Doc. Number:	15720	Approvals		
Revision:	1		Signature	Date
FileName:	15720.doc	Engineering:		
Title:	Streaming to NAQS with a Taurus			
Originator:	M. Turcot			
Date:	Feb. 14 2005			

Revision History

Rev.	Date	Author	Description
1	Feb. 14 2005	M. Turcot	Initial Revision

STREAMING TO NAQS WITH A TAURUS

Note: This document assumes that the user is familiar with the use and configuration of NaqsServer and its associated configuration files. Please note that streaming data from a Taurus requires the latest versions of NAQS, DLLs, Java, and all associated software. This procedure was designed for an acquisition system installed on a Windows PC. Always backup your files before modifying them.

Step 1: Acquisition System Setup

1. Copy the NpToNmxp.jar and NpToNmxp.bat to your \nmx\bin directory.

2. Copy the arp-light.jar to c:\Program Files\Java\j2re1.4.2_04\lib\ext where j2re1.4.2_04 is the latest jre that is installed on your machine. If you are running a newer or older version of the java runtime environment the directory names maybe different.

3. Open the Naqs station file. (\nmx\user\naqs.stn)

4. Create a new Instrument Prototype for the Taurus. Please note that the SOH Bundles per packet is set to 27.

```
[ InstrumentPrototype ]     // predefined instrument - all fields mandatory
TypeName = Taurus           // prototype name - may be same as model
Model = Taurus              // instrument type
MemoryKB  = 512             // instrument ReTx buffer size
SohBundlesPerPacket = 27    // bundles per soh packet
RequestInterval = 60       // interval between retx request messages
SohChannelName = SOH        // extension for soh file (NUL if none)
```

```
SohBufferSize = 1           // file size in MB
SohBufferPath = RingBuff    // where files are located
InetHostName = Dynamic      // return IP address for instrument
InetPort = 32000            // return IP port for instrument
```

5. Create the Channel Prototypes for the Taurus. Again, please note the Bundles per packet is set to 27. Make certain to modify the Channel Prototypes to reflect the settings of your system (i.e. sensor, ringbuffer path, ringbuffer size).

```
[ ChannelPrototype ]        // predefined channel - all fields mandatory
TypeName = BHZ-3            // label for this type
Name = BHZ                  // channel name
Component = 1               // digitizer component (refers to current
instrument)
Sensor = Trillium           // pointer to predefined [Sensor] characteristics
Azimuth = 90                // azimuth in degrees clockwise from North
Dip = 0                     // dip in degrees (positive down)
Depth = 0                   // has to be defined for each channel
BundlesPerPacket = 27       // bundles per data packet
RingBufferSize = 10        // file size in MB
RingBufferPath = ringbuff  // where files are located
ResponseFile = none         // name of SEED response file

[ ChannelPrototype ]        // predefined channel - all fields mandatory
TypeName = BHN-3            // label for this type
Name = BHN                  // channel name
Component = 2               // digitizer component (refers to current
instrument)
Sensor = Trillium           // pointer to predefined [Sensor] characteristics
Azimuth = 0                 // azimuth in degrees clockwise from North
Dip = 0                     // dip in degrees (positive down)
Depth = 0                   // has to be defined for each channel
BundlesPerPacket = 27       // bundles per data packet
RingBufferSize = 10        // file size in MB
RingBufferPath = ringbuff  // where files are located
ResponseFile = none         // name of SEED response file

[ ChannelPrototype ]        // predefined channel - all fields mandatory
TypeName = BHE-3            // label for this type
Name = BHE                  // channel name
Component = 3               // digitizer component (refers to current
instrument)
Sensor = Trillium           // pointer to predefined [Sensor] characteristics
Azimuth = 90                // azimuth in degrees clockwise from North
Dip = 0                     // dip in degrees (positive down)
Depth = 0                   // has to be defined for each channel
BundlesPerPacket = 27       // bundles per data packet
RingBufferSize = 10        // file size in MB
RingBufferPath = ringbuff  // where files are located
ResponseFile = none         // name of SEED response file
```

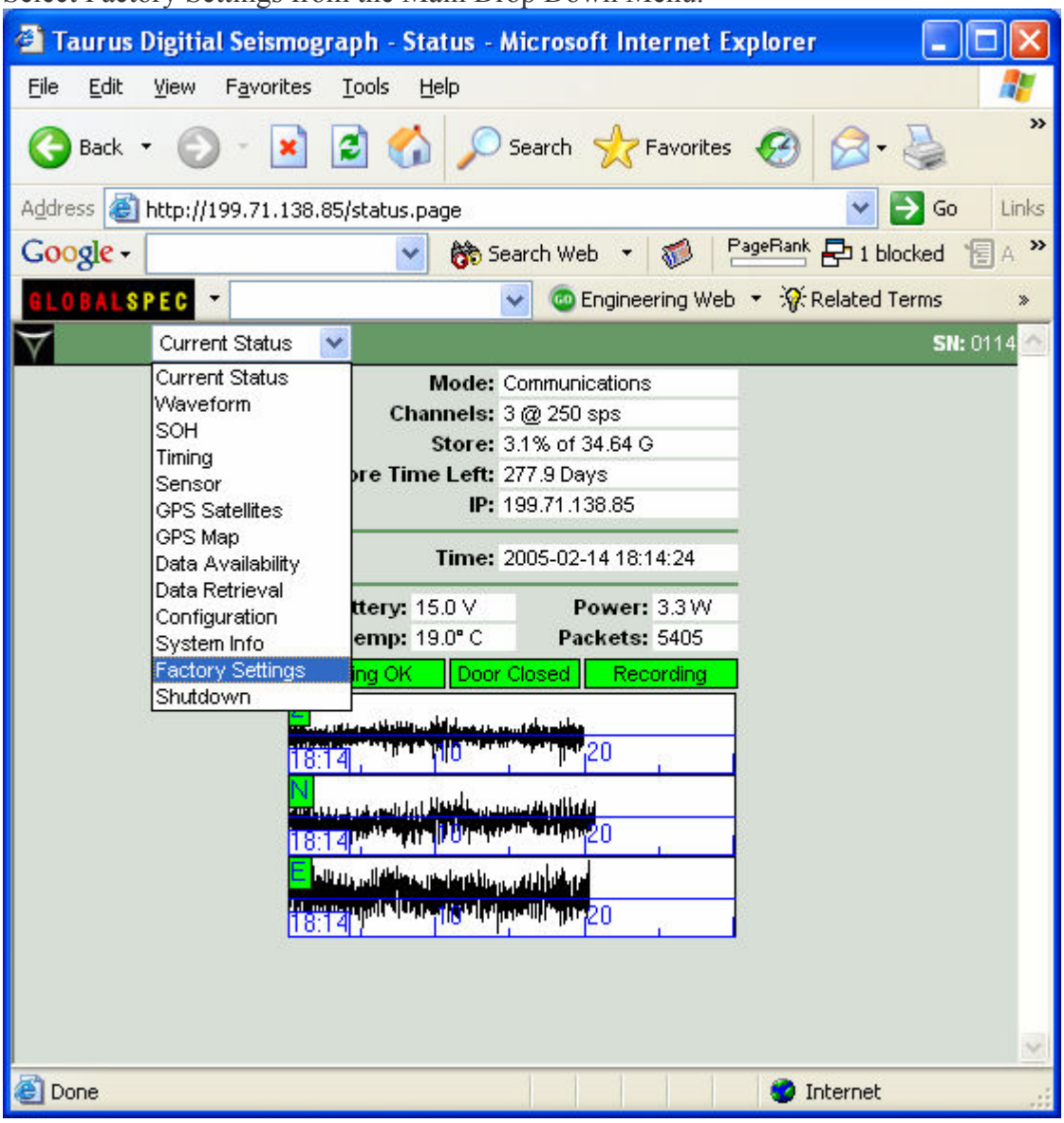
6. Create the Station Prototype. Remember to change the SerialNumber field to reflect the serial number of your Taurus.

```
[ Station ]
Name = STN01
Description = Top of hill, new vault
Latitude = 47.48
Longitude = 16.36
Elevation = 1022.3

// These are the instruments associated with the preceding station

[ Instrument ]              // instance of an instrument
Prototype = Taurus             // instrument type
SerialNumber = 0            // serial number - mandatory

// These are the channels associated with the preceding instrument

[ Channel ]                 // instance of a channel
Prototype = BHZ-3           // use settings from this prototype

[ Channel ]                 // instance of a channel
Prototype = BHN-3           // use settings from this prototype

[ Channel ]                 // instance of a channel
Prototype = BHE-3           // use settings from this prototype
```

