



*Building the National Virtual Collaboratory  
for Earthquake Engineering Research*

**NEESgrid**

**Technical Report NEESgrid-2004-31**

**[www.neesgrid.org](http://www.neesgrid.org)**

Draft Whitepaper Version: 1.0.2  
Last modified: September 9, 2004

**Connecting cameras to the Data Turbine**

**Paul Hubbard<sup>1</sup>**

<sup>1</sup> Argonne National Laboratory

Feedback on this document should be directed to [hubbard@mcs.anl.gov](mailto:hubbard@mcs.anl.gov)

---

**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. Summary .....	2
2. Revision History.....	2
3. Introduction – the Role of the Data Turbine in NEESGrid .....	3
4. Video versus image data: What’s the difference?.....	3
5. What can I do once the images are in the turbine?.....	3
6. What about audio?.....	4
7. What about Pan-Tilt-Zoom (PTZ) heads? .....	4
8. What about shutter control for jcamera? .....	4
9. Supported image sources .....	4
10. Increasing the amount of memory for the turbine .....	5
11. Code Access, Mailing List and Bugzilla Archive .....	5
12. Testing, Benchmarking and Profiling.....	6
13. How to Get Started .....	6

**1. Summary**

This report documents how to interface a digital camera, frame grabber or telepresence system into the Data Turbine, and thence NEESGrid. Also covered are uses of the live data, debugging tools, and interface methods.

**2. Revision History**

Initial release: September 2, 2004.

1.0.1: Added JPG-via-URL.

1.0.2: Added JVM memory, note about Tomcat and WebDAV.

### 3. Introduction – the Role of the Data Turbine in NEESGrid

The Data Turbine, sometimes referred to as ‘RBNB’ for Ring-Buffered Network Bus, is a commercial Java-based program that handles data streaming, archiving, synchronization, plugins, and many other features. It’s free for non-commercial use, and we’ve negotiated and OEM redistribution license and some enhancements as well. It’s a replacement for the NSDS in v3.0.

Due to its power and flexibility, it continues to assume a larger role in the NEESGrid code. For example, we now have a program that will assemble video into a DAQ-synchronized Quicktime movie based on DAQ events!

### 4. Video versus image data: What’s the difference?

To the data turbine, all data is a discrete series of time-stamped items. In our case, those items (aka datum, plural data) are JPEG images. Since a series of single-frame digital images resembles a video telepresence stream once they’re inside the turbine, this document has to clarify the terminology a bit.

Here, video refers to a series of images, captured at a constant sample rate. For example, a QCIF stream (320x240) at 30 frames per second (fps) would be considered a video stream. A 5megapixel source, capturing every 10-30 seconds, would not.

The distinction is purely semantic. The differences are in the resolution and frame rate. However, once the data is in the turbine, there’s no other difference.

### 5. What can I do once the images are in the turbine?

There are a wide variety of things you can do with the image data once its in the turbine. If you find another innovative use, please share it with us!

All of these are documented at <http://www.mcs.anl.gov/neesgrid/rbnb-apps.html>

1. Save to disk. The DataVideoSink program will write images and numerical data to permanent storage.
2. Generate lower-resolution thumbnails, which are also Turbine channels. The JPEGThumbnailer program will do this for you. Quite useful for remote monitoring of high-res or high-framerate channels.
3. View, VCR-style, from anywhere. Creare’s rbnbjview.jar program, bundled with the turbine, allows you to view many channels at once in a single window. You can mix full channels, thumbnails, any JPEG source in the turbine.
4. Write your own image-processing plugin that runs with the turbine. For example, you could write a simple plugin that uses image processing to determine if your sample has broken or fallen over during the night. Since these can run on your server, more elaborate algorithms are very possible.

5. Generate DAQ-synchronized movies. Very popular, and quite useful for any experiment or demo that runs slowly.
6. Stream out video for remote participants. Particularly if you use the inherent distribution abilities of the Turbine (server replication, parent-child) you can provide video feeds to many participants without slowing down your experiment.
7. Pull single-frame images in a web page via URL fetch. See <http://rbnb.creare.com/rbnb/rbnbVideo.html> for examples and URL syntax. You may need to modify the NEES startup script for this, as the WebDAV interface requires that the turbine be started from Tomcat.

