

Customizing the DAQ code for your site

Paul Hubbard
April 23, 2003

Outline of talk

- Walk through API calls and their use
- Taken from 'Example DAQ.vi'
- See 'Example DAQ' and others for API usage in context
- Questions welcome!
- Follow along at <http://www.mcs.anl.gov/neesgrid/example/>

Metadata save.vi

- Writes 'Event ID' (opaque string), channel names (comma delimited), sample rate (0 for variable), data file name and channel units (from MAX) to an INI file.
- Read by 'server daemon.vi' & datafile routines (saved into file header)
- Call this before any DAQ is started

Write file header.vi

- Writes metadata into a slightly different format, designed for parsing by the repository
- Call before acquiring any data; easiest to call right after metadata save and chain the errors
- Will overwrite any existing file with no warning!

DAQ Status Write.vi

- Writes one of the enumerated statuses into the shared (global variable) area
 - Running, stopped, error, etc
- Read by 'server daemon' and sent to NSDS upon request.
- Should be called whenever status changes
- Can also add your own strings, see 'Get status and time.vi'

Data array to NSDS stream.vi

- Input: Vector of real-valued measurements and a list of associated channels
 - Order must match!
- Output: ASCII data, with timestamp in ISO8601 format, with only subscribed channels included
- Reads from the global list of subscribed channels, as maintained by server daemon.
- Can set timestamp and precision

Example:

2003-01-24T15:42:02.73399	ACH1	-0.700531	Temp	27.603149
2003-01-24T15:42:02.75000	ACH1	-1.961975	Temp	27.587891
2003-01-24T15:42:02.75500	ACH1	2.678223	Temp	27.618408

TCP Conditional data send.vi

- If no connection, returns immediately
 - Status and channel handle maintained by server daemon and this routine
 - No CPU load if no network connection
- If no data subscribed, returns immediately
- If error returned from TCP:
 - changes status of network to 'disconnected' - one time glitch only
 - The driver will initiate a reconnect, streaming resumed by the NSDS

Data array to datafile stream.vi

- Formats *all* channels to save to disk
 - Tab-delimited, channel IDs not repeated
 - We assume all data goes to disk
 - Therefore, only inputs data and not channel IDs
- Just builds the string; doesn't save

E.g.

Event ID: 19364841J12
Active channels: ACH1,Temp
Sample rate: 200.000000
Channel units: V,Deg C

Time	ACH1	Temp
2003-01-24T15:42:02.73399	-0.700531	27.603149
2003-01-24T15:42:02.75000	-1.961975	27.587891
2003-01-24T15:42:02.75500	2.678223	27.618408

FTP Three-phase write.vi

- Uploads finished data file to FTP server, sometimes called the local repository
- Requires NI 'Internet Toolkit' add-in package
- Writes datafile.txt, then datafile.txt.written
 - Filename just an example
 - .written is a signal to the repository that uploading is completed
- Sometimes first upload fails, reason unknown. Also need tcpfix.llb!

Things done with LV code

- Save to disk - use their code that opens and closes the file at each step.
- DAQ - this allows the easy drop-in of third-party hardware (e.g. UEL) or software
- INI files

Server daemon interactions

- Sets TCP status, which governs the data sender
- Sets connection handles for control and data
- Repeats status, list of channels, channel units
- Handles subscriptions and associated list
 - No data sent unless server says so!

Other Notes

- Examples are low-speed code; not tuned for throughput
- One-time glitch on TCP error
 - Queueing code took more CPU, but is in CVS for exploration
- Server daemon is TCP-responsive; needs to poll the socket for better interactive response
- NSDS simulator essential for testing