LoggerNetSource Software Design Document

Lawrence J. Miller

Cyberinfrastructure Laboratory for Environmental Observation Systems

6/3/08

San Diego Supercomputer Center

Revision Summary:

1. Date: May 21, 2008

Revision: 0.1

Changes: Initial version

Table of Contents

1. Introduction .................................................................................................................. ##

1.1 Goals and Requirements................................................................................. ##

1.2 Acronyms ....................................................................................................... ##

2. Design Overview ......................................................................................................... ##

2.1 System Overview ........................................................................................... ##

2.2 Problem Statement ......................................................................................... ##

3. Functional Design ........................................................................................................ ##

3.1 User Interfaces ............................................................................................... ##

3.2 Component management................................................................................ ##

3.3 Resource management ................................................................................... ##

3.4 System monitoring ......................................................................................... ##

4. Technical Design.......................................................................................................... ##

4.1 Components ................................................................................................... ##

4.2 Software API usage ........................................................................................ ##

4.3 Devices ........................................................................................................... ##

4.4 Device Internals ............................................................................................. ##

4.5 Device Configuration ..................................................................................... ##

4.6 Device Interfaces............................................................................................ ##

4.7 Device drivers................................................................................................. ##

4.5 Messages ........................................................................................................ ##

4.8 Configurations ................................................................................................ ##

4.9 Events ............................................................................................................. ##

4.10 Sampling ...................................................................................................... ##

4.11 Logging ........................................................................................................ ##

4.12 Error handling .............................................................................................. ##

4.13 Time ............................................................................................................. ##

4.14 GUI Tools .................................................................................................... ##

4.15 System Configuration and Management ...................................................... ##

5. General Issues .............................................................................................................. ##

6. Software Engineering Infrastructure ............................................................................ ##

6.1 Development practices ................................................................................... ##

6.2 Version control .............................................................................................. ##

6.3 Distribution mechanism ................................................................................. ##

1. INTRODUCTION

1.1 Goals and Requirements

This design is intended to produce a software program, called LoggerNetToRbnb, that will acquire real-time and archived data streams from a Campbell Scientific data logger via LoggerNet. This functionality will need to be automated and be triggered by the real-time acquisition of data. The data streams in LoggerNet and DataTurbine must be identical with respect to data values and time stamps of the data points.

1.2 Acronyms and Glossary

DAQ – data acquisition

RBNB – technical synonym for DataTurbine

DataTurbine Source Program – A DataTurbine utility application program that submits data streams to the DataTurbine ring buffer

API – Application Programmer Interface – an abstraction of the functionality of software that enables external use of and integration with the software

UML – Unified Modeling Language – used to draw schematics of interrelationships, especially object oriented relationships in software engineering.

OSDT – Open Source Data Turbine – an initiative sponsored by the National Science Foundation (NSF) to develop and promote the use of DataTurbine.

2. DESIGN OVERVIEW

2.1 System Overview

The system environment in which LoggerNetSource will be operating will be a version of Microsoft Windows on which the Campbell LoggerNet program is actively acquiring data and writing the results as files. LoggerNet will be configured to invoke LoggerNetSource as a task after each successful call to LoggerNet to process the resultant data file. This design decision was necessary because Campbell uses a closed proprietary network communications protocol in the LoggerNet implementation, and the fie output is the earliest point at which the data stream can be acquired openly. The system must have the Java Runtime Environment of at least version 1.5 installed and available. The target DataTurbine server must be reachable over the network or the Internet.

2.2 Problem Statement

The fundamental problem of accurately reproducing LoggerNet real-time data streams in DataTurbine in an automated manner will be accomplished through the deployment of LoggerNetSource and by using the features of LoggerNet itself and the host filesystem.

