Building the National Virtual Collaboratory for Earthquake Engineering Research

NEESgrid

Technical Report NEESgrid-2003-01
www.neesgrid.org

Draft Whitepaper Version: 1.0
Last modified: January 10, 2003

Installing the NEESgrid Data Acquisition Code

Paul Hubbard[1]

[1] Argonne National Laboratory

Feedback on this document should be directed to hubbard@mcs.anl.gov

Introduction

This document is intended for NEESgrid sites wishing to install the supplied code for data acquisition (DAQ) and data handling. It covers getting the code, installation, testing and some ideas for site customization.

Other Suggested Reading

The 'NSDS-Driver' document covers the details of the data transfer and control protocols. If you want to modify the driver, change the streaming code, or just understand the details, that is a useful item.

The 'Driver installation' document covers downloading and running the NSDS (NEESgrid Streaming Data Server) driver, a piece of C code that interfaces between your DAQ and the NSDS.

Requirements

You will need to have the driver and NSDS installed in order to test data streaming. For the repository functions to work, you'll need an FTP server (which we strongly recommend only be available to the DAQ PC for security reasons. For DAQ testing, you need only the DAQ PC itself. The DAQ PC needs:

1) LabVIEW 6.1
2) LabVIEW Internet toolkit
3) A Labview-supported data acquisition board
4) A CVS client, such as Tortoise
5) As noted in the system architecture document, Windows XP/2000/NT

Background

As part of the NEESgrid effort, the System Integration team supplies code to implement the functions we need, such as streaming data, site repository integration, and so forth. We also supply example implementations of DAQ code that are intended as starting points for further customization.

The LabVIEW code you will be working with implements the following features:

1) A server daemon to listen for driver / NSDS requests
2) Client library for your use, containing subroutines for
    a. Streaming numeric data to the driver / NSDS
    b. Save to disk (tab-delimited ASCII) in an agreed-upon format for the repository
    c. Convert numeric data to ASCII for streaming and saving
    d. Download of metadata, for repository-controlled DAQ
    e. Generate fake data for testing
    f. Read and write experimental metadata

   g. Read and write commands and data from TCP ports
   h. Generate ISO 8601 timestamps
3) Example code
   a. Fake DAQ
   b. Semi-fake DAQ (Low speed analog input, multichannel)
   c. Simplest Fake DAQ, stream only
   d. Zombie DAQ – repository-controlled DAQ
4) Testing code
   a. NSDS simulator
   b. NSDS stress tester
   c. Stress test fake DAQ
   d. Super Zombie DAQ

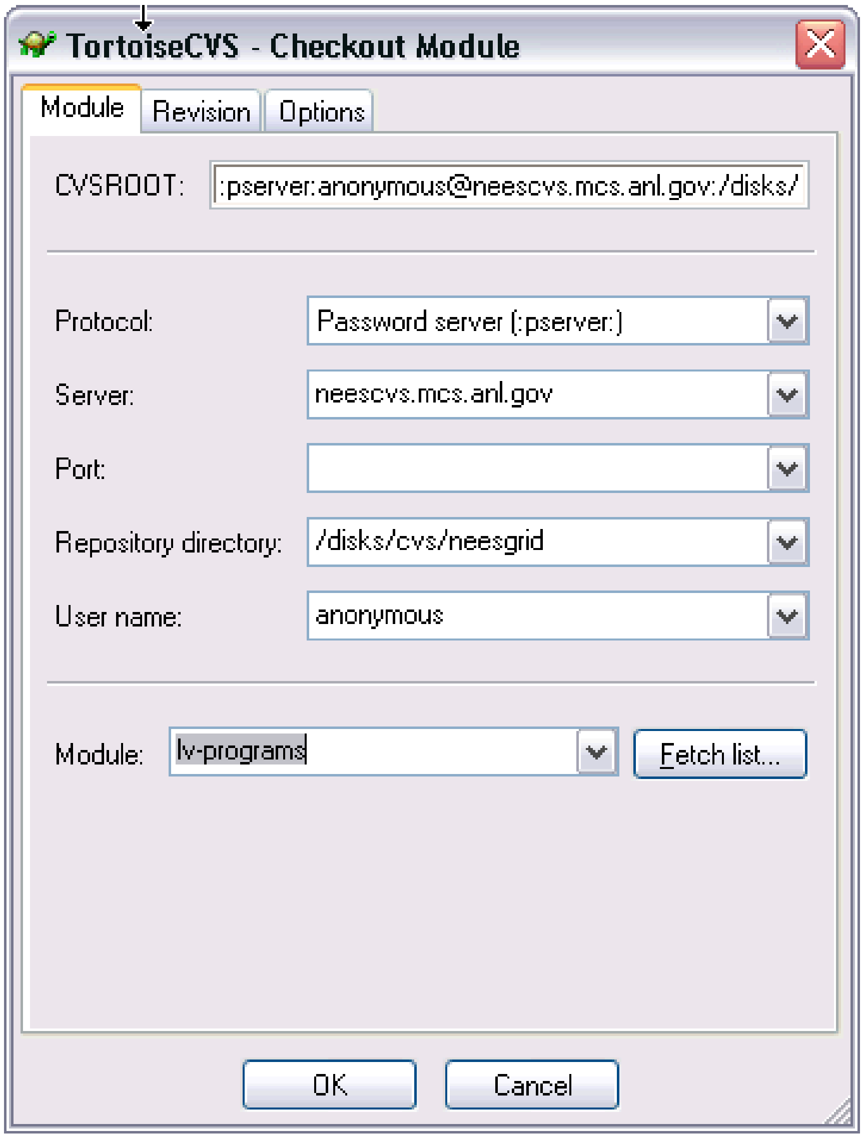
This document is intended to get you started; it does not cover all of the above. If are interested in the other components, I urge you to sign up for the DAQ mailing list, and ask questions there. Instructions can be found at http://www.mcs.anl.gov/neesgrid/