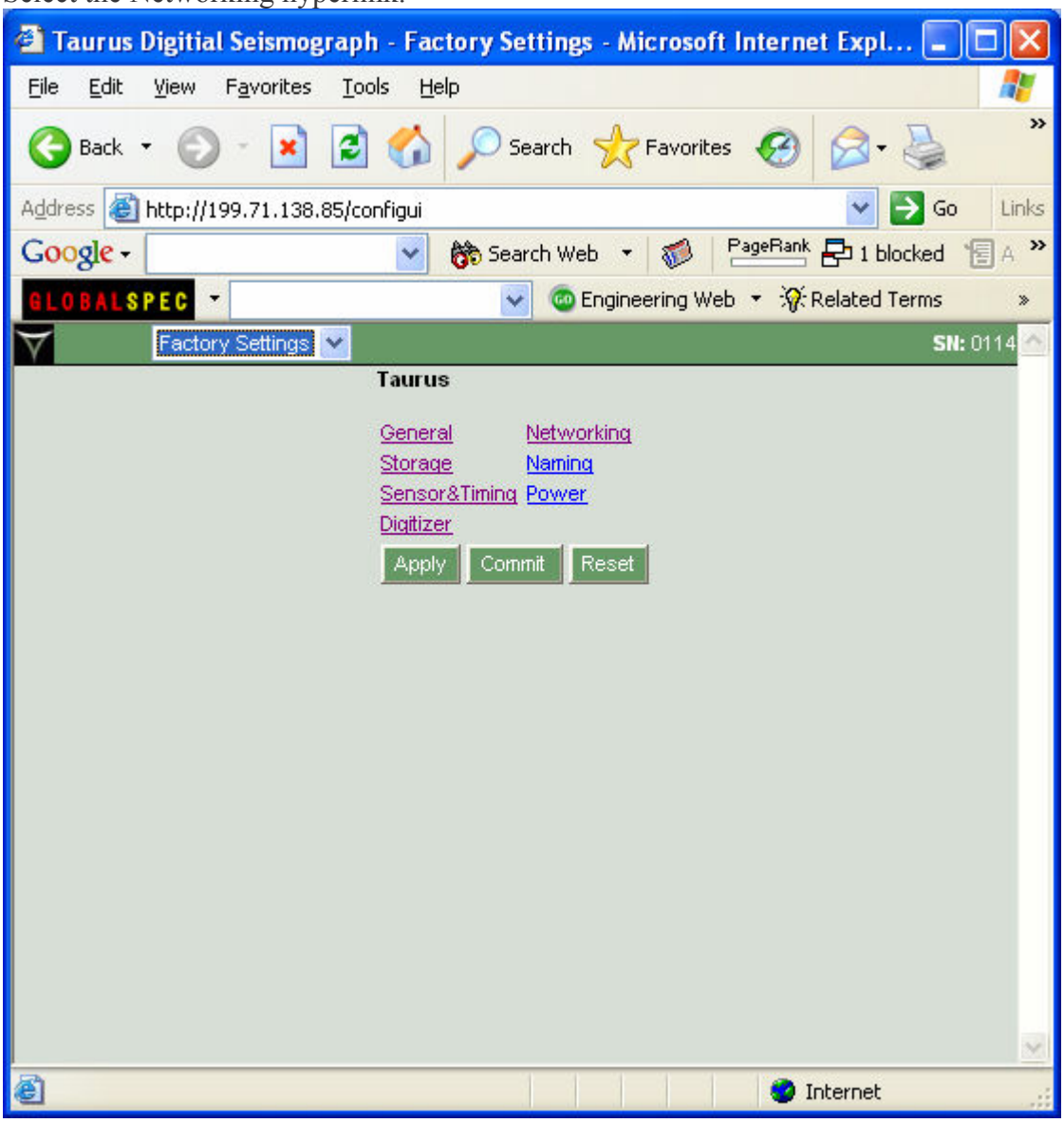
Step 2: Taurus Configuration

1. Log on to the Taurus using the integrated user interface or by using an external browser. (See sections 3.1.2 and 3.1.4 respectively of the Taurus manual on how to log on.)