## 6. What about audio?

One of the items that is being worked on is audio support for the turbine. A previous distribution included demonstration programs for this, but the support was subsequently removed. Part of our contract with Create includes audio, so its return is only a question of time. Adding audio makes many other applications possible: telepresence, audio capture and recording for experiments, etc.

## 7. What about Pan-Tilt-Zoom (PTZ) heads?

I have preliminary code that controls a PTZ head connected to any Axis unit; this code is pre-alpha and not ready to release. Keep an eye out for beta releases.

## 8. What about shutter control for jcamera?

This is trickier, since Jcamera is a first-class NTCP application. To make this work inside a web application, we have to convert the NTCP client into a Java applet. We're working on the idea, but we don't yet know if it's possible to do.

## 9. Supported image sources

There are several disparate methods for getting JPEGs into the turbine. Here are the ones supported as of 9/2004:

1. JMF-supported video capture. The Java media framework can use any video capture source supported by Video for Windows drivers. In practice, we've used BT878 and 848 cards successfully. To do this, you use the rbnbjpeg.jar distributed with the Turbine; docs are included in the distribution. This is potentially a large class of sources, as most video sources include VfW drivers!
2. Axis video server. Based on code from Jason Hanley, AxisSource grabs frames from any Axis unit over the network. Well tested and quite stable.
3. FlexTPS server. The FlexTpsSource program listens to the MJPEG source in FlexTPS, meaning that you can tap any public or private streams defined therein.
4. PictureSource is a simple program that will repeatedly upload a series of images from disk into the turbine. Useful for remote debugging/testing, or perhaps if your source writes into share disk space.

5. jcamera. <http://www.mcs.anl.gov/neesgrid/jcamera> is an NTCP-controlled high-res (5 megapixels or more, depending on the camera) source that uses a consumer-grade camera as a source. Low frame rate (best case is 1 frame every 10-15 seconds) but very high resolution, and multiple cameras are supported.

These sources connect to the turbine in a wide variety of ways: PCI bus, TCP/IP, etc.

## 10. Increasing the amount of memory for the turbine

By default, Java gives 256MB of memory to the virtual machine running the turbine. While more than enough for most java programs, it's often useful to increase this to allow for more channels and longer time histories. To do so, edit

```
/usr/local/nees/etc/rc.d/init.d/rbnb
```

and change

```
java -jar lib/rbnb.jar &
```

to

```
java -Xmx1024M -jar lib/rbnb.jar &
```

Or whatever you want; this example allocates 1024MB (1GB) to the Turbine. We've used up to 1500MB on a 2GB machine with no problems.

You'll have to restart the turbine for the change to take effect.

## 11. Code Access, Mailing List and Bugzilla Archive

CVS access to the NEESgrid code archive is documented at <http://www.mcs.anl.gov/neesgrid/cvs.html>. A Bugzilla archive for logging and tracking bugs and feature requests is available at <http://bugzilla.ncsa.uiuc.edu/neesgrid>.

More information about our use of the Data Turbine can be found at <http://www.mcs.anl.gov/neesgrid/turbine/> and <http://www.mcs.anl.gov/neesgrid/dtpatterns/>

## 12. Testing, Benchmarking and Profiling

- a) RBNB Admin is a good way to view what channels are present in the Turbine, show listeners, and so forth.
- b) RBNB Plot can also show live server statistics
- c) It's easy to find channel names with Admin, and then test them with rbnbjview.
- d) For fast testing, the URL syntax is very useful.
- e) On the server running the turbine and/or applications, Unix tools such as top and ps are useful to find CPU usage.

## 13. How to Get Started

Given the variety of possible image sources, getting started is a matter of budget and priorities. Probably the easiest thing to do is

1. Download the 'turbine' package from NEESGrid CVS (<http://www.mcs.anl.gov/neesgrid/cvs.html>) and get it compiled
2. Download the data turbine from <http://outlet.creare.com/> and start up a server
3. Run PictureSource to get some images into the turbine
4. Run rbnbjview or your browser to view them

This gets you into the milieu at zero cost and minimal time. You can then try other image sources, such as Axis or FlexTPSSource, to see how they work.

Next, I'd suggest looking at the thumbnailer to get an idea of the possibilities.

Have fun!