

# Data Monitoring Tool Status

John G. Zweizig  
*LIGO, Caltech*

LSC collaboration meeting  
Livingston, March 16-18, 2000

---

## Progress since July LSC meeting

- Installation
  - LHO Status
  - LLO Progress, status
- Changes to existing functionality
  - Improved distribution and online documentation.
  - New Functionality in container classes.
- New Functionality
  - Background Data Analysis environment
  - Writing frame files - Designer data sets.
- Future plans
  - Data API (LDAS) code development.
  - Trigger generation.
  - Communication with background processes
  - Miscellaneous improvements.

---

## Installation at LHO

As of last LSC meeting:

- One Sun E-450 (*sand.ligo-wa.caltech.edu*)
- 4 × 400MHz CPUs, 1 GBy memory
- ATM (OC-3) to CDS, GC and LDAS networks

Recent improvements

- Second (identical) E-450 (*stone.ligo-wa.caltech.edu*)
- Dedicated Frame builder (via Gbit Ethernet).
- Designer dataset collection on GC node (fortress).
- Process management function.

Usage

- On average 4 monitors, 2-3 users (< 10% total cpu power).
- No problems so far with “Honor system” resource management.
- Access granted on individual basis (contact J. Zweizig or D. Sigg to request an account).

---

## **Installation at LLO**

Sun E-450 *delaronde.ligo-la.caltech.edu* newly installed

- Identical configuration to sand & stone.
- No connection yet to CDS or LDAS networks (no ATM adapter).
- Online environment mirrored from LHO
  - System applications (/usr/local) directories.
  - DMT environment (/export/home/dmt) directories.
- Dedicated Frame Builder broadcasts data over Gbit Ethernet.
- In use since 3/11/2000.

---

# Distribution and Documentation

## Recent releases

- Version 1.1 (January 4, 2000)
  - