-Integral phaselock system. The defaults for this value range from 0.00005 for the 4120 to 0.0004 for the HRDCU.
- ***DELTAF=value*** - Overrides the default frequency to count ratio for the VCO. This represents the change, in Hz, produced by one count difference in the VCO control value. The defaults for this value range from 0.0056 for the 4120 to 3.9 for the HRDCU.

Sysmon

Sysmon provides a simple interface for checking system operation and doing tape changes where there is no terminal available. The interface consists of two bi-color LEDs plus a pushbutton. This interface is currently only available for Q4120 systems. Sysmon is also used to generate alerts under certain conditions relating to tapes and acquisition status as well as setting the OS9 system clock to agree with actual time.

4120 User Interface

One LED and the pushbutton are used for changing tapes. The LED has the following states :

- ***OFF*** - Tape running normally, no errors. A tape change can be started by pushing the pushbutton.
- ***FLASHING RED*** - Error, such as full tape or write protected tape. The tape may or may not be in the drive, in either case, a tape change can be started by pushing the pushbutton.
- ***STEADY RED*** - Pushbutton has been pushed, you may release the button.
- ***FLASHING RED/GREEN*** - Sysmon has requested a tape change be started. Tapeserv will flush logs to tape before writing EOF and ejecting the tape, so this could take quite a while.
- ***STEADY GREEN*** - Tape should now be ejected, you can remove the tape and put a new one in. After checking that the tape can be written the LED will change to either *OFF* indicating the change has been sucessful, or *FLASHING RED* indicating that there is something wrong with the tape. If there is a problem with the tape, push the button again, remove the tape when you get *GREEN*, and try another tape.

The other LED shows general acquisition operation :

- ***OFF*** - Acquisition not operating.
- ***FLASHING RED*** - Acquisition operating, with poor clock (quality is 1 or 2).
- ***FLASHING RED/GREEN*** - Acquisition operating with degrading clock (quality is 0).
- ***FLASHING GREEN*** - Acquisition operating with locked clock (quality is 3, 4, or 5).

All Systems

Sysmon checks tape and acquisition operation every 5 seconds. In addition to the LED functions available on the 4120 only, it can also generate alerts based on current operating conditions. See the configuration section on how to set up which alerts are supported.

Sysmon is also used to set the OS9 clock to keep it fairly close to actual time. This is useful when acquisition restarts in the case where a clock (such as GPS) is not able to immediately lock on. The data is then timetagged using the OS9 clock until actual time is available so it is nice if it is fairly close.

Sysmon periodically analyzes the clock from the stations listed in the "clocks" configuration entry to try to determine which station time to use. The interval defaults to 60 minutes, but can be changed in the configuration section. At the start of the interval it takes 30 samples from each station, spaced 5 seconds apart. The samples are calculated as the difference between the acquisition time for that station and the OS9 clock time. At the end of the 30 samples, the average difference between the two is calculated. Another 30 samples are then taken, this time the deviation between the current station-OS9 time and the average is taken, squared, and added to a total "deviation" counter for that station. After these 30 samples are taken, any stations that had poor or degrading clock quality during the 60 total samples, or whose "deviation" exceeds the threshold are excluded. Of those stations not excluded, the highest time (lowest latency) station is used. If the difference between that station and the OS9 clock exceeds the tolerance value, the OS9 clock is reset. Note this function (with fewer quality controls) was in Aqsample in Ultra-Shear, but due to multiple data sources is now handled by Sysmon.

Chart Configuration Section

The Chart server is a background process accepts commands through a pipe to create plots on a HPGL printer. The configuration file is ignored until the keyword **[chart]** and parameters are processed until another keyword is found.

pipename=path where path is the name of the pipe that chart will create and other programs will write their commands to. No default. Can also be specified in the [aqglobal] section as chartname=path.

pipesize=bytes The pipe must be large enough to hold the worst case number of commands that may be sent at the same time, commands require approximately 80 bytes each. Default is 256 bytes. Can also be specified in the [aqglobal] section as chartsize=bytes.

outpath=path This is the port name for the printer/plotter.

format=HPGL The format to use. The default (and currently the only supported format) is HPGL.

size=letter | legal | A4 The paper size. Default is legal.

dpi=resolution Specifies the resolution (in dots per inch) you want to represent, defaults to 300. Higher resolutions produce more data to the printer and may overrun internal memory of HP printers due to bad firmware design in their HPGL translators.

maxpoints=samples This is the maximum number of samples per trace. This must be the worst case of any of the charts, calculated as HOURS * 60 * SAMPLE_RATE / TRACES. Default is 10000, absolute maximum is 100000. Note that system memory equal to MAXPOINTS * 4 bytes is obtained by Chart on startup.

pre_delay=seconds This command allows delaying the specified number of seconds after sending "pre_string" and before issuing any HPGL commands. Default is 0. 20 was a good value for the LJ2, and 15 for the LJ4.

post_delay=seconds This command allows delaying the specified number of seconds after sending "post_string" and before processing the next chart. Default is 0. 45 was a good value for the LJ2, and 30 for the LJ4.

pre_string=string This is typically used on HPGL emulation's on printers to setup the printer before accepting HPGL commands. Some users may also wish to have their printers left in normal mode between charts. For instance, to change from HPCL and perform other setups required, the following commands can be used on the LaserJet Series 2 with a Pacific Data Products Emulation Cartridge : **^[&l32259.1057J** while on a LaserJet 4 additional commands are required to also set landscape mode, set margins, and for non-letter paper, set the multipurpose tray as the input. For legal on the LJ4 : **^[E^[&l4H^[&l3A^[&l1O^[9^[&l0E^[%0B**. For letter on the LJ4 : **^E^[&l2A^[&l1O^[9^[&l0E^[%0B**. And for A4 on the LJ4 : **^[E^[&l4H^[&l26A^[&l1O^[9^[&l0E^[%0B**. Note that the ^ symbol is used indicate a control character follows, **^[** is Escape. **l** is lower case letter L, **1** is numeral one, **O** is uppercase letter O, and **0** is numeral zero.

post_string=string This is typically used on HPGL emulation's to put the printer back into HPCL mode. For the LJ2 with PDP cartridge : **^[&l1057.32259J** and for the LJ4 : **^[%0A^[E**.

