

Technical Report NEESgrid-2004-04

www.neesgrid.org

(Whitepaper Version: 0.5)
Last modified: March 1, 2004

Guide to the NEESgrid Reference Implementation

David Gehrig¹

¹ National Center for Supercomputing Applications,
University of Illinois at Urbana-Champaign

Feedback on this document should be directed to dgehrig@ncsa.uiuc.edu.

Acknowledgment: This work was supported primarily by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0117853.

Table of Contents

1.	About This Document	3
2.	The NEESGrid Reference Implementation	3
2.1.	What is NEESgrid?.....	3
2.2.	The Reference Implementation	3
2.3.	The MOST Experiment.....	4
3.	The Two Reference Experiments.....	5
3.1.	Block Diagrams	5
3.2.	Boxology in Brief	7
4.	Requirements for the Three-Site Simulation	7
4.1.	The NEESpop Server.....	7
4.2.	The Computational Node (“MatLab Machine”)	7
5.	Installing and Running the Three-Site Simulation.....	8
5.1.	Installing the NEESpop.....	8
5.2.	Installing the MatLab Computational Node.....	10
5.3.	Running the Three-Site Simulation	10
6.	Requirements for Mini-MOST.....	10
6.1.	The DAQ PC	10
6.2.	The TPM Server	10
7.	Constructing the Mini-MOST Device	11
8.	Setting up the TPM Server (if any)	11
9.	Installing the DAQ Software	11
10.	Running the Mini-MOST Experiment.....	12
11.	Glossary	12

Table of Figures

Figure 1:	The MOST Structure.....	4
Figure 2:	Subassemblies of the MOST Structure	5
Figure 3:	Block Diagram of the Three-Sites Simulation.....	6
Figure 4:	Block Diagram of the Mini-MOST Simulation.....	6

1. About This Document

This document describes the NEESgrid Reference Implementation, two standard NEES-based experiments which demonstrate the capabilities of the NEES system. This guide will provide an overview of the NEES system, a discussion of the two standard experiments distributed with the NEESgrid software, and an explanation of how to get these experiments up and running.

This document does not cover all facets of the NEESgrid system. More NEES information of all kinds is available at www.neesgrid.org; this document will provide references into the NEESgrid site when appropriate. In particular, there will be references to NEESgrid Technical Reports; these are available for download in PDF format at www.neesgrid.org/news/documents.php.

This document does not describe every step in the installation procedures necessary to get the NEESpop up and running starting from a fresh-out-of-the-box collection of computers. For complete and detailed instructions, on a component-by-component basis, you'll want to visit www.neesgrid.org/documents/Release2.2. That's where you'll get the full story.

A glossary of NEESgrid-related terms is at the end of this document.

2. The NEESGrid Reference Implementation

2.1. *What is NEESgrid?*

Technically, NEESgrid is a collection of software services designed for use at NEES experiment sites or by remote collaborators. Loosely, NEESgrid is the software backbone that makes the NEES network possible; a “software service” just means a program running somewhere on the system which you (or your equipment) can interact with in certain defined ways. Through NEESgrid software, researchers can communicate with each other, plan and execute experiments, and publish results using the research equipment and computing power that are available through NEES.

For more about the organization and goals of NEES, see www.nees.org. For more about NEESgrid and the NEES System Integration effort, see www.neesgrid.org. For a description of the NEESgrid services, see the appendix of this document.

2.2. *The Reference Implementation*

Because the NEESgrid software consists of several different types of software services and is also dependent on the research equipment that it will support at a given site, a Reference Implementation has been created to act as a reference model, demonstrating how the services of NEESgrid interact.

The Reference Implementation includes the software necessary to run two complete experiments, both of which are miniaturized versions of the large-scale experiment discussed in the next subsection. By setting up your own NEES installation and running these experiments, you can confirm that your system has been properly set up. You can also use the standard experiments as examples you can then alter to work with your own experiments.

