

Connecting to an RBNB DataTurbine using Python via WebDAV

1. Overview

The RBNB DataTurbine provides a WebDAV service for interacting with the server. This document covers the software dependencies needed to interact with the DataTurbine via the WebDAV protocol and gives an example of a WebDAVSource class that loads tchain data into the DataTurbine.

2. Dependencies

2.1 Download and install pycurl

If you don't already have the pycurl module on your system, download and install it from <http://pycurl.sourceforge.net/>. You can also use easy_install directly using:

```
easy_install http://pycurl.sourceforge.net/download/pycurl-7.19.0.tar.gz
```

2.3 Download and install DataTurbine

The RBNB DataTurbine software runs on Mac OS X, Windows, and Linux, and can be downloaded from <http://code.google.com/p/dataturbine>. The current version is 3.2b4. Double click on the downloaded .jar file to install it.

2.4 Download the BBL software repository

The python [webdav example driver](#) is maintained in the BBL software repository located at <https://bbl.ancl.hawaii.edu/projects/bbl>. You can use a Subversion client to checkout a working copy of the source code with a Subversion client ([TortoiseSVN](#) (Win), [SCPlugin](#) (Mac), or [eSVN](#) (Linux)), or via the command line:

```
svn checkout https://bbl.ancl.hawaii.edu/projects/bbl
```

You can also download a snapshot of the code as a zip file from <https://bbl.ancl.hawaii.edu/projects/bbl/trunk/kilonalu-0.5.0.zip>. Unzip the .zip file in the location of your choice.

3. WebDAVSource Example

3.1 Overview

The WebDAVSource driver is a bare-bones example of an instrument driver written in Python that communicates with the DataTurbine via the WebDAV protocol over HTTP. There is no exception handling, external configuration, or logging in order to keep the code brief. The webdav python module is within the 'instruments' package and the 'webdav' subpackage, but can be called standalone as well.

The webdav module can be run from the command line using the standard Python interpreter. The general flow of

execution is:

- webdav.py is run via Python.
- webdav.main() is called, which sets up initial variables.
- Using the set variables, webdav.main() creates an instance of WebDAVSource
- WebDAVSource.__init__() sets the class attributes, creates a source connection to the DataTurbine.
- execution is returned to webdav.main(), and it calls WebDAVSource.start() to start the data streaming.
- WebDAVSource.start() calls WebDAVSource.connect() to get a connection to the instrument. In the example, it opens a data file rather than a TCP socket.
- WebDAVSource.start() iterates through the lines of the file and calls WebDAVSource.insertData(), sending it a sample to insert.
- WebDAVSource.insertData() creates a timestamp, appends it to the sample string, and configures a cURL object with the source and channel metadata and the data sample. Samples are uploaded via HTTP PUT.
- Once all data lines are inserted, WebDAVSource.stop() is called, which in turn closes the data file with file.close() and closes the sample I/O stream with StringIO.close().
- With the connections closed, execution is returned to the webdav module and sys.exit() is called.

Note that the WebDAV calls are limited to:

1. WebDAVSource.__init__() for the connection and registration
2. WebDAVSource.insertData() to insert the data

3.2 Start the DataTurbine

Open a terminal (Mac or Linux) or a Command Prompt (Windows) and make a directory to store the ring buffered data, change directories to the RBNB installation program directory, and start the server. For instance:

For Mac:

```
mkdir /tmp/rbnb
cd /Applications/RBNB/V3.2B4/bin
java -jar rbnb.jar -H /tmp/rbnb -a localhost:3333
```

For Windows:

```
mkdir C:\temp\rbnb
cd "C:\Program Files\RBNB\V3.2B4\bin"
java -jar rbnb.jar -H C:\temp\rbnb -a localhost:3333
```

For Linux:

```
mkdir /tmp/rbnb
cd /usr/local/RBNB/V3.2B4/bin
java -jar rbnb.jar -H /tmp/rbnb -a localhost:3333
```

The server will start up both Apache Tomcat and the DataTurbine, and will wait for connections.

3.4 Run the WebDAVSource driver

Open another Terminal or Command Prompt and change directories to the webdav source code subpackage folder. Then, execute the webdav.py script using Python. For instance:

For Mac:

```
cd /Users/cjones/bbl/trunk/src/python/instruments/webdav
python webdav.py
```

For Windows:

```
cd C:\Users\cjones\bbl\trunk\src\python\instruments\webdav
python webdav.py
```

For Linux:

```
cd /home/cjones/bbl/trunk/src/python/instruments/webdav
python webdav.py
```

The Python code should run, connect to the DataTurbine, insert each sample line, and then exit cleanly. It is configured by default to load data from a data file with only ten lines. If you have any trouble, send me a note at csjones@hawaii.edu.