text_pen=string HPGL was designed initially for pen plotters, so you could select between different pens as the plot was drawn. Chart allows setting up to 5 different pens depending on the item to be drawn. For a real plotter, the commands would be simply SPn where n specifies the pen to use. For emulation's there are various methods used to select the width of the pseudo-pen. On the LJ2 with PDP cartridge a command of the form **SP1,w** sometimes works, other times **SP1;PTw** works, depending on which pen is being specified, where w is the width of the pen. For the text pen, **SP1,3** is recommended on the LJ2 and **SP1;PW0.2** for the LJ4. The text pen is used for writing all text. Default is **SP1**.

minute_pen=string This pen is used to draw the minute lines on the chart. **SP1PT1** is recommended on the LJ2, and **SP1;PW0.05** on the LJ4. Default is **SP1**.

trace_pen=string This pen is used to draw the wave form traces when neither an event or calibration is in progress. **SP1,2** is recommended on the LJ2, and **SP1;PW0.1** on the LJ4. Default is **SP1**.

event_pen=string This pen is used to draw the wave form traces during events. **SP1,4** is recommended on the LJ2, **SP1;PW0.3** on the LJ4. Default is **SP1**.

cal_pen=string This pen is used to draw the wave form traces during calibrations. **SP1,3** is recommended on the LJ2, **SP1;PW0.2** on the LJ4. Default is **SP1**.

top=mm Sets the top margin in millimeters. The default of 0 is recommended for the LJ2, 20 for the LJ4.

left=mm Sets the left margin. The default of 0 is recommended for the LJ2, 2 for the LJ4.

right=mm Sets the right margin. The default value of 0 is recommended for the LJ2, -2 for the LJ4.

bottom=mm Sets the bottom margin. The default value of 0 is recommended for the LJ2, -2 for the LJ4.

Command Lines

Most programs send error/status messages to the standard error device. This device is normally redirected into a pipe that the "pmsgd" program reads to build the message log. This pipe name must have the form of "/pipe/log[.<station>].<program>". <program> is the name of the program generating the message. Some programs only operate on data for one station, such as servers, dacommo (except for station=any), clock, and lockrep. These programs should set the <station> portion of the pipe name to the station they operate on. Most programs, such as aqsample, embed the station the message refers to (if specific to a station) in the message itself and so the optional station portion of the pipe name is not included. The "aqgo" file for the example configuration is described below :

```
*
* kill remnant processes
*
echo "...purging remnant processes"
aqstop
```

Aqstop is a procedure file that stops acquisition (both aqsample and aqcom), shuts down all other acquisition processes, and deletes all the data modules created by the acquisition programs.

```
*
* Setting priorities of ifman and sockman
*
setprio ifman 6500
setprio sockman 6500
```

Setprio is used in this case to increase the priority of the networking processes on a heavily loaded system. The first command line parameter is the name of the process, the second parameter is the new process priority.

```
echo ...starting piped message demon with output to /nil
shell "pmsgd -a=5000 >/nil <>>/nil&"
sleep -s 5
```

This starts the piped message daemon and creating a 5000 entry Aqlog message log if it doesn't already exist. In some installations the standard output is redirected to a printer port for logging. Pmsgd has the following command line options available :

- ***-c=path*** specifies the name of the configuration file, defaults to /r0/aqcfg.
- ***-a=n*** specifies how many entries to put into the Aqlog file if it doesn't already exist. Defaults to 1000.
- ***-m=n*** specifies how many entries to put into the msglog file used by aqshell if it doesn't already exist. Defaults to 100.
- ***-t=sec*** sets the timeout for the standard output port in seconds.

```
echo ...starting fingerd
shell "fingerd <>/nil >>-/pipe/log.fingerd &"
sleep -s 2
```

The finger daemon process allows a limited set of commands to be executed over a network connection using the "finger" command. Fingerd has no command line parameters.

```
echo ...loading IIR filters
loadiir /r0/iirfilters <>/nil >>-/pipe/log.loadiir
```

Loadiir reads IIR filters from the file specified by the first command line parameter and places them in the data module "IIRDATA".

```
echo ...starting tape server
shell "tapeserv <>/nil >>-/pipe/log.tape &"
sleep -s 1
```

Tapeserv has as its only command line option the name of the configuration file, which defaults to /r0/aqcfg.

```
echo ...starting continuous data buffer server with profiler
shell "bufserv -q=C ^2000 >>-/pipe/log.BufC <>/nil&"
sleep -s 3
bufwait bufsvcq_C 60
*
echo ...starting event data buffer server with profiler
shell "bufserv -q=E ^2000 >>-/pipe/log.BufE <>/nil&"
sleep -s 3
bufwait bufsvcq_E 60
```

This set of commands starts up the two buffer servers, the first for continuous data, the second for event data. Bufserv has as its command line options :

- ***-c=file*** specifies the name of the configuration file, defaults to /r0/aqcfg.
- ***-q=x*** where x is C for continuous and E is for event.

Bufwait is run after starting a buffer server to wait for it to initialize and has as it's first parameter the name of the server module and the second parameter is the maximum number of seconds to wait.

```
echo ...starting K2 servers"
servk2 <>/nil >>-/pipe/log.hrv0.servk2 ^1500 &
sleep -s 2
servk2 1 <>/nil >>-/pipe/log.hrv1.servk2 ^1500 &
sleep -s 2
```

The server number is zero if not specified as the first command line parameter. Following the optional server number can be the name of the configuration file, which defaults to /r0/aqcfg. Only the K2 server has this server number parameter, all other servers (which use the name "server") have only the optional configuration file override option.

```
echo "...starting remote command manager"
rcman -v=2 >>-/pipe/log.rcman <>/nil &
sleep -s 2
```

Rcman has the following command line options :

- ***-v=n*** sets the verbosity level, default is zero.
- ***-l*** will lockout calibrations during event detections.

```
echo "...starting comlink process for remote Q680"
dacommo -v=1 >>-/pipe/log.hrv.dacommo <>/nil &
sleep -s 5
echo "...starting comlink processes for K2s"
dacommo -v=1 -m=commo0 >>-/pipe/log.hrv0.dacommo <>/nil &
sleep -s 5
dacommo -v=1 -m=commo1 >>-/pipe/log.hrv1.dacommo <>/nil &
sleep -s 5
echo "...starting comlink process for Log"
dacommo -v=1 -m=logcommo >>-/pipe/log.log.logcommo <>/nil &
sleep -s 5
echo "...starting comlink process for Any "
dacommo -v=1 -m=anycom >>-/pipe/log.any.commo <>/nil &
sleep -s 5
```