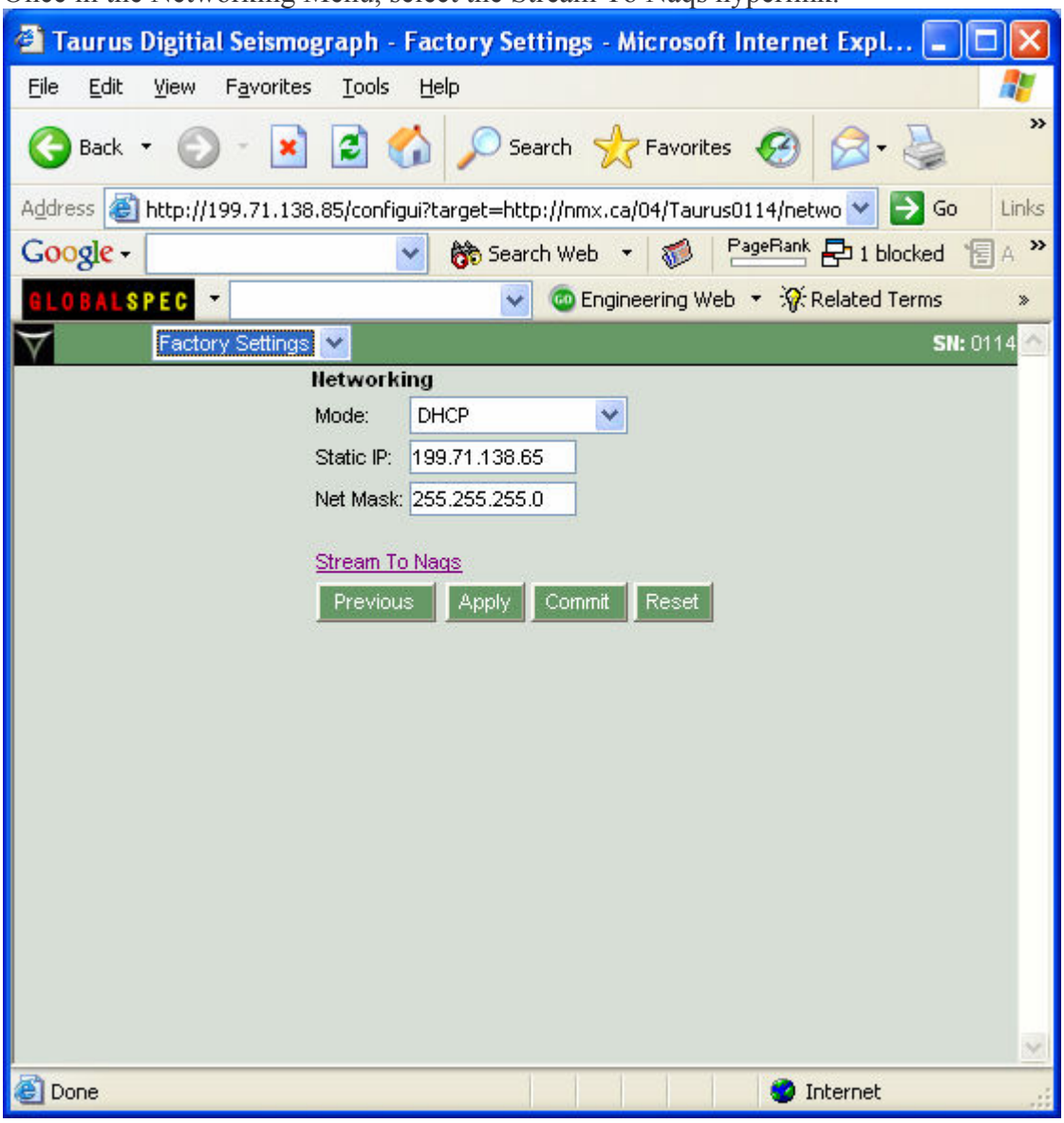
2. Select Factory Settings from the Main Drop Down Menu.