3. FUNCTIONAL DESIGN

LoggerNetSource is a command-line DataTurbine Source program that will read a data file that has been generated by LoggerNet. It will then parse metadata, such as channel names and physical units, as presented by LoggerNet as a file header, and translate this information to the DataTurbine paradigm. Lastly, LoggerNetSource will read the stream of data points and put them onto the DataTurbine ring buffer. Here is a list of design points that will be necessary to accomplish this:

- LoggerNetSource must be able to be directed to an arbitrary data source, a file in this case.

- LoggernetSource must accurately submit both the data values and the timestamps for these data to DataTurbine.

- LoggerNetSource must be able to be directed to an arbitrary DataTurbine server host.

- LoggerNetSource must be able to interpret and operate with an arbitrary configuration of Loggernet channels that are provided by LoggerNet.

- LoggerNetSource must be able to invoked on an event-driven basis by LoggerNet.

- LoggerNetSource must not adversely effect the hoss system or any other software applications that are running on it.

4. TECHNICAL DESIGN

The components of LoggerNetSource will be a main driver class and a class to manage the information parsing. This program will pilot the newly designed OSDT object oriented class hierarchy for DataTurbine utility applications, for which a UML diagram is represented in Figure 1. This class hierarchy serves primarily to abstract and effectively utilize the DataTurbine API, com.rbnb.sapi in order to enable application development for the OSDT community.

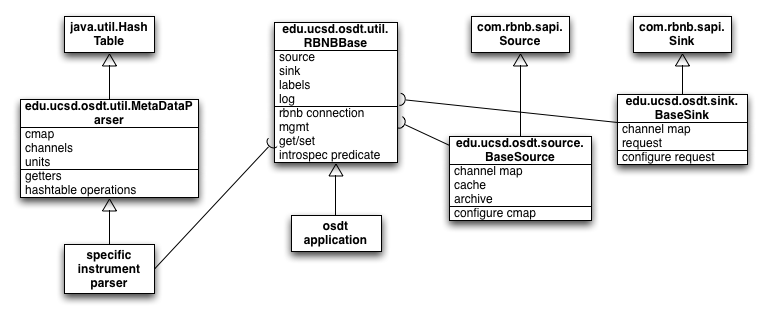


Figure UML diagram of the OSDT class hierarchy

Figure 2 depicts a diagram of the LoggerNetSources activity during a duty cycle. The actions are serial:

1. open the specified file
2. open a TCP connection to the specified DataTurbine server
3. interpret the file header and process it as metadata by generating an instance of com.rbnb.sapi.ChannelMap that contains the specified channels, each of which has the specified physical units as metadata
4. enter a loop to process each line of data in the file by:
   1. tokenizing the comma-delimited list of numbers
   2. translating the timestamp into seconds since January 1, 1970, which is the time convention used by DataTurbine

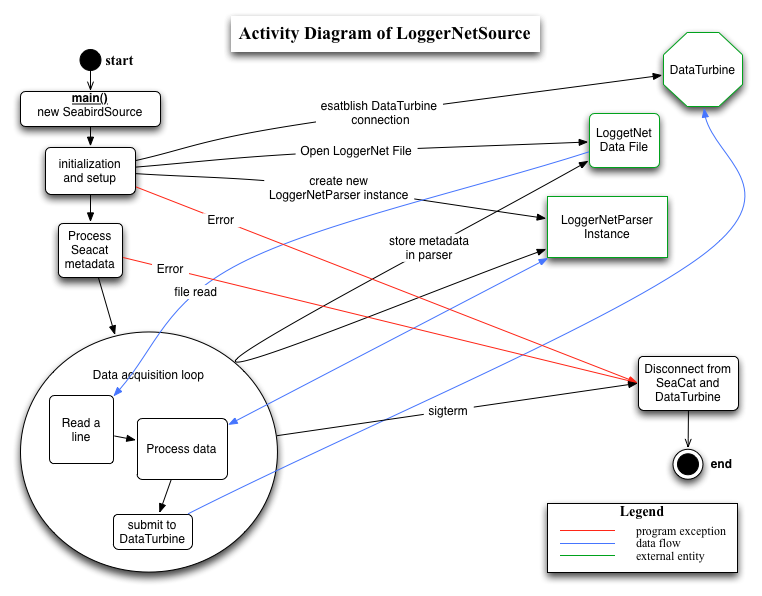


Figure 2 - Activity diagram of LoggerNetSource

Following is an example of a LoggerNet data file, a description of each, and detail of how each is processed.