Dacommo has the following command line options :

- ***-c=file*** specifies the configuration file, default is /r0/aqcfg.
- ***-v=n*** specifies the verbosity level.
- ***-m=modname*** specifies the data module name, default is "dacombuf".

```
echo "...starting comserv"
comserv hrv </nil >-/pipe/log.hrv.comserv >>-/pipe/log.hrv.comerr ^3000 &
sleep -s 2
```

The command line parameter for comserv is the name of the remote station.

```
echo "...starting aqdacserv"
xmode /t1 noupc nobsb bsl noecho nolf null=0 nopause pag=0 bsp=00 del=00 eor=00
xmode /t1 eof=00 reprint=00 dup=00 psc=00 abort=00 quit=00 bse=00 bell=00
xmode /t1 type=00 baud=9600 xon=00 xoff=00 tabc=00 tabs=0
aqdacserv -v=1 <>/nil >>-/pipe/log.aqdacserv ^3500 &
sleep -s 1
```

Xmode sets up the serial port (/t1 in this case) for binary operation at 9600 baud. Aqdacserv has the command line options ;

- ***-v=n*** sets the verbosity level, defaults to 0.
- ***-c=file*** sets the configuration file, defaults to /r0/aqcfg.

```
echo "...starting sendmail"
sendmail <>>/nil &
sleep -s 1
```

Sendmail has as it's only command line option the configuration file name, which defaults to /r0/aqcfg.

```
echo "...starting alert"
alert <>/nil >>-/pipe/log.alert &
sleep -s 5
```

Alert has as it's only command line option the configuration file name, which defaults to /r0/aqcfg.

```
echo "...starting colert"
colert <>/nil >>-/pipe/log.colert &
sleep -s 2
```

Colert has as it's only command line option the configuration file name, which defaults to /r0/aqcfg.

```
echo ...waiting for clock external lock
clockwait <>>/nil >-/pipe/log.clockwait
```

Clockwait has the following command line options :

- ***-x=n*** sets the digitizer number to wait for, default is zero. If used, this must the first command line option.
- ***seconds*** sets the maximum number of seconds to wait for the clock quality to reach 3 or higher, defaults to 300.

```
echo ...starting sysmon
sysmon <>/nil >>-/pipe/log.sysmon &
```

Sysmon has as it's only command line option the configuration file name, which defaults to /r0/aqcfg.

380/4120/730/HRDCU Server Support

Servers for other digitzers than the K2 require the "clock" program to correlate digitizer time with an external source of timestrings and timemarks, such as from a GPS clock. The startup sequence is also modified, a typical sequence is :

```
echo ...starting server and waiting for intialization
server </nil >/nil >>-/pipe/log.xxxx.server ^2500 &
sleep -s 1
servwait
```

xxxx is replaced with the name of the station that this server will be providing data from. Servwait delays while server is initializing the data module. Servwait will be converted to run with digitizers other than 0 in the future.

```
echo ...starting serv_start
serv_start 30 75 </nil >/nil >>-/pipe/log.xxxx.servstart &
```

Serv_start has as it's first parameter the delay in seconds before server is started. The optional second parameter specifies the time in seconds after starting before switching to external timemarks. For the 4120 server with the autostart and autoext configuration parameters set, serv_start is not required.

```
xmode /t2 noupc nobsb bsl noecho nolf null=0 nopause pag=0 bsp=00 del=00 eor=00
xmode /t2 eof=00 reprint=00 dup=00 psc=00 abort=00 quit=00 bse=00 bell=00
xmode /t2 type=00 baud=9600 xon=00 xoff=00 tabc=00 tabs=0
clock </nil >/nil >>-/pipe/log.clock ^2200 &
sleep -s 5
```

This sets up the serial port for an external clock for binary operation at 9600 baud. Some configurations use an alternate method to communicate with the clock and do not require serial port setup. Clock has as it's only command line option the name of the configuration file, which defaults to /r0/aqcfg.

```
echo ...starting phase lock report generator
lockrep -l <>>/nil >-/pipe/log.lr &
echo ...waiting for clock external lock
clockwait <>>/nil >-/pipe/log.clockwait
```

Lockrep with the -l option stays in a loop and periodically reports the progress of the phase lock system in the clock program. Lockrep has the following command line options :

- ***-f=secs*** where *secs* is the number of seconds to set "lockable seconds in hold", whatever that is.
- ***-l*** will cause lockrep to stay in a loop, and it will report status whenever there is a change.
- ***-p*** will enable the phase lock function.
- ***-x*** will disable the phase lock function
- ***-d=mins*** will set the external delay to *mins* minutes.
- ***-g=opt*** will send the *opt* string to the gps clock. The most common string is "2" which causes a status dump.
- ***-n=ms*** will set the server timemark noise filter to *ms* milliseconds.
- ***-j=mins*** will set the server timemark jump value to *min* minutes, zero to disable the jump filter.

Clockwait is used as before, to wait for external time lock (quality 3 or higher).

Procmon was developed for use on the 730 to reset the system (short watchdog) should any critical programs fail. Procmon is also needed on new 4120 systems to keep the watchdog reset if no dacommo is running.

```
procmon <>/nil >>-/pipe/log.procmon &
```

The following configuration options are available for procmon :

- ***names=<program name>{,<program name>}*** will add the program to the list of processes to monitor. If there is more than one copy of a program running (such as multiple dacommos) then all will be added to the list. To just reset a 4120 watchdog without doing the process checking, this entry is not used.
- ***timeout=nn*** where *nn* is the interval in minutes for procmon to check the named processes.The default is 15 minutes.
- ***initial=nn*** where *nn* is the time in minutes after procmon starts before it expects all named processes to have started.

- ***verbose=y*** will enable generating messages at "interval" minutes listing each process being monitored.
- ***dogreset=y*** will reset the watchdog at "interval" minutes, this is used on 4120 systems without any dacommo running.

Other Background processes

The chart program has as it's command line option the name of the configuration file, which defaults to /r0/aqcfg.

Tmon is normally started before any acquisition files and is not stopped when acquisition stops. It provides dial-in control and data retrieval capabilities, along with the optional fax dispatching and scheduling. The command format is :

```
tmon port logfile [configuration-file]
```

Acquisition and Retrieval

After the background processes are started the next step is normally to start the "AqShell" program with the automatic startup option to begin acquisition. Aqshell has the following command line options :

- ***-c=file*** sets the configuration file, defaults to /r0/aqcfg. This is also passed onto the aqsample and/or aqcom command lines.
- ***-r*** will cause aqsample, aqcom, and to start automatically. Aqsample will be started if there is a [aqlcqs] section in the configuration file, aqcom will be started if there is a [aqcom] section, and aqblk will be started if there is a [aqblk] section.
- ***-x=mem*** passes onto the aqsample command line.
- ***-v=n*** passes onto the aqsample and/or aqcom command lines.
- ***-l*** enables display of logical channels from the digitizer in addition to the physical channels.

Aqshell can start aqsample, aqcom, and aqblk, or those programs can be started manually without using aqshell. Aqsample has the command line options :

- ***-c=file*** sets the configuration file, defaults to /r0/aqcfg.
- ***-v=n*** sets the verbosity level, defaults to the "verbosity" parameter in the [aqglobal] section.
- ***-x=mem*** sets the size of the aqcommon module, else it is calculated automatically.

Aqcom and Aqblk have the following command line options :

- ***-c=file*** sets the configuration file, defaults to /r0/aqcfg.
- ***-v=n*** sets the verbosity level, defaults to the "verbosity" parameter in the [aqglobal] section.

Retrieve can be started from Aqshell or started manually. Retrieve is also commonly started when a user logs in. Retrieve has the following command line parameters :

- ***-c=file*** sets the configuration file, defaults to /r0/aqcfg.
- ***-q=x*** where *x* is C for continuous data or E for event data. Once inside retrieve, buffer servers can be changed.
- ***-nl*** will prevent user commands from being added to the user log file.
- ***-nt*** will prevent retrieve from setting a 60 second timeout on the user's terminal.
- ***-f=path*** makes retrieve read commands from the*path* and disables STP transfers (used by fingerd).
- ***-t=path*** makes retrieve read commands from the*path* (used by tentacle).
- ***-r=nn*** sets the maximum number of 4K seed records that can be downloaded, the default is 2000 (8MB).
- ***-m=nn*** sets the number of user log entries (if the log does not already exist), the default is 1000.
- ***-ol*** will override the limit on the number of retrieve programs running at one time for this invocation of retrieve only.

Control and Status Commands

Dacommstat displays information about a comlink. The command line format is "dacommstat [-f=prio] [-v] [datamodulename]". The datamodulename defaults to "dacombuf" if not specified. The F option allows flushing a queue based on priority. The V parameter will show additional detail when queues are segmented by seedname.

Servmsg is used to send a message to a Q380 or Q4120 digitizer. The first command line parameter is the channel and the second parameter is the message to send.

Alertstat displays the contents of the Fax and Email queues.

Setcorr changes the clock correction value for aqdacserv to the command line parameter. If no command line parameter is used, the current value will be displayed.

Detset can be used to edit detectors running in aqsample. Aqshell forks this process for the "Edit detection parameters" menu item.

Tapestat reports on the status of each of the tape drives if tapeserv is running. This command was created to allow tape manipulation over the com link. If no command line perameters are present, it simply reports the status. The tape it is currently writing on is indicated as the "active" drive. If command line parameters are used, there must be two of them. The first is the command letter, and the second is the tape path name (such as mt0). The following commads are allowed :

- ***C*** - change to the tape specified. If this is the active tape, then this indicates a tape change operation.
- ***E*** - erase volume header, this is normally only used on an inactive tape but will work on an active tape, the effect is to throw away any data on the tape and start over, no log flush is done before erasing the volume header.
- ***F*** - flush logs to tape, this can only be done on the active tape.
- ***H*** - hide tape from tapeserv, this can only be done on an inactive tape. This closes the path to the tape, so that you can use other tape utilties on that drive.
- ***I*** - indivisible tape change, this can only be done on the active tape.
- ***L*** - load and analyze new tape, this can only be done on an inactive tape.
- ***O*** - overwrite confirmation, if you do a tape change to a tape that has previously recorded data on it, you will need to send this command when you see the message from tapeserv (FROM TAPE:).
- ***R*** - rewind tape, this can only be done on an inactive tape.
- ***S*** - show tape to tapeserv, this can only be done on an inactive tape. This opens the path to the tape, so that tapeserv can monitor it's activity.
- ***T*** - retension tape, this can only be done on an inactive tape, and on a drive that supports retension.
- ***U*** - unload tape, on DAT drives this will eject the tape.

Dpda can be used to execute comlink commands to a remote system, the command line parameter is the station name of the remote system. Aqshell forks this process for the "Control Remote Station" menu item.

Servtest is an interactive program to communicate with digitizer servers. The optional command line parameter is the digitizer number, defaults to zero.

Comset can be used to change comlink settings from on a local system. Aqshell forks this process for the 'K' menu item. It has as it's command line parameter the name of the comlink data module.

Recset can be used to change recording enables on a local system. Aqshell forks this process for the 'G' menu item. It has as it's command line parameter the name of the acquisition common module.

Optset can be used to change verbosity of certain programs on a local system, or via the comlink shell command, from a remote system. With no command line parameter it will display the process ID, the module name (and optional modifier), the verbosity, and 3 other options, whose meaning is program dependent. You can show the values for only one module by using its name on the command line. You can change the verbosity by having two parameters, the name and the new verbosity. Likewise, to set option 1 you need to specify the name, the verbosity, and the new value for option1. This can continue on for up to 3 options, requiring 4 command line parameters. Some programs allow more than one copy, such as dacommo. In these cases the name must be the module name, a forward slash, and a modifier that is used to select which copy of the program you want to change, such as dacommo/dacombuf. The following programs have no modifiers and no options other than verbosity :

Aqsample, Aqcom, Aqblk, tapeserv, alert, aqdacserv, dss, procmon, and sysmon.

The following programs have modifiers or other options :

Module Name	Modifier	Option1	Option2
Dacommo	data module		
Rcman		<>0 = lock out cals during event (-L)	
Aqshell	path name	<>0 = show logical channels (-L)	>0 = show all channels (showall)
bufserv	C or E		
clock		Lockability	

Tape Operation

The two most important operations in tapeserv are the initial analysis of a tape and writing new data on that tape. The exact steps performed are determined by the *retension*, *append*, *verify*, and *stream* configuration variables.

