LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note	LIGO-T010062-00 - D	5/25/01

Data Access Tools

Daniel Sigg

Distribution of this draft:

all

This is an internal working note
of the LIGO Project.

LIGO Hanford Observatory
P.O. Box 1970 S9-02
Richland, WA 99352
Phone (509) 372-8106
FAX (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (504) 686-3100
FAX (504) 686-7189
E-mail: info@ligo.caltech.edu

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS NW17-161
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

www: http://www.ligo.caltech.edu/

1 Introduction

The purpose of this document is to enable the data flow from the LDAS archive(s) to the already existing diagnostics tools in a format they can understand and with an interface which is intuitive for users. It contains a proposal how this data flow is managed and how the interface is organized.

1.1 Summary

From a user (physicist) point of view, the most important features are:

- It should be possible to access data in a consistent manner, regardless of the medium on which it is stored, its location, host operating system, etc.
- All data should be accessed via a meaningful name.

The proposed solution includes a data flow manager (DFM) which servers as the intermediary between client applications requesting data and data archives. The proposed data flow manager has the following important features:

- It can directly interfaces the LDAS archive (getFrameData).
- It can directly interface a local file system which contains frame data.
- It can merge data streams (frame files) originating from different locations into a single data stream (frame file).
- It can stage data asynchronously to avoid repeated latencies associated with retrieving files from tapes individually. In effect, the user can say at the outset “here is a preview of what I am going to ask for” and the DFM attempts to stay ahead of the actual data processing.
- It is able to provide information about available data sets and their content.

For someone using the data viewer, the diagnostics test tool or the data monitoring tool the data flow and its management should be transparent. Typically, during program start-up the user will be presented with a list of available data sets, their content and their start and stop time. After selecting a data set these programs should look and feel identical to the current experience, i.e., the user can select channels from a list, choose a measurement time and starts the analysis or visualization process.

1.2 Universal Data Set Names (UDN)

Data sets are organized into logical sets. Their name resembles a directory name on a UNIX system, but there has to be no direct connection between a UDN and a file location. In fact, a UDN generally doesn’t describe an individual file but rather a set of files which belong together. Examples:

```
//ligo/raw/lho/e1 : data from the first engineering run
//ligo/raw/lho/s0003 : data form the third science run
//ligo/raw/lho : all full frame data from LHO
//ligo/raw/llo : same for LLO
//ligo/raw : all raw data from both observatories
//ligo/raw/2001 : all LIGO raw data from 2001
//ligo/trend/min/llo : minute trend from Livingston
gold.ligo-wa.caltech.edu://ligo/trend/min/lho : server name is explicit
//ligo/trend/sec : LIGO second trend
```

```
//ligo/level2 : LIGO level 2 data set
capella.ligo.caltech.edu:~pshawhan/myfiles : files on a remote machine
/export/raid1/copter/00-06-17_16:32:28 : files on the local machine
```

Physically, data sets can reside at different locations. The data flow manager brings them together on the local machine and make them accessible to the client application. Data sets can also be stored in multiple archives—for example, minute trend data will be kept at the observatories as well as in the main archive. If multiple archives exist and a server is not explicitly specified by the user, the data flow manager will have to choose which location is the most convenient. It is not envisioned that a UDN can be omitted from a request. A feature which would automatically figure out which data set could be used to fulfill a request is probably too unpredictable and a wrong choice could easily be made without the user immediately noticing.

A data sets has an associated list of available objects (e.g., channel names) and a time stamp indicating start and stop time. This information is made available to the user through the data flow manager, in order to assist the selection of data and to enable graphical user interfaces to compile channel selection lists. A data set must also have a list of files and storage locations associated with it; typically, this information is hidden and only used by the server for data retrieval.

1.3 Canonical Frame Format (CFF)

Data can be delivered through a network connection or a local file system. The standard data format is frames (CDS NDS format is also supported for backward compatibility). Even so the frame format is standardized most client programs make implicit assumptions about the size and structure of the received data blocks and how these data blocks are represented within the frame file(s). To avoid compatibility problems and to enable the data flow manager to combine data streams efficiently, frame data is delivered in a canonical format. The canonical frame format is essentially the format in which it is written at the observatories. Four types of frames are recognized: full frames of one second length (FF1), full frames of 32 seconds length (FF32), second-trend data frames (STF) and minute-trend data frames (MTF). The conventions are as follows:

- FF1 (full data frames): each frame covers 1 second of data, the data is aligned with the GPS 1 second clock, multiple frames can be concatenated into a single file, missing or invalid channel data are marked bad or are omitted on a channel-by-channel and a second-by-second basis, the raw/adc data structure is used within the frame file, data vectors can be compressed, frames contain a valid table of content.
- FF32 (full data frames): each frame covers 32 seconds of data, the data is aligned to a multiple of 32 of the GPS clock, multiple frames can be concatenated into a single file, missing or invalid channel data are marked bad or are omitted on a channel-by-channel and a frame-by-frame basis, the raw/adc data structure is used within the frame file, data vectors can be compressed, frames contain a valid table of content.
- STF (second-trend data frames): each frame covers 60 second, the data is aligned to a multiple of 60 of the GPS clock, multiple frames can be concatenated into a single file, missing or invalid data points are marked by the floating point representation of a NaN (not a number) or are omitted if the whole 60 second is bad, the raw/adc data structure is used within the frame file, data vectors can be compressed, frames write a contain table of content.

- MTF (minute-trend data frames): each frame covers 60 minutes, the data is aligned to a multiple of 3600 of the GPS clock, multiple frames can be concatenated into a single file, missing or invalid data points are marked by the floating point representation of a NaN (not a number) or are omitted if the whole 60 second is bad, the raw/adc data structure is used within the frame file, data vectors can be compressed, frames contain a valid table of content.

1.4 Tools

1.4.1 Data Monitoring Tool (DMT)

A C++ environment to work through a stream of frame files supporting both interactive and batch mode. Its primary mission is to monitor on-line data, but it can also be run with off-line data. Making it available for off-line use will allow monitor and trigger programs developed for machine studies to run unchanged on old data. It is envisioned that an additional DMT environment will be installed near the main archive to allow fast data throughput of recorded data.

1.4.2 Diagnostics Test Tool (DTT)

Its main focus is stimulus-response tests with the main detector, but is can also be useful for viewing and performing Fourier analysis (power spectra estimate, coherence, cross-power spectra and transfer functions) of recorded data. It allows to inspect time series data, and to store or read data snippets in XML, ASCII or binary format.

1.4.3 Data Viewer (DV)

Data viewer is the replacement of a scope in the control room. But, it is also useful to study long-term trends using the data stored in the second and minute trend archives.

1.4.4 Time-Frequency Analyzer

A package to calculate and display time-frequency plots (TBD).

2 Data Flow Manager

3 Fantom

The frame and NDS translation module (Fantom) is a stand-alone application which can merge frame streams, or split them up into separate streams. It uses either sockets or files as input devices and it implements a smart input feature which allows it to recognize multiple delivery formats automatically and which avoids unnecessary copy operations. The output of Fantom is either through sockets or through files; when using sockets it can also support the NDS delivery protocol. Fantom can be used to read FF1 and FF32 frames and generate trend frames on-the-fly. If requested, it can also transform between FF1 and FF32. Since it also supports trend frames which are shorter than their canonical representation, it can be used to transform trend streams generated by the frame builder into STF and MTF format.

3.1 Command Overview

Fantom has its own command line interface which can be invoked by starting the program without any arguments. It can also be run in batch mode, either through a configuration script or by specifying the desired parameters through command line arguments.

```
Usage: fantom : start interactive mode
       fantom -c 'file' : start batch mode reading commands from 'file'
       fantom -i -c 'file' : start interactive mode initializing from 'file'
       fantom -e [commands] : start batch mode reading the argument list
       fantom -i -e [commands] : start interactive mode with argument list
       fantom -h : this help
The command list has to be separated by semi-colon; typically surrounded by
quotes to prevent the shell from interpreting them.
In batch mode a go/quit command is automatically executed last.
```

An overview of the fantom commands is listed below:

Parameter	Configuration file / standard input
input/output configuration:	
open input (numbered)	in 0 open in 1 open file:///home/sigg/input_chn_1.txt in 2 open dir:///home/sigg/frames in 3 open tape:///dev/rmt/0n –f *.F in 4 open port://8090 in 5 open net://red.ligo-wa.caltech.edu:8092
open output (numbered)	out 0 open out 1 open dir:///home/sigg/result#60 out 2 open port://8091 out 3 open net://red.ligo-wa.caltech.edu:8093 out 4 open dmt:///offline/sigg –l 3000000
add input device/name	in 1 add file:///home/sigg/H-658085674.F in 2 add tape:///home/sigg/test/H-658085600.n100.tar
add output device/name	out 1 add file:///home/sigg/frames/H-658085674.F out 2 add dir:///home/sigg/test.#3600
flush ouput	out 1 flush

Parameter	Configuration file / standard input
close input/output	in 1 close out 2 close
output format:	
frame type	out 1 type FF32 out 2 type STFC0 out 3 type NDS out 4 type FF1N32C2
output channels	out 1 channels {“name1” [rate] “name2”...}
Conversion commands:	
convert	go go 32 auto / stop / wait
parameters:	
set/get	set/get clock set/get start ‘time’ set/get duration ‘interval’
others:	
read config. file	read filename
log file	log logfile
summary web page	web htmlfile
status/cmd. port	port number
quit after n frames	quit frame 10
quit after n sec w/o input	quit timeout 120

3.2 Input and Output Specifications

An input or output specification has the following format:

device [names] [options]

- [options]: option string of the form –’letter’ ‘value’
- [names]: names specifying files, directories, or addresses, etc.
- device: a device name such as file:// or tape://

Device names are terminated by a colon and double slashes, i.e., “://”. The following devices are supported:

Device	Description	Supported names and options
file://	file names and wildcards	Files can be specified by any valid UNIX format including wildcards. Example 1: “file:///home/sigg/mydata/H-658085674.F” represents a single file. Example 2: “file:///export/raid2/E2/00-11-12_17:10:19.0/*.F represents all frame files in the specified directory.
dir://	files located in a directory list	A directory name can specify a numbering scheme; the format is: 'dirname'[@startdir[.startfile]:stop[.stopfile]][#filenum] If any of the options are appended to the directory, auto-increment support is enabled. The start argument describes the first directory number and optionally the first file number. Similarly, the stop argument describes the last directory number and optionally the last file number. The file number argument is used to determine how many files should be created per directory. Example 1: "dir://test.@3:5" represents the files "test.3/*", "test.4/*" and "test.5/*". Example 2: "dir://test.@4#3600" represents the directories "test.4/", "test.5/", etc., assuming no more than 3600 files per directory.
tape://	tar archives on file or magnetic tape	If the device is a magnetic tape it should be of the format “/dev/rmt/0n”. If the device is a tar archive on disk, the name represents a file name. The following options are supported: –p ‘filepos’: file position to start (read only) –a ‘archnum’: number of archives per tape (write only) –n ‘filenum’: number of files to read (read), or number of files per archive (write) –f ‘files’: file name or wildcard (read only). –d ‘directory’: directory name to use; can contain [#filenum] (write only). –r ‘robot’: tape robot specification. The tape robot specification is of the form: ‘name[@startslot:stopslot[:first]][#tapes]’ where ‘name’ represents the tape robot—currently supported are CY0 (Cybernetics TL-8) and manual (stand-alone drive), ‘startslot’ and ‘stopslot’ denote the first and last slot to be used, ‘first’ is the number of the first tape to be used, and ‘tapes’ represents the number of tapes to be read.
ftp://	FTP location	The name must contain a valid ftp server followed by a slash and a valid file name. The format is: “host[:port]/file” Example: “ftp://ldas.caltech.edu/lho/H-658085674.F” represents a single file in the directory lho.
http://	web location	Same as FTP but must point to a file on the web.
lars://	archive access through dfm/LARS	

Device	Description	Supported names and options
nds://	NDS server	Similar to the net device, but must specify an address of an NDS server; the default port number is 8088. Three UDNs are supported: /frames for full frames, /trend for second trend data, and /minute-trend for minute trend data. If the UDN is omitted, /frames is assumed. Example: “nds://red.ligo-wa.caltech.edu/trend” asks for second trend data from the NDS server on red at port 8088.
dmt://	shared memory partition for the data monitoring tool	The name describes a shared memory partition which can be read by DMT monitor processes. The following options are supported: –l ‘length’: length of each memory buffer (in bytes); default 1000000. –n ‘num’: number of memory buffers; default 2. –o: off-line support; all clients must have read the buffer before it is discarded Example: “dmt://sigg_1 –l 3000000 –n 3” opens a shared memory partition under the name sigg_1 with three buffers 3 million bytes long each.
eof://	end of file marker	No need for a name.

3.3 Frame Type Specification

The frame format is specified by using a string of the following format:

XX ## N ## C #

- C # — compression type: 0 – none, 1 – default
- C — compression identifier
- N ## — number of frames per file: default 1
- N — frame number identifier
- ## — frame length in seconds (only for full frames)
- XX — format: FF – full frames, STF – second trend frames, MTF – minute trend frames, TXT – text file, BIN – binary file

Examples:

- FF32C0 – full frames, 32 second long, 1 frame per file, no compression,
- STF – second trend frames, compressed,
- FF1N128C2 – full frames, 1 second long, 128 frames per file, compression level 2.

4 Protocols

4.1 Data Request Protocol

4.2 Data Delivery Mechanisms

Appendix A XML Configuration

A lidax or fantom configuration can be saved to and retrieved from a LIGO lightweight format object. An example which reads data from three sources (/raw/lho/E3 at the archive, /raw/llo/E3 at the archive and /export/raid2/E3/pre.# in the local disk system) and sends it to a shared memory partition:

```
<LIGO_LW Name="Lidax">
   <Time  Name="Start" Type="GPS">615445949</Time>
   <Param Name="Duration" Type="double">60</Param>
   <Param Name="Server[0]" Type="string">LARS</Param>
   <Param Name="UDN[0]" Type="string">/raw/lho/E3</Param>
   <Param Name="Channel[0][0]" Type="string" Unit="channel">H0:PEM-*</Param>
   <Param Name="Rate[0][0]" Type="double">256</Param>
   <Param Name="Channel[0][1]" Type="string" Unit="channel">H0:LSC-*</Param>
   <Param Name="Server[1]" Type="string">LARS</Param>
   <Param Name="UDN[1]" Type="string">/raw/llo/E3</Param>
   <Param Name="Channel[1][0]" Type="string" Unit="channel">L0:PEM-*</Param>
   <Param Name="Rate[1][0]" Type="double">256</Param>
   <Param Name="Server[2]" Type="string">Local file system</Param>
   <Param Name="UDN[2]" Type="string">dir:///export/raid2/E3/pre.#</Param>
   <Param Name="Client[3]" Type="string">Shared memory partition</Param>
   <Param Name="UDN[3]" Type="string">/LHO_offline -o -l 300000 -n 4</Param>
   <Param Name="Format[3]" Type="string">FF1N1C0</Param>
   <Param Name="Log" Type="boolean">1</Param>
   <Param Name="Logfile" Type="string">lidax.log</Param>
   <Param Name="Web" Type="boolean">1</Param>
   <Param Name="Webfile" Type="string">lidax.html</Param>
   <Param Name="Email" Type="boolean">1</Param>
   <Param Name="Email" Type="string">sigg_d@ligo.caltech.edu</Param>
   <Param Name="Progress" Type="boolean">1</Param>
</LIGO_LW>
```

When storing a set of configuration record in a file or when sending it over the network, a well-formed XML doument has to be built. It follows the LIGO-LW defintion and may look like:

```
<?xml version="1.0"?>
<!DOCTYPE LIGO_LW SYSTEM "http://www.cacr.caltech.edu/projects/ligo_lw.dtd">
<LIGO_LW>
   <LIGO_LW Name="Lidax">
      ...
   </LIGO_LW>
</LIGO_LW>
```

The definition for the parameters is as follows:

Name	Type	Dim	Description
Server	st	N	Name of data source: – LARS: LIGO archive server – NDS <address>: Network data server at <address> – Local file system – Shared memory partition – Local tape drive/robot – Realtime: if supported
Client	st	N	Name of data sink: – Local file system – Shared memory partition – Local tape drive/robot
Start	time	1	Start time in GPS sec.
Duration	d	1	Duration in sec.
UDN	st	N	Universal data set name
Channel	ch	N×M	Selected channel name
Rate	d	N×M	Selected data rate
Format	st	N	Frame format
MonitorName	st	O	Name of DMT monitor program
MonitorArg	st	O	Name of DMT monitor command line arguments
MonitorData	st	O	Descibes the UDN of the monitor data source
MonitorKill	b	1	Terminate monitors when done?
Log	b	1	Write a log
Logfile	st	1	Name of log file
Web	b	1	Write a html summary
Webfile	st	1	Name of html file
Progress	b	1	Show progress bar
Email	st	1	e-mail address for “done” message

The index N is used to indicate the server or client identification number. The same identification number must be used when specifying UDN, channel, rate and format. If no channel information is present, the default is all channels. If a channel information doesn’t have a rate specification, the default is full rate. More than one channel and rate selection can be specified for each server/UDN or client/UDN pair.