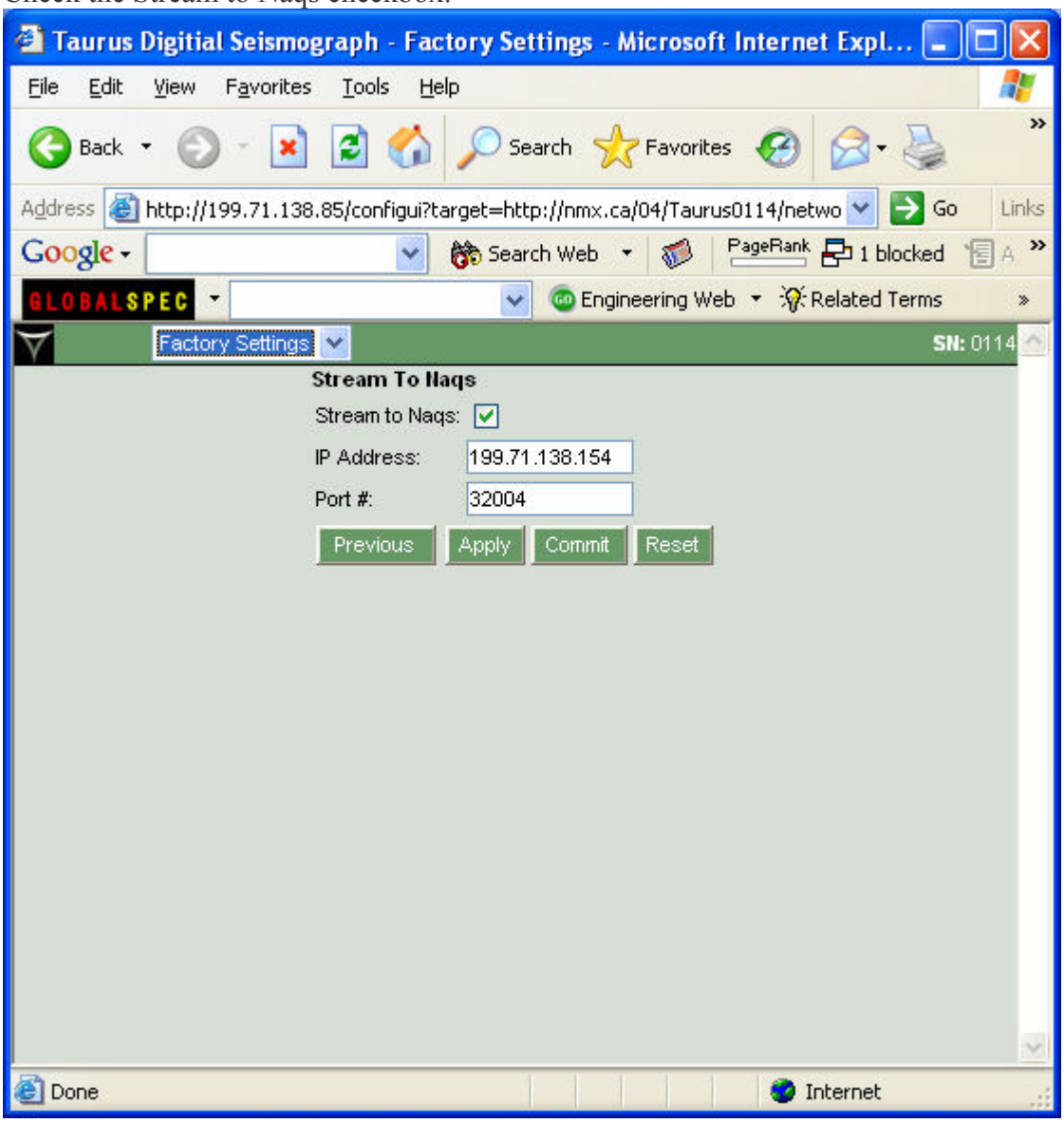
3. Select the Networking hyperlink.



4. Once in the Networking Menu, select the Stream To Naqs hyperlink.



5. Check the Stream to Naqs checkbox.



6. Input the unicast IP address of the Naqs computer or the multicast IP address as indicated in the Naqs.ini file.

7. Input the Port # from which NpToNmxp will be receiving UDP data, usually 32004. This port should **not** be the same as the port setup in the Naqs.ini file (Port = 32000).

8. Apply and commit your new settings once you have finished.

Step 3: Receiving Data

1. Restart NaqsServer. If you have nmx watchdog running, simply close the Naqs window and watchdog will restart the Naqs. If you are using the command prompt, go to the \nmx\user directory and type *naqsserver* to start the software.

2. Start NpToNmxp. If you are using the command prompt, go to the \nmx\user directory and type *nptonmxp*. In the NpToNmxp window, the log will show messages like: "V 2005-01-26 14:03:37 NpClient…….(5) Oldest SeqNumber info not received yet. Caching packet: 55228: Taurus…" This will go on as long as a minute before the first oldest sequence number is received.

Note:

The NpToNmxp program can be integrated into the nmx watchdog program using the following entry added to the watchdog.ini:

```
[ WatchEntry 4 ]
ProgramTitle = "NpToNmxp"
ProgramPathname = "java -Xrs -cp c:\nmx\bin\NpToNmxp.jar
ca.nanometrics.npToNmxp.NpToNmxp"
WorkingDirectory = "c:\nmx\user"
ExitAction = Restart
PingsSemaphore = TRUE
StartDelay = 6s
```

Increment the NumberEntries parameter in the top block by 1. Edit the WatchEntry parameter to reflect the entry number in the watchdog.ini file (The above example indicates that this is the fourth entry). Be extra prudent when modifying the watchdog.ini file as it is case and space sensitive.

Restart your watchdog.