Tape analysis is done when tapeserv initially starts, when a new tape is loaded, or when an error has occurred on a tape and retry is > 0. The steps taken are :

```
rewind tape
if retension
  then
    retension tape
    rewind tape
read first 4K record
if no volume header
  then
    rewind tape
    write 100 4K records
    rewind tape
    read and compare 100 4K records
    rewind tape
    if compare is ok
      then
        tape is blank
      else
        tape is faulty
  else
    if append
      then
        seek to end of data
        if record address > fullat
          then
            rewind tape
            tape is full
          else
            if stream
              then
                rewind tape
                seek to last 4K record
              else
                backup 1 4K record
            read last 4K record
            if SEED sequence agrees with record number and is less than warnat
              then
                tape is ready
              else
                rewind tape
                tape is full
      else
        rewind tape
        tape is full
```

The other major operation is writing logs to tape. Writing logs is normally triggered when any one of the logs exceeds the percentages set in the configuration file. Writing can also be triggered automatically if tapeserv detects that there are enough records in the data log to fill up the tape to the *fullat* parameter. Logs are also written on the current tape when a change to a different drive is requested. The logs will be written starting with the timing log, then the message log, and then the data log. It does not consider a log written until all new records in the log have been written, unless the tape reaches the *fullat* parameter in which case it will stop writing. The sequence for writing logs is :

```
if retension
  then
    retension tape
if SEED sequence = 0 {tape is empty}
  then
    rewind tape
    write volume header record (SEED sequence 0}
  else
    seek to end of data
    if verify and not stream
      then
        backup 1 4K record
        read last 4K record
        if SEED sequence on tape does not agree
          then
            tape is faulty, abort writing logs
          else
            seek to end of data
write contents of logs onto tape until logs empty or tape reaches fullat
if verify
  then
    if tape was empty before writing logs
      then
        rewind tape
      else
        if stream
          then
            rewind tape
            seek to last 4K record written before writing new logs
          else
            space backward to last 4K record written before writing new logs
    verify tape blocks until new end of data
if stream
  then
    rewind tape
mark log records as written to tape
if tape has reached fullat
  then
    seek to end of data
    write 40 file marks
    tape is full
```

Typical Configuration Values

Version 2.2 of tapeserv incorporates a new "fill to full" algorithm that will stop writing log files to tape once the *fullat* parameter has been reached. Note that if the EOT parameter is used, additional records past this value will be written, and consideration should be given to tape length tolerances and possible effects of writing data in groups instead of filling the tape all at once, as well as room for the File Markers. It is recommended that the *fullat* parameter be set to 90-95% of the rated tape size.

The following parameters are recommended for most drives and tapes :

overwrite=n	verify=y	append=y	report=y

Overwrite=y can be used if you have two or more tape drives to allow the oldest tape to be overwritten.

Archive 2150S Drive with DC6150 Tapes

warnat=130	fullat=140	retension=y

Archive 2525S Drive with DC6150 Tapes

warnat=130	fullat=140	retension=y	stream=y

Archive 2525S Drive with DC6250 Tapes

warnat=220	fullat=235	retension=y	stream=y

Archive DAT Drives

warnat=1100	fullat=1200	prevent=y

HP DAT Drives

warnat=1100	fullat=1200	prevent=y	semiload=y

Fax and Email Alerts

In order to send FAX messages the system must be equipped with a Telebit "WorldBlazer" or other compatible modem. There are two standards for FAX modems, the older "Class 2" and the newer "Class 2.0". The WorldBlazer uses the older Class 2 protocol and this software, as of now, has only been tested with the WorldBlazer. The software can share the modem with the normal dial-in retrieval function.

In order to send Email messages the system must have Ethernet installed and be connected to a Telebit "NetBlazer" or compatible network dial-up router, or have some other connection to the Internet. You must also have the "IP" address of an accessible SMTP server to route your mail.

Overview

- The "alert" program is responsible for :
  1) Building an internal list of all recipients of faxes or Emails and the detections that will cause the generation of the fax or Email.
  2) Monitoring the input pipe "evtreq" for detections. When those detections are received it will check the list to determine which recipients, if any, have specified that detection. For those recipients, their template file is converted into a temporary file that includes the specifics of that detection and the name of that file is passed to the input pipe for either a fax or Email to be sent.
- The "tmon" program :
  1) If the modem configuration file indicates that it supports class 2 fax operation it will read the configuration file and build a data module "FAXSVC" that contains the transmission queue.
  2) If fax operation is supported, it also creates an input pipe "faxreq" that the alert program (or any other user) can use to request that a file be transmitted as a FAX. The request has the format : <filename> ["DEL"]↵. The optional DEL tells tmon to delete the file after it has been successfully transmitted.
- The "sendfax" program is forked from "tmon" and tries to send a single FAX, this program is not normally started by a user.
- The "sendmail" program is responsible for :
  1) It reads the configuration file and builds a data module "MAILSVC" that contains the transmission queue.
  2) It creates an input pipe "mailreq" that the alert program (or any other user) can use to request that a file be transmitted as Email. The request has the format : <filename> ["DEL"]↵. The optional DEL tells sendmail to delete the file after it has been successfully transmitted.
  3) If there are any files in the queue it tries to contact a SMTP server to send all files in the queue.
- The "alertstat" program shows the current status of the FAXSVC and MAILSVC transmission queues.
- The "gif2qub" program is a utility that allows conversion of black and white GIF format graphics files into a "Quanterra Uncompressed Bitmap" file that can be used within a FAX.

Retry Specification

The fax and Email systems use a similar specification of how often to retry sending faxes and Emails. The first format is supported by both, the second only for faxes.

- ***<initial>TO<final>*** is the normal method and indicates that after the first failure, wait "initial" seconds before trying again. After each failure, the value is doubled, but not to exceed "final" seconds. For faxes, if the attempt was made after a wait of "final" seconds, a failure will remove the file from the transmission queue, as if it were sent successfully. For Email, the retry delay will go up to and then stay at "final" until all Email is successfully sent, at which point the delay returns to "initial".
- ***<count>FOR<seconds>*** is an alternate form for faxes only. It indicates that there are to be "count" number of retries, with a delay of "seconds" between each.