2.3. *The MOST Experiment*

In July 2003, NEESgrid was used to perform an experiment that would have otherwise been impossible. A simple steel framework was subjected to a pseudo-dynamic test. This framework was split into three subassemblies. One subassembly was built in Boulder, Colorado; a second subassembly was built in Champaign, Illinois; the third wasn't built physically at all, but was modeled in a MatLAB numerical simulation at a different site on the University of Illinois campus. These components were coupled using NEESgrid and marched through a 1500-time-step "quake." Figure 1 shows the structure that was modeled in the MOST experiment. Figure 2 shows how that structure was split into subassemblies. The subassemblies are named by their location during the MOST experiment; the Mini-MOST hardware, for example, simulates the subassembly at the University of Colorado at Boulder, and is therefore referred to in the MatLAB code as "UCol." The other two components are "NCSA" and "UIUC."

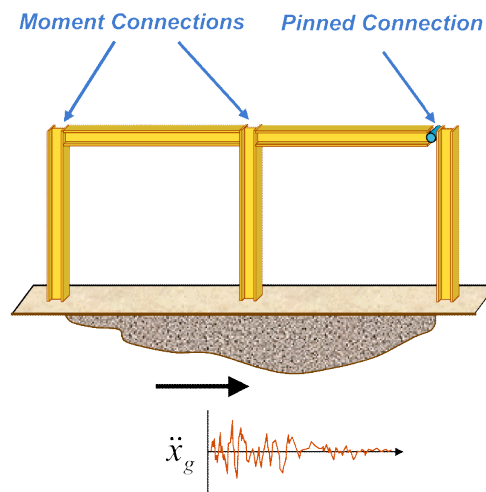


Figure 1: The MOST Structure

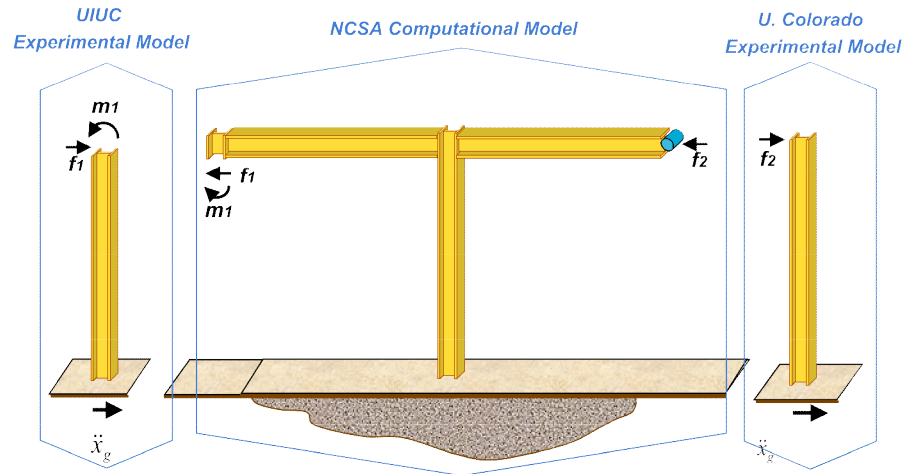


Figure 2: Subassemblies of the MOST Structure

More information about the MOST experiment itself is available at www.neesgrid.org/most.

3. The Two Reference Experiments

The two reference experiments described in this document involve the same theoretical setup: the same structure is being modeled, and it is broken into the same subassemblies. However, in the Mini-MOST experiment, two of the components are modeled numerically, while the third is performed physically in tabletop format. In the Three-Site Simulation, all three of the subassemblies are numerically simulated on the same computer.

We recommend that researchers planning to perform these experiments begin with the Three-Site Simulation and then make the necessary modifications to perform the Mini-MOST experiment.

3.1. Block Diagrams

Here are block diagrams for the Three-Sites Simulation and the Mini-MOST Simulation. For an explanation of the components within the box diagrams, see Section 3.2, Boxology in Brief, on page 7 and Section 11, Glossary on page 12.