Getting the Code from CVS

The NEESgrid code is hosted in a CVS archive. CVS, the Concurrent Version System, is a method of source code control; it tracks code modifications and who made them. For the purposes of this document, consider CVS as a new method of getting things, similar to HTTP or FTP.

The recommended CVS client for Windows is called Tortoise. It is free (GPL) and available from http://www.tortoisecvs.org/

Once you have installed it and have rebooted, right-click on the desktop and select 'CVS Checkout…' from the menu. Fill it out as shown here:



CVSROOT: :pserver:anonymous@neescvs.mcs.anl.gov:/disks/

Protocol: Password server (:pserver:)

Server: neescvs.mcs.anl.gov

Port:

Repository directory: /disks/cvs/neesgrid

User name: anonymous

Module: lv-programs

Once you press Enter, the Tortoise program will download the code into a folder on your desktop called ‘lv-programs’. (This is the name of the project in CVS containing the code.)

Functional Breakdown

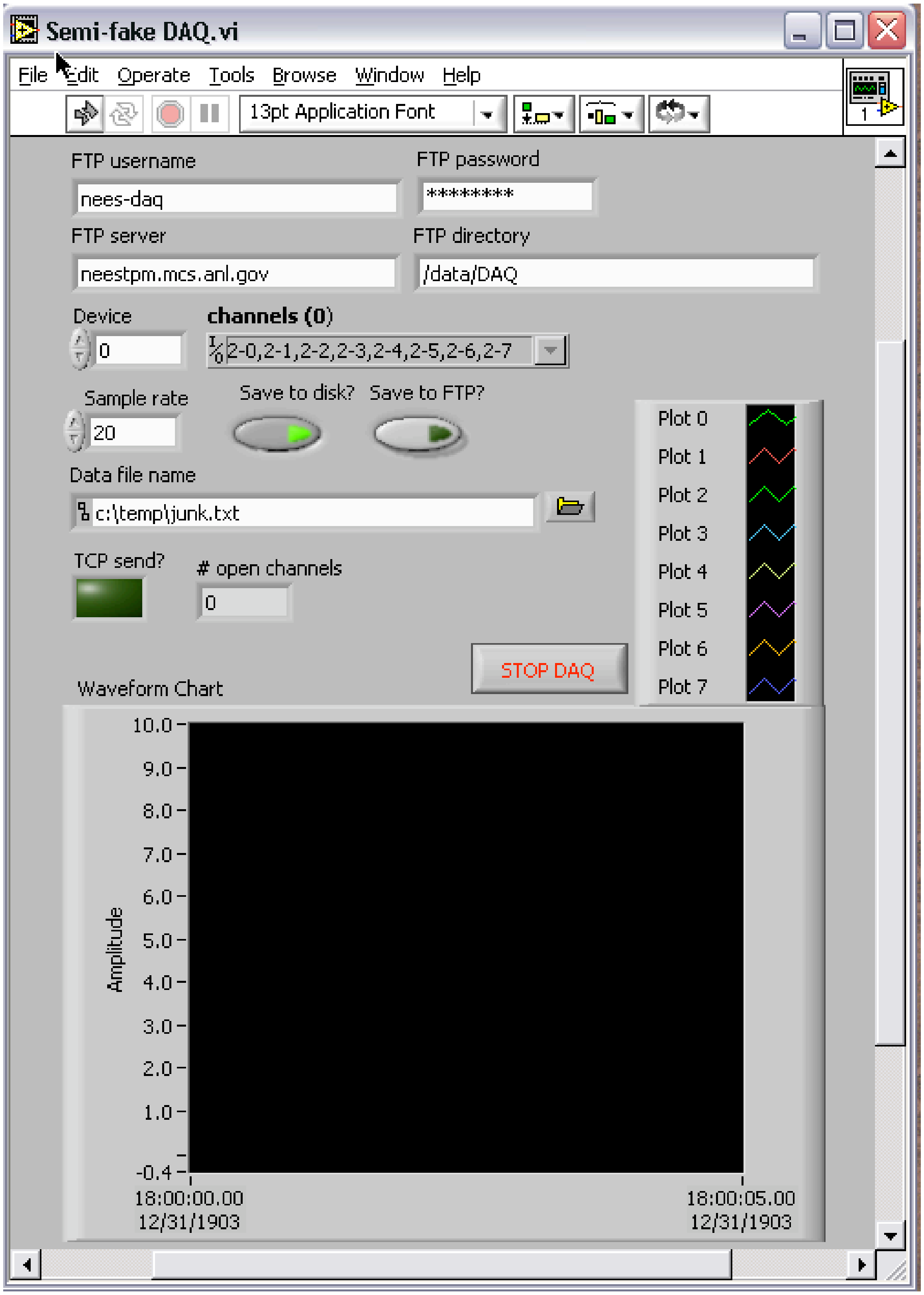
There are two major pieces of code used – the server daemon, and the client library. The server is responsible for talking with the driver / NSDS, and sets up things like TCP connections, subscribed channel lists and so forth. The client library is called from the DAQ program to stream data (If anyone has subscribed), and all of the other supporting functions.

The Server Daemon

Open ‘nees-progs.llb’ and then ‘server daemon.vi’. Go ahead and run it. This **must** be running any time you want to stream data, allow others to see the DAQ’s status, etc.

DAQ Programs

Next, open ‘Semi-fake DAQ.vi’. This is a simple example program that reads analog input data, graphs it in realtime, and streams it out to the driver / NSDS if a subscription is active. You will need to edit the channel list to match your local configuration.


Semi-fake DAQ.vi

FTP username: nees-daq

FTP password: *********

FTP server: neestpm.mcs.anl.gov

FTP directory: /data/DAQ

Device: 0

channels (0): 2-0,2-1,2-2,2-3,2-4,2-5,2-6,2-7

Sample rate: 20

Save to disk?

Save to FTP?

Data file name: c:\temp\junk.txt

TCP send?

open channels: 0

Once you have selected channels, press the Run button, and you should see data being acquired and graphed.

[screen capture – data capture]

Save to Disk

You'll notice a couple of buttons on the Semi-Fake DAQ front panel, labeled 'Save to disk' and 'Save to FTP'. Let's try 'save to disk' first, and then try FTP.

First, stop the VI if its running, and then enter a filename into the field. This is where your data will be saved. I usually use c:\temp\junk.txt for testing.

Secondly, press the 'Save to Disk' switch and hit run. It should light up green, indicating that data will be saved to disk. You will see data being captured as above, only now its also being saved to disk for later analysis.

Press 'Stop DAQ', and then open the datafile with your analysis tool of choice, e.g. Excel, Kaleidagraph, MATLAB, DaDSP, etc. It should import OK, though you may have to tell the tool to skip the first six lines of metadata.

You might also want to look at those header lines by eye, just to take a look at what's being saved and how it's formatted.

FTP Upload to Repository

Once you have Save to Disk working, stop the program and take a look at the four fields at the top – FTP username, machine name, remote directory, and password. This is where your local Unix sysadmin comes in handy you need him or her to set up an FTP server on your local repository (or test server) for this to work.

This is how it works: When the experiment is complete, the FTP code is called to upload *two* files, the data file and a semaphore file with '.written' appended to the filename. The '.written' file tells the repository that the upload is complete, and that it may import the data into the archive.

This function uses and requires the National Instruments Internet Toolkit, which contains the FTP routines. This is not included with LabVIEW; it's a separate toolkit that you need to purchase.

Testing Data Streaming and the NSDS Simulator

Next, we will test the streaming of experimental data. We'll need the following to be running:

1) A data source, such as Fake DAQ or Semi-Fake DAQ
2) The server daemon, to manage the communications
3) The NSDS simulator.vi (or NSDS stress tester) to subscribe to the data

4) The driver (probably running on a separate Unix machine)

(Note that you have to give the NSDS simulator a channel name – these are configured in the NI ‘Max’ program, and by default are usually of the form 1-0, 1-1, etc.)

Start 1-3 first, then go to the Unix machine and run the driver, telling it the name of the DAQ machine:

```
./driver -daq_machine {machine name} -nsds_machine {machine_name}
```

e.g.,

```
./driver -daq_machine neeslabview2 -nsds_machine neeslabview2
```

Check the output of the driver – it should report that it connected successfully to the DAQ, and is try

You should see the server daemon process several messages, all four LEDs on its front panel should light up green, and data should start appearing in the NSDS simulators’ chart.

Further Steps

At this point, you should have working DAQ and data streaming, and can consider yourself a first-class citizen of the NEESgrid. From here, you can explore the code, experiment with the various stress test applications, or start modifying the code to suit your lab.

Please consider joining the NEES-DAQ mailing list and contributing your code and feedback to the community at large.