The same format is used in either the configuration file, or in individual fax templates.

Template Files

Template files are in a small subset of HTML (Hyper Text Markup Language). This language was chosen because it is simple to write and understand, and there is wide knowledge of it because it is used for World Wide Web pages. HTML comments are used to specify parameters for the fax and Email sending programs.

Character Set

There are only 3 special characters used in HTML, they are the less than sign "<", the greater than sign ">", and the ampersand "&". If you need to include these 3 characters into your file, the ampersand is used as an escape character :

```
&lt = <        &gt = >        &amp = &
```

Note that trailing spaces are ignored, and leading spaces are also ignored, except for preformatted text. Line lengths of HTML files are limited to 250 characters. The fax or Email output is limited to 70 characters per line.

Tags

Tags are used to specify a formatting option to the HTML parser. They usually consist of one word starting with a less than, and ending with a greater than, such as <BODY>. The following tags are supported by both the Email and fax programs, except for IMG :

- ***<!--html comment-->*** These are used to provide information such as fax number and detector specifications. These will be discussed later. All of these must occur before the start of the text.
- ***<BODY>*** Indicates the start of the actual text to send.
- ***</BODY>*** Indicates the end of the actual text to send, essentially end of file.
- ***<HR>*** Draws a horizontal line across a fax page, or a line of 70 hyphens "-" on an Email.
- ***
*** Line Break. The line structure of an HTML file is ignored, except for preformatted text, this tag is used to force the start of a new line. Without any line breaks, the text is simply word wrapped.
- ***<P>*** Paragraph. Similar to
 but adds a blank line.
- ***<PRE>*** Start of preformatted text. Line breaks are now taken into account in the file, keeping in mind the line length is limited to 70 characters. Also, leading spaces are also printed instead of being ignored.
- ***</PRE>*** End of preformatted text, return to normal mode.
- ***<UL>*** Start of un-numbered list. Used to begin an indented, bulletted list of items, such as the detector results shown on the first page. The indent is 3 characters, and these lists may be nested.
- ***</UL>*** End of un-numbered list.
- ***<LI>*** List item. Indicates the text that follows is a new list item.
- ***<IMG SRC="filename">*** Inserts a bit mapped file for faxes, is ignored for Email. The bit mapped file will be centered (left to right). The file must be in Quanterra Uncompressed Bitmap format and the double quotes are required.

HTML Comments

The HTML comments are used to provided needed routing, retry, and detector information for an individual recipient. The following comments are supported :

- ***<!--FAX=faxnumber-->*** indicates that this is a fax template and specifies the telephone number to dial. Normally the modem uses pulse dialing, but this can be changed to tone dialing using the T option. The string must conform to the "DIAL" command string specified in the WorldBlazer manual, but in summary :
  - ***0-9*** = Telephone number digits
  - ***T*** = Use tone dialing
  - ***W*** = Wait up to 60 seconds for a valid dial tone.
  - ***,*** = Pause for 2 seconds
  - ***@*** = Wait for silence after a remote ring before continuing.
  - ***!*** = Hook flash.
  - ***$*** = Wait for calling card billing tone prompt
- ***<!--ID=identifier-->*** will override the "faxid" parameter in the config file for this recipient. For example, if you know your fax machine will accept text, you could put something more meaningful than the telephone number here. Likewise, for Email messages this will override the "mailid" parameter.
- ***<!--FINE-->*** indicates to send a fax file in fine (200X200) resolution rather than normal (200X100) resolution.

- ***<!--EMAIL=address-->*** indicates that this is an Email template and specifies the recipient's Email address.
- ***<!--RETRY=retrystring-->*** is used to override the "faxretry" parameter in the config file for this recipient.
- ***<!--PRI=priority-->*** where priority is used to sort the recipients in the alert program. This doesn't have much effect on Email messages because they are all sent in a batch once the SMTP server is contacted. For faxes this will determine the order that the initial attempt to send a fax is done when a detection occurs. Since it can take a minute or more to send a fax, the person with the lowest priority may see a significant delay before they receive their fax. Higher priorities are sent first, the default priority is zero.
- ***<!--DET=detectorlist-->*** This determines under what conditions a recipient will be sent a fax or Email. The detectorlist is a series of detector names, separated by spaces. To specify an event detector you use the station name, a period, the SEED channel name, a colon, and the detector name, such as HRV.BHZ:SPWW. Coincidence detector names can also be used, and the SYSMON program can also generate alerts. Note that the detector list is limited to 80 characters, however, multiple Detector tags can be used. A particular type of detection can be accepted from any station by using an asterisk "*" instead of the station name.

FAX Template Structure

A typical fax template :

```
<!--FAX=Txxxxxxx--> <!--RETRY=30to200--> <!--PRI=2-->
<!--FINE--> <!--DET=CO1 HRV.BHZ:SPWW *.TO-->
<BODY>
<P>
<HR>
<IMG SRC="detalert.qub">
<HR>
FROM: Station RAND<BR>TO: Bob Reimiller<BR>
<HR>
($INCLUDE)
<HR>
<IMG SRC="seisconn.qub">
</BODY>
```

The "xxxxxxx" of course was replaced with the actual telephone number, and the T indicates tone dialing. The retry in the file overrides the default retry time-out. The priority is 2. The fax is to be sent in fine resolution. The following detectors are defined : CO1, HRV.BHZ:SPWW, and the acquisition timeout alert on any station generated by Sysmon. The <IMG SRC="detalert.qub"> puts in the "DETECTION ALERT" file, and the <IMG SRC="seisconn.qub"> puts in the "QUANTERRA YOUR SEISMIC CONNECTION" file. The ($INCLUDE) line is replaced by the "alert" program with the details of the detection.

If you wanted to identify the sender of the fax as something other than it's phone number, and your fax machine supports it, you could have something like <!--ID=STATION HRV--> anywhere before <BODY>.