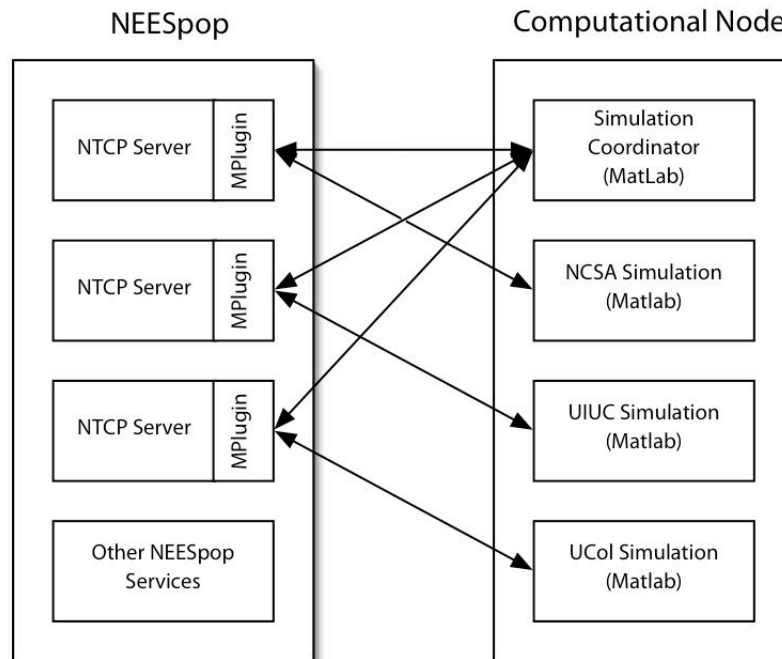


Figure 3: Block Diagram of the Three-Sites Simulation

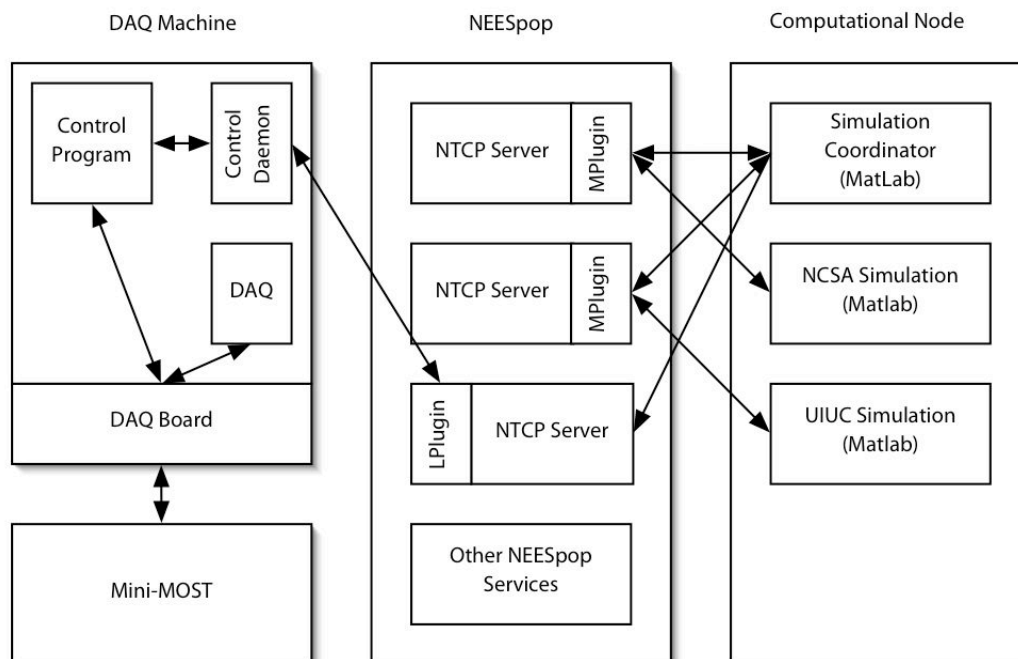


Figure 4: Block Diagram of the Mini-MOST Simulation

3.2. *Boxology in Brief*

The boxes represent either computers or programs (“processes”) running on each computer at the same time.

The Three-Site Simulation, shown in Figure 3, requires two computers, one a Linux box acting as the NEESpop, and the other a Windows machine running Matlab programs representing different subassemblies of the MOST structure. There is also another Matlab program acting as simulation coordinator; it gives orders to the simulation components through NTCP, and collects the data at the end of each step. It also keeps the simulation components synchronized.

The Mini-MOST configuration, shown in Figure 4, is similar to the Three-Site Simulation, except that the UCol subassembly is modeled physically rather than computationally. Another PC is brought into the system to act as the DAQ machine, which interfaces between the NEESpop and the data acquisition hardware (including a which, in turn, interfaces between the DAQ machine and the sensors and controls on the Mini-MOST.

Note that the Mini-MOST block diagram doesn’t include the optional Telepresence Management system for the web cameras.

