



*Building the National Virtual Collaboratory
for Earthquake Engineering Research*

NEESgrid

Technical Report NEESgrid-2003-14

www.neesgrid.org

(Draft Whitepaper Version: 0.1)

Last modified: June 26, 2003)

NSDS Java API

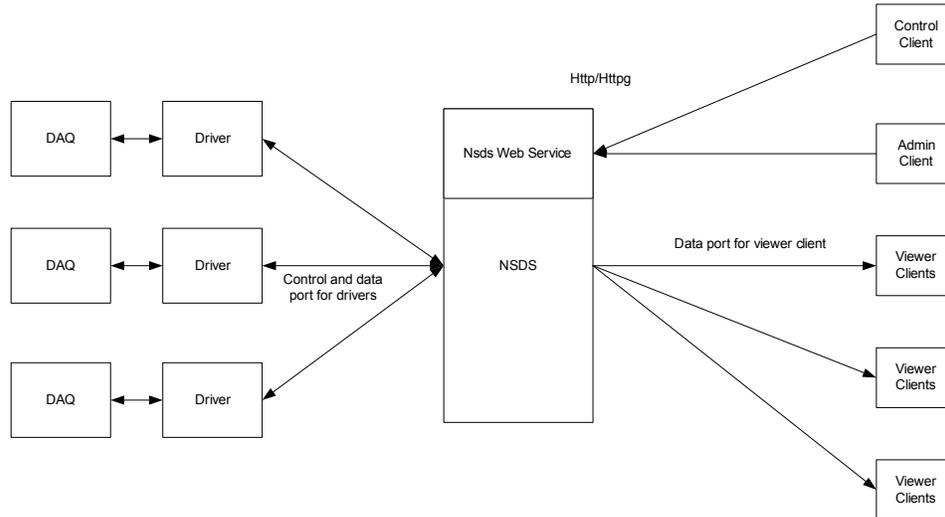
Gaurang Mehta¹

¹ University of Southern California Information Sciences Institute, Marina del Rey, CA 90292

Feedback on this document should be directed to gmehta@isi.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.

Summary: This document draft describes the NSDS Java API. It provides detailed component descriptions in list form for the NSDS web service interface and the driver interface/API. It also describes the NSDS defaults.



NSDS System Diagram

The NSDS is a server code written in java using the OGSA-TP3 toolkit. It requires jdk1.3.*, Jakarta-tomcat 4.1.x and ant.

The NSDS has 3 interfaces, the web-service interface for the client control channel, the TCP/IP interface for the client data channel, and the TCP/IP interface for the driver control and data channel.

NSDS Web Service Interface :

This interface allows clients(both admin and simple) to send control commands to the NSDS over http using soap. Both secure (using transport level GSI) as well as unsecure methods of communicating with the NSDS are available. The methods that can be invoked on the NSDS Webservice are:

1. Create Channel:

```
public String createchannel()
```

This returns a unique string which is used as an identifier by the client when it connects to the client data channel interface.

2. Subscribe channel

```
public boolean subscribe(String experiment, String sensor, String channelname)
```

This subscribes the client data channel identified by the channelname to the data coming from the experiment and sensor specified.

3. Unsubscribe channel

```
public boolean unsubscribe(String experiment, String sensor ,String channelname);
```

This unsubscribes the client data channel identified by the channelname from the data coming from the experiment and sensor specified.

4. Close Channel

```
public boolean closeChannel(String channelname);
```

This closes all the subscribed daq channels if any are open and then closes the data channel to the client.

5. Query Daq

```
public String queryDaq(String daqdriver);
```

This returns the status of driver specified by daqdriver (need to change this to instead take the experiment name and sensor as parameters).
Returns streaming or stopped as the reply.

Admin Methods :

6. Add Trial

```
public boolean addTrial(String trial, String sensor, String driverid, String daq_port);
```

This adds a line in the nsdsdriverinfo file which maps a given experiment(trial) and sensor to a driver and daq_port(daq-channel)

7. Remove Trial

```
public boolean removeTrial(String trial, String sensor);
```

This removes the entry in the nsdsdriverinfofile for the given experiment(trial) and sensor.

8. Flush Trial

```
public boolean flushTrial(String trial);
```

This removes all entries from the nsdsdriverinfofile for the give experiment(trial).

9. Flush File

```
public boolean flushFile();
```

This removes all the entries from the nsdsdriverinfofile

Driver Interface/Api

Currently the drivers interface with the NSDS via TCP by opening a control and a data socket. Following is the protocol between the driver and the NSDS.

1. Open Port

In order to receive the daq data to the nsds the nsds sends “open-port <daq-channel identifier>” command to the driver over the control port. The driver in response set ups the communication between the daq and the driver and starts streaming data for that channel to the NSDS over the data connection. It also sends an acknowledgment to the NSDS on the control connection saying “Streaming data on port <daq channel id>”

2. Close Port

To stop the streaming data from the Daq the NSDS sends a “close-port < daq-channel identifier >” command to the driver over the control port. The driver responds by stopping sending data on the data connection to the NSDS and acknowledges the closing of the streaming on the control connection by saying “Closing data on port < daq-channel id>”

3. Daq Status

This returns the status of the driver and the daq (replies are either streaming or stopped). The command sent to the driver is “daq-status”.

Currently the only way to interface a driver is to speak this ascii protocols and connect over the network. In the next version we will have support for a pluggable module interface where in you can provide a api to which the nsds will make a call.

NSDS defaults.

The nsds uses a configuration file located or symlinked from /etc/nsds.conf.
The default port created for the client to connect to the nsds for data is 41410

The default port created for the driver control channel is 42420 and data channel is 42421

The mapping of logical experiment names and sensors to the drivers and the daq-channel is kept in a file which defaults to /tmp/nsdsdriverinfo.conf

All these defaults can be changed from the /etc/nsds.conf file.