Email Template Structure

A typical Email template :

```
<!--email=steim@quanterra.harvard.edu--> <!--PRI=2-->
<!--DET=CO1 HRV.BHZ:SPWW *.TO-->
<BODY>
TO: steim@quanterra.harvard.edu<BR>
SUBJECT: Detection Alert<BR>
<HR>
($INCLUDE)
<HR>
</BODY>
```

This file has the same detector list and the same priority as the fax template. Retry is not specified for Email templates because the retry is for contacting the SMTP server, not for individual recipients. Note that the first line of the body needs to be in the form "TO: name", or else your Email will say "Apparently To", just a quirk of SMTP. The second line needs to be the "SUBJECT: description" line, or else your Email won't have a subject. If you wanted a return address that is someone other than the one specified in the "mailid" configuration parameter, you could have added <!--ID=address--> in the template anywhere before <BODY>.

FAX Modem Configuration

In order for "tmon" to setup a fax transmission queue and input pipe, there must be a file called "config_modem" either in the current directory when tmon is started, or in /r0. This file must have the line "TCMDFAXSTD=2" somewhere in the file, after the first two lines that don't start with "%".

Detectors

In order to generate a an alert, a program only needs to write to the "evtreq" pipe, using the format :

```
DETNAME (<text> | FILE=filename [DEL]) ↵
```

The entire entry must be less than or equal to 250 characters to be accepted. For example, if TAPENOD is the detector name for the tape going off-line, sysmon could write "TAPENOD Tape off-line at 96/05/13 18:01:17↵" to the pipe and anyone who specified TAPENOD in their template would receive the fax or Email. If the detector name and text will not fit into 250 characters you can put the text into a file, and then use the second form of the request. For example "TAPENOD FILE=systemp DEL↵" will cause the alert program to read from file "systemp" to get the text, and then delete it once it has generated all the alert files. If the DEL string is not included then the file will not be deleted by alert.

Note that the text or the content of the file are considered HTML text, along with the rest of the templates. If you place no HTML tags in the text it will just word wrapped. Note that individual lines in the file must be terminated by a↵ (carriage return) and that no line may exceed 250 characters. For instance, AqSample might write to the evtreq pipe :

```
HRV.BHZ:SPWW Murdock-Hutt Detection on HRV.BHZ:SPWW<UL>
<LI>Time=1996/03/30 07:18:42.078<LI>Quality=C A 0  00929
<LI>Peak Amplitude=4455<LI>Period=0.50<LI>Noise=4</UL> ↵
```

Opaque Data Blockette Piped Record Format

The Aqblk program was created to allow GPS or other data to be written as a series of blockettes. Aqblk does not care about the contents of the blockettes, other than it be SEED opaque blockette, so that it is not specfic to handling GPS data, but for the purpose of this docuement, that is the intended use. The data sent to the pipe from the user's GPS receiver communications program consist of a 20 byte header, the actual blockette, and a 2 byte checksum. The pipe name will be in the form of "/pipe/blk.<station>.gps", such as "/pipe/blk.hrv.gps". The header is :

Byte Offset	Contents
0-1	SEED Location, normally spaces
2-4	SEED Channel, such as "GPS"
5	Flush Flag : Bit 0 TRUE indicates that comlink output should be flushed after this blockette. Bit 1 TRUE indicates that blockette log output should be flushed after this blockette.
6-7	16 Bit integer blockette length (not counting this header or checksum)
8-17	Time stamp for blockette, standard "BTIME" SEED format.
18-19	Reserved, must be zero.

This header is followed by the blockette, which is a maximum of 456 bytes in length, and then a two byte checksum which is calculated by doing a 16 bit addition over the length of the header and blockette. If the blockette length is odd, then the checksum is calculated based on a zero pad byte.

Our software will build data records using the standard 48 byte data record header followed by a blockette 1000, followed by the blockettes, up to the appropriate 512/4K byte limit, or until the "flush" flag is encountered.

So that we can assemble data records without knowing the format of individual records we insist that all blockettes have the following format as the first 14 bytes (Variable length opaque data blockette) :

Byte Offset	Contents
0-1	Blockette Type = 2000
2-3	Next blockette's byte number. Always zero when you write the blockette to the pipe.
4-5	Length of blockette in bytes.
6-7	Offset from start of this blockette to start of data.
8-11	Record Number
12	Word Order, 1 = Motorola Format
13	Data Flags, Bit 1 set indicates that a new record is required.

The command line for the Aqblk program is :

aqblk [c=configfile] [-v=n] >>logdevice_or_file

The config file defaults to /r0/aqcfg. n is the verbosity level. Error/status messages are written to the standard error path.

DSS

All DSS (Data Subscription Service) packets use the QSP (Quanterra Serial Protocol) header structure (QDP). The basic transport mechanism is UDP/IP with the QDP header providing a CRC and sequencing control information.

QDP Format (12 bytes)

CRC		
Command	Version	Length
Sequence #	ACK #	Spare (Zero)

- **CRC** is the standard Quanterra CRC value, and includes this header (not including itself) and any data may follow.
- **Command** is a Quanterra specific command, indicating the content of any data that follows.
- **Version** is used to accommodate updates to the protocol.
- **Length** is the actual length of the data, and may be between 0 and 512 and must be even.
- **Sequence** number is a modulus 256 record number to providing sequencing of datagrams. This is normally set by the client so that it can track responses from the server. Timeout resets can leave this field zero, since there is no response. The server leaves this field zero for data packets.
- **ACK** number is used to acknowledge previously received datagrams. The server sets this to the sequence number of the client request so the client can associate responses with requests. The server leaves this field zero for Data packets.

DATA Format

The format of the DATA area is dependent on the QDP Command byte :

DSS_REQ ($F0)

Used to request a data item be added from the list. Each request has the following header :

Data Identification Word	Priority
Reporting Interval (seconds)	

The data identification word is a unique word generated by the client that will be attached to the data so the client can identify it. This word is also used when deleting requests. Priority is local to this client, with higher priority requests being sent first. The reporting interval indicates how often the DSS server sends this data to the client. For simple data and GPS requests, a value is picked out of the data stream at this interval, no filtering is done. For other requests, multiple samples at the source data rate will be processed for that many seconds before reporting. For instance, if the source data is 20Hz, and the reporting interval is 5 seconds, there will be 100 data values processed. Selecting an interval of zero results in one report being generated and the request removed.