4. Requirements for the Three-Site Simulation

4.1. *The NEESpop Server*

- 1+ GHz dual Pentium-III or Pentium IV
- 1 GB RAM 2 x 64-bit, 66 MHz PCI slots
- 36 GB SCSI disk
- 2 x Gigabit Ethernet NICs
- Redhat Linux version 7.2 or 7.3

4.2. *The Computational Node (“MatLab Machine”)*

The machine which will run the MatLAB computational components should be a PC with at least 128MB memory – although the more the merrier – and at least a 1 GHz processor. It will also need an IP connection to communicate with the NEESpop.

The computer should also be running Windows XP. (There is no theoretical reason the software shouldn’t run using Windows 2000, Mac OS X, or UNIX, but it hasn’t been fully tested on those platforms.)

The computer will also need MatLab 6.5, and a few other software prerequisites. Descriptions and installation instructions for these components appears under the heading “MatLab NTCP Toolboxes” at <http://neesgrid.org/documents/Release2.2>.

5. Installing and Running the Three-Site Simulation

5.1. *Installing the NEESpop*

- **Request NEESgrid user accounts (“Alliance Accounts”).** Security in grid services like NEESgrid use something called X509 certificates (sometimes called “credentials files”) to control authorization. You will need to apply to the National Computing Science Alliance to receive one of these files, and this process may take several days, so do this as early as you can. See http://www.neesgrid.org/get_involved/user_accounts.html for full instructions.
- **Login as a root user.**
- **Go to the NEESgrid website.** The latest release of NEESgrid software is always available for download at <http://www.neesgrid.org/software>. The release comes in the form of four packages in the form of downloadable compressed archives (“tarballs”). Comprehensive documentation is also available here for system requirements and installation of NEESgrid software. This document provides an overview of what you’ll find in more detail there.
- **Read through the requirements, installation documentation, and (if any) errata documents.** Installing this software is not a simple task, and it’s important to have an idea of the entire installation process ahead of time.
- **Install the prerequisites for the NEESpop distribution.** The hardest part about installing the NEESpop software isn’t the NEESpop installation itself, but installing the prerequisite software that NEESpop services need to run. Some of these components, such as the MySQL database, will probably have already been installed in your machine as part of a standard RedHat install. Others, such as Jakarta Ant, you will need to download separately. Instructions on how to install such components are linked from the appropriate step in the NEESpop installation instructions.
- **Download and unpack the NEESpop distribution.** This is a gigantic tarball, nearly two hundred megabytes in packed form. To unpack the tarball, type

```
tar xzf neesgrid-2.2-neespop.tar.gz
```

(although you may have a different version number than 2.2). This will expand the compressed file.
- **Invoke the NEESpop configuration script.** Type

```
./configure
```

at the root prompt. The computer will ask you a series of questions about host names, URLs, and such. You will need to know the username and password for a MySQL user capable of granting access to other MySQL users. (If you want a more comprehensive set of questions to handle special cases, use

```
./configure --prompt
```

instead; in most cases this won’t be necessary.)
- **Invoke the NEESpop install script.** Type

```
./install
```

at the root prompt. The installation will take roughly half an hour to forty-five minutes, depending on the speed of the NEESpop computer. The installation will start by asking you a series of questions, such as the URL for the TPM machine.

If the installation doesn't go right, chances are that you've missed installing one of the prerequisites. In this case, you should follow the instructions on how to remove a NEESpop installation, found at the end of the installation document, and then fix the problem and try again. It's important to do the uninstall or else the subsequent installation will fail.

- **Start NEESgrid services on the NEESpop.** Follow the installation directions for restarting the NEESgrid services. (The first time, of course, you won't be restarting them, but the procedure is pretty much the same.)
- **Set up grid security certificates on the NEESpop.** Using the account information you got in the first step above, use the `ncsa-cert-request` utility on your NEESpop to create *host* and *ldap* certificates for that machine, as described in the installation directions.
- **Set up "fake DAQ."** The "fake DAQ" is a test utility that simulates a DAQ machine.
- **Register with Big Brother.** "Big Brother" is a grid-based system monitor used to report on whether the various components of your system are up and running. To have your NEESpop included among the others listed there, register it by email by following the installation directions.
- **Configure the NSDS channels.** For the test described in the next step, this won't be necessary, because you've already set up the Fake DAQ.
- **Check the CHEF installation from a web browser.**
 - Open up a web browser, and point it to the URL of the NEESpop you've just installed. A page should come up with the NEESgrid logo and giving you the option of creating a new login.
 - Create the login, click on the button labeled **Membership** (one of the stack of buttons along the left side of the screen), check the "NEES Sample" checkbox, and click the **Join** button. A new tab labeled **NEES** should appear near the NEESgrid logo.
 - Click on that tab, and then the **Dataviewer** button on the left side of the window.
 - In the **Experiment Browser** area of the screen, select **Local Streaming Server (test data)** and then hit the **Continue** button.
 - In the **Current Layout** area of the next page, click the **Add X-Y Graph** button to go to a page for defining graphs.
 - Change the **Y-Axis** dropdown box to **Medium** and click the **Add View** button.
 - Back in the **Current Layout** area, click on the **View Current Layout** button.

After a few moments you should see a graph. (You may need to download a Java Runtime Environment installation into your browser first; your browser will tell you if this is necessary.) After a few seconds more the graph should turn into a moving sine wave. If it does, then your CHEF installation checks out okay.

5.2. *Installing the MatLab Computational Node*

Instructions on installing the Three-Site Simulation onto the computational node are available at <http://neesgrid.org/documents/Release2.2> under the heading “Matlab NTCP Toolbox and Examples”. You are strongly recommended to read through this sequence of pages *before* you get out your screwdriver.

5.3. *Running the Three-Site Simulation*

Complete instructions for performing the Three-Site Simulation are also available at at <http://neesgrid.org/documents/Release2.2> under the heading “Matlab NTCP Toolbox and Examples”.

6. Requirements for Mini-MOST

6.1. *The DAQ PC*

The Mini-MOST system requires a dedicated PC to act as the interface between the instrumented model and the NEESpop.

The DAQ PC needs:

- LabVIEW 6.1
- LabVIEW Internet Toolkit
- A Labview-supported data acquisition board
- Windows XP/2000/NT
- Recommended: At least 512MB of memory, and fast disk drive(s)
- Unzip program, such as WinZip or pkzip, to unpack the distribution

6.2. *The TPM Server*

The TPM server should be at minimum a PC Intel P4 or better, with 512 Mb RAM, 80 Gb hard drive, 800 Mhz, with a 1Gb Ethernet Card.

The operating system should be a RedHat Linux 7.3 , 8.0, or 9.0 *Full Workstation Software* installation. The installation should automatically include the following, but you should double check their existence after the installation just to be sure:

- Apache httpd Version 1.3.27 or 2.0.x
- Perl 5.x with Perl CGI.pm
- Imagick/convert graphics utility
- curl
- wget
- gcc & associated libraries

The following ports must be open: 80, 443, 8900, 22.

7. Constructing the Mini-MOST Device

The Mini-MOST hardware consists of machined parts, sensors, a stepper motor, and NI DAQ equipment.

Construction plans, wiring diagrams, and drawings are all available at <http://www.neesgrid.org/mini-most>. A components list is included as an appendix to the document “System Requirements of Mini-MOST Experiment,” which is available for download there.

8. Setting up the TPM Server (if any)

- **Set up the camera.** Follow the manufacturer’s instructions and record the IP address through which the camera is addressed.
- **Login to the TPM computer as root.**
- **Download the TPM installation instructions and package.** These are available on the same site as the NEESpop installation instructions.
- **Read the installaton instructions.** You’ll be glad you did.
- **Run the configuration scripts.** The TPM installation instructions list a number of perl scripts to be run. The configuration script will ask you for configuration information, and then output some configuration files to be used by the other installation scripts.
- **Verify the set up from a web browser.** If you go into the **NEES** tab on the NEESpop CHEF set-up, as described in the last point of the NEESpop section, you’ll see a **Video Cameras** button. If the TPM machine has been configured properly, you should be able to see the output of the video camera(s) on your web.

9. Installing the DAQ Software

Full information about installing and running the Mini-MOST LabView code is available at <http://www-unix.mcs.anl.gov/~hubbard/neesgrid/mmmost/>.