LINE 1 – device information – currently ignored by LogerNetSource  
"TOA5","CR3000","CR3000","2214","CR3000.Std.07","CPU:OET\_9-nov-07.CR3","47939","LS\_30Min"

LINE 2 – channel labels – used to create the DataTurbine channel map – the TIMESTAMP channel is discarded because timestamp information is committed to DataTurbine with every data point, and it is handled specially by DataTurbine internally.  
"TIMESTAMP","RECORD","WindSp\_ms\_S\_WVT","WindSp\_ms\_U\_WVT","WindDir\_DU\_WVT","WindDir\_SDU\_WVT","Rain\_mm\_Tot","AirTemp\_C\_Avg","RH","SR\_Tot\_Avg","SR\_Dif\_Avg","Pyr\_kW\_Avg","PAR\_Den\_Avg","Pyr\_kW\_Max","PAR\_Den\_Max","Pyr\_kW\_Min","PAR\_Den\_Min","Pyr\_kW\_Std","PAR\_Den\_Std","BP\_mbar","SoilTemp\_C\_Avg","SoilWVC\_Avg","Encl\_RH\_Max","PTemp\_C\_Max","Batt\_Volt\_Min","Rad\_Lico2\_Avg","Rad\_Lico2\_Max","Rad\_Lico2\_Min"

LINE3 – physical units – presented in the same order as are channel names to which they directly map  
"TS","RN","meters/second","meters/second","Deg","Deg","mm","Deg C","%","umol m2 s1","umol m2 s1","kW/m²","umol m2 s1","kW/m²","umol m2 s1","kW/m²","umol m2 s1","kW/m²","umol m2 s1","mbar","Deg C","%","%","","Volts","kW/m²","kW/m²","kW/m²"

LINE 4 – sampling method – describes how raw sample reads are processed by the Campbell datalogger to produce a calibrated data value – currently ignored by LoggerNetSource  
"","","WVc","WVc","WVc","WVc","Tot","Avg","Smp","Avg","Avg","Avg","Avg","Max","Max","Min","Min","Std","Std","Smp","Avg","Avg","Max","Max","Min","Avg","Max","Min"

LINE 5+ - data point – a comma-separated list of doubles that correspond to the channel listing in the header. The first value is a timstamp that is translated to seconds since the epoch for use as a DataTurbine timestamp. This is done by formatting the timestamp as an ISO 8601 string, after which it is processed by the edu.ucsd.util.ISOtoRbnbTime class to produce a DataTurbine timestamp. The remaining values are submitted to DataTurbine as doubles.

“2007-11-12 07:30:00",0,1.994,1.885,253.6,18.72,0,24.27,84.5,542.1,381.2,0.19,533.3,0.272,739.2,0.134,402.2,0.037,91.5,0,26.02,0.011,12.23,22.56,13.08,0.209,0.302,0.148

5. GENERAL ISSUES

LoggerNetSource depends on Loggernet itself to invoke it upon designated LoggerNet events, and the Loggernet output as a file on the local filesystem. Issues with the LoggerNet task management feature, LoggerNet’s configuration to write output files, and the integrity of the host computer system’s filesystem will all directly effect LoggerNetSource.

6. SOFTWARE ENGINEERING INFRASTRUCTURE

LoggerNetSource is maintained in the OSDT Google code repository:

<http://code.google.com/p/oss-dataturbine/source/browse/trunk/apps/oss-apps/src/edu/ucsd/osdt/source/numeric/LoggerNetSource.java>

<http://code.google.com/p/oss-dataturbine/source/browse/trunk/apps/oss-apps/src/edu/ucsd/osdt/source/numeric/LoggerNetParser.java>