Following the header is the data request. The format of each request is still under development, but the following have been defined so far :

$01=Simple Data	Report Flag	Station (first 2 characters)	
Station (second 2 characters)		Location	
Seedname			Format

Data is sampled from the data stream at the reporting interval. Bits 0-2 of format are used to determine data format :
1=16 Bit Integer
3=32 Bit Integer
4=IEEE floating point (32 bits)
5=IEEE double precision floating point (64 bits)

The Report Flag determines what action the server will take when the requested data is not available, such as when acquisition is not operating. A value of zero indicates that the server merely skips this report. Values between 1 and 249

request the server send that many dummy data reports (with the appropriate bit set to indicate Data Not Available (REF_DNA)) before skipping the report. A value of 250 tells the server that you want the dummy data reports continuously. Values from 251 to 255 are reserved. In any case as soon as the server has determined that the data is available again, it will resume sending actual data reports.

$02=GPS Location	Report Flag	Station (first 2 characters)
Station (second 2 characters)		

Reports the latitude, longitude, and elevation as reported by the GPS clock.

$03=Dead Channels	Report Flag	Station (first 2 characters)
Station (second 2 characters)		

Reports the number of channels that have died on the specified digitizer.

$04=Detection Count	Report Flag	Station (first 2 characters)
Station (second 2 characters)		Location
Seedname		$00
Detector Name (Pascal String)		

Reports the number of detections that have occurred on the specified detector over the reporting interval.

$05=Record Count	Report Flag	Station (first 2 characters)
Station (second 2 characters)		Location
Seedname		$00

Reports the number of 512 byte (max) data packets on the specified channel.

$06=Time since Boot	Report Flag

Reports the number of seconds since the system re-booted.

$07=Min,Max, & Avg	Report Flag	Station (first 2 characters)
Station (second 2 characters)		Location
Seedname		Format
Seconds		

Reports the maximum value, minimum value, and average value on the specified channel. Format is the same as for simple data. The source is always the digitizer. Seconds determines the total number of data points (Seconds * Data Rate) are used for the report. Seconds must always be equal to or less than the reporting interval.

$08=$\int Val^2$ & Avg	Report Flag	Station (first 2 characters)
Station (second 2 characters)		Location
Seedname		Format
Seconds		

Reports the summation of the square of each data value divided by the number of points, along with the average value on the specified channel. Fields are the same as Min, Max, & Avg Request.

DSS_DEL ($F1)

Used to request that one or more data items be removed from the list :

Number of Entries	Data Identification Word 1
Data Identification Word 2	.....Data Identification Word N

DSS_TOR ($F2)

Time-out reset. The client must send this packet periodically to continue receiving packets, the anticipated default time-out is 10 minutes at this time.

DSS_REG ($F3)

Requests registration with the server. Any data requests will be ignored until the client has registered. Registration requires a password, different passwords corresponding to higher or lower "privilege". The higher the privilege, the faster data for that client will be processed, and the less likely that requests will be refused. The password is a Pascal String and uses 8 bytes. The Client Name is a Pascal string (up to 250 characters) and uses length+1 bytes, padded if needed to make the packet even length.

Password (string[7])
Client Name (string)

DSS_LRQ ($F4)

Requests that the server send a DSS_RQL packet listing all requests for this client. There are no data fields.

DSS_REM ($F5)

Requests that the sending client be removed from the server's list of clients. This should be sent by a client before exiting, but is not absolutely required.

DSS_RQL ($FF)

Lists all pending requests for this client along with current status :

Number of Entries	Data Identification Word 1
Status 1	.....Data Identification Word N
......Status N	

Status Codes :
$8000 OFF_DNA Data not available

DSS_REF ($FE)

Server refuses request

Refusal Code

Refusal Codes :

1	REF_TO	Timeout	8	REF_DUP	Duplicate Data_ID
2	REF_IDS	Invalid Data Source	9	REF_OOM	Out of Memory
3	REF_SEI	Seconds exceeds Interval	10	REF_IDF	Invalid Data Format
4	REF_INP	Invalid Password			
5	REF_URC	Unregistered Client			
6	REF_TMR	Too many requests			
7	REF_URQ	Unknown Request			

DSS_DAT ($FD)

A data packet consists of one or more records. The offset to the next entry is relative to the start of actual data record (past the QDP header). An offset of zero indicates the last record. The offset is the lowest 9 bits (OFF_MASK=$1FF) of the "Status and offset to next entry" field. The upper bits are status, such as OFF_DNA ($8000).

Simple Data entries have the following format :

Data Identification Word	Status and offset to next entry
Time field (IEEE double precision) - seconds since 1984	
Value (16-64 bits)	

GPS Location Data entries have the following format :

Data Identification Word	Status and offset to next entry
Latitude (-xx.xxxxxx degrees) 10 bytes	
Longitude (-xxx.xxxxxx degrees) 11 bytes	
Elevation (-xxxx.x meters) 7 bytes	

Dead Channel requests have the following format :

Data Identification Word	Status and offset to next entry
Number of Dead Channels	

Detection Count requests have the following format :

Data Identification Word	Status and offset to next entry
Number of Detections	

Record Count requests have the following format :

Data Identification Word	Status and offset to next entry
Number of Records	

Time since Boot requests have the following format :

Data Identification Word	Status and offset to next entry
Seconds since reboot (IEEE double precision)	

Min,Max, & Avg requests have the following format :

Data Identification Word	Status and offset to next entry
Time field (IEEE double precision) - Seconds since 1984	
Minimum Value (16-64 bits)	
Maximum Value (16-64 bits)	
Average Value (16-64 bits)	

$\int Val^2$ & Avg requests have the following format :

Data Identification Word	Status and offset to next entry
Time Field (IEEE double precision) - Seconds since 1984	
Sum of Squares (16-64 bits)	
Average Value (16-64 bits)	

DSS_ACK ($FC)

This acknowledges packets from the client when there is no error or return value, these include DSS_REQ and DSS_DEL.

DSS_PRG ($FB)

This packet tells a user that a report has been purged.

Data Identification Word	Purge Code

Purge Codes :

1	PRG_ESL	Excessive Server Loading (DSS is using too much CPU time)
2	PRG_BPS	Bytes Per Second (Too much data being transmitted by DSS)

Adding New Reports

Request to add new reports to the DSS server need to be submitted to Quanterra. In order for us to validate our implementation we must have some example input data along with the expected output. Any parameters that DSS needs must be passed as part of the DSS_REQ packet. The format and purpose of these parameters must be explained in full so we can document them.

We are currently looking into adding Hiroo Kanamori's Peak Acceleration algorithim, pending meeting the above requirements.