- **Login to the DAQ computer with administrator privileges.**
- **Install LabView and LabView Internet Toolkit.** Note that LabView isn’t an absolute requirement, since the DAQ software has an API you can do to directly. However, the reference implementation uses LabView, and you may find that to be a good model to work from.
- **Download the DAQ installation instructions and package.** These are available on the same site as the NEESpop installation instructions.
- **Configure the DAQ data channels.** A DAQ channel represents information from a single sensor or instrument; a DAQ channel needs to be defined for each instrument you want to read from. If you’re using National Instruments DAQ equipment, you can configure the channels using their MAX (Measurement and Automation Explorer) utility.
- **Initialize the DAQ board.** This can also be done with MAX.

10. Running the Mini-MOST Experiment

Running the actual Mini-MOST Experiment is slightly different than running the Three-Site Simulation. Here are a few notes on similarities and differences.

- The NTCP configuration is the same for both experiments.
- Only three Matlab windows need to be opened, since there is no Matlab simulation of the UCol substructure.
- Three NTCP servers need to be started in both cases, and they are started the same way.
- The number of experiment steps can be configured by editing `Sim_Coordinator3Sites` the same way you may have done for the Three-Site Simulation.

The basic sequence for running the Mini-MOST experiment goes as follows:

- Start up the DAQ program. See <http://www-unix.mcs.anl.gov/~hubbard/neesgrid/mmost/> for full instructions on running the Mini-MOST LabView DAQ code.
- Start up the three NTCP servers on the NEESpop.
- Start the following three Matlab programs from the same directory you used to run the Three-Site Simulation:
 - `NCSA_comp_site.m`
 - `UIUC_exp_site.m`
 - `Sim_Coordinator3Sites.m`

The experiment should then run to completion similarly to the Three-Site Simulation, with the exception of the clanking of the Mini-MOST hardware.

11. Glossary

Some of the things listed in this glossary you won't have to deal with directly to perform the reference experiments, but they are included here in case you run across a new NEESgrid acronym and don't know what it means.

API: abbreviation for “application programming interface,” the interface (calling conventions) by which an application accesses services. Several APIs are defined and documented within NEESgrid. For instance, the NTCP services communicate with an NTCP control plugin according to a certain API; the plugin is written with this API in mind, or else it won't be able to talk to the NTCP services.

Big Brother: a system-wide NEESgrid “health monitor” that periodically queries NEESpops and reports, in a simple table, which ones are up and running and which services are available. The Big Brother report is available through CHEF.

CHEF/Collaboratory Tools: A collection of collaborative tools along users to communicate with other researchers in an experiment, to observe experiments remotely using streamed video and data, and to browse data and information in local and global data repositories. Each NEESpop includes a CHEF installation that produces the web pages you see when you connect to an equipment site with your browser.

Computational Node: Any computer being used for number crunching on a network. In the context of the Three-Site Simulation and Mini-MOST, the computational node is the machine running the Matlab programs.

Credential/Certificate: A file, used by grid security routines, that proves you are who you say you are. Once you have an Alliance account, you can request a certificate that allows you to use secured NEESgrid services.

Data Channel: one particular stream of data, such as the data from one particular load cell during an experiment.

Data Repository: one of two places where NEESgrid data is held. A **local repository** holds information at an equipment site as part of the NEESpop services at that site. There is also a global repository, representing a centralized location where all NEES experiment data can be uploaded for use by other researchers.

DAQ: Short for “data acquisition.” In the context of NEESgrid, it usually refers to a dedicated Windows XP machine acting as an interface between the instrumented experiment and the NEESpop, or more specifically the software running on that machine.

DAQ Board: The data capture interface hardware. The DAQ machine communicates with it, and it is in turn connected to the Mini-MOST hardware.

DAQ Driver: In the provided code, the driver is a simple set of TCP pipes on the NEESpop used to pass information from the DAQ code and DAQ Server Daemon to NSDS.

DAQ Server Daemon: A standalone LabVIEW program that is running in the background on the DAQ machine. Its job is to set up the TCP connections, handle the control channel, and set global variables for the library code.

E-Notebook: a web-based electronic lab notebook, accessible through CHEF, that lets users store, review, and share text, images, and documents.

Fake DAQ: a service installed on NEESpop that simulates a DAQ machine outputting sine waves. This can be used to test data streaming capabilities as you are getting your system set up.

Global Repository: see Data Repository.

Globus Toolkit (GT): a collection of grid infrastructure services developed by the Globus consortium (www.globus.org).

GridFTP: a grid-secure file transfer utility used to move data from the local NEESpop to another machine, such as that of a remote researcher. This capability is accessible through CHEF.

LabView: a commercial program from National Instruments (www.ni.com/labview) for data acquisition. It is used as the basis of the DAQ software in the Mini-MOST experiment.

LabView Plugin: a code plugin for the NTCP server that translates NTCP instructions and information into a simple ASCII format which the Labview DAQ software can understand.

Local Repository: see Data Repository.

MAX: the Measurement and Automation Explorer, software from National Instruments. It is run on the DAQ machine. You will use MAX to set up data channels from the DAQ machine for Mini-MOST, and to initialize the DAQ board.

Matlab: A commercial product of the Mathworks (www.mathworks.com/matlab), a programming language and interpreter used for numerical calculation and simulation. Both the Three-Site Simulation and Mini-MOST use Matlab programs to simulate components of MOST.

Metadata: data *about* data. In NEESgrid, metadata means any information that NEESgrid services can store that isn't purely instrument data. Metadata is needed to describe the experimental setup in order to let the system know *what* the data in the data repository represents.

Mini-MOST: A miniaturized version of the MOST experiment, developed to act as a reference implementation of a NEESgrid experiment, and to act as a training aid.

MOST: The Multisite Online Simulation Test, the first major experiment performed using NEES, which took place in July, 2003. For more, see www.neesgrid.org/most.

MPlugin (Matlab Plugin): a code plugin for the NTCP server similar to the LabView plugin but allowing communication between NTCP and Matlab.

MyProxy: A security service that lets you have controlled access to the services on a NEESgrid computer without generating a security credential on that machine. You won't need to know the details.

NEESpop (NEES point-of-presence): either a core collection of NEESgrid services, or the machine that's running them. The NEESpop is the main point of net contact between an equipment site and remote users.

NFMS: NEESgrid File Management Services. NFMS provides clients with the ability to locate files independently of how and where they are stored, as well as the ability to negotiate transactions with storage systems. Using NFMS, a client can reliably and securely store and retrieve data from a NEESgrid repository without having to keep track of the physical location of the file or use any API's specific to any particular storage architecture. NFMS is designed to accommodate a variety of storage architectures and data transfer protocols, including GridFTP.

NMDS: NEESgrid Metadata Services. These services let other NEESgrid software create, manage, and retrieve metadata objects in a data repository. The equipment sites probably won't need to dig into this part of the NEESgrid infrastructure.

NSDS: The NEESgrid Streaming Data Server, which allows data collected by the DAQ machine to be streamed live to a remote observer.

NSDS Driver API: This API defines the interface between the generic NEESgrid streaming data services and the equipment-specific streaming data drivers needed to enable streaming data. An NSDS driver is already written in LabView for the Mini-MOST.

NTCP: The NTCP service provides a common protocol (the NEESgrid Teleoperations Control Protocol) that can be used by remote applications to interact with a running experiment over NEESgrid. Components include the core NTCP server, which, when used with a "**NTCP control plugin**", is used to communicate with a backend control system (or simulation). NTCP is used to perform remote operation of physical experiments or simulations; it is documented in *NEESgrid Technical Report TR-2003-07*.

NTCP Control Plugin: software written by the equipment site to act as an interface between the NTCP services and the individual pieces of equipment being controlled. These plugins can be written either using the **Control Plugin C API** or the **Control Plugin Java API**, depending on your preferences. The plugin architecture is described in *NEESgrid Technical Report TR-2003-16*; a sample plugin is documented in *NEESgrid Technical Report TR-2004-02*.

NTCP Helper: the NTCP Java client API, which provides an interface to the NTCP server.

NTCP Matlab Client Library: a set of Matlab functions that call the NTCP Java Client API to communicate with an NTCP server.

NTCP Matlab Backend Library: a set of Matlab functions that call the MPlugin Java client API to communicate with the NTCP server running on the local NEESpop.

NTCP Server: a service on NEESpop for handling NTCP; see NTCP.

Tomcat (or Jakarta Tomcat): an open-source system for generating web pages using Java. CHEF is built on top of this, and most equipment sites won't need to worry about it unless they want to add new collaborative tools to CHEF.

TPM System: Telepresence Management System. This is a dedicated computer used to control the streaming video cameras and broadcast the video to remote viewers. The Mini-MOST experiment allows for the option of using a TPM system.

X509: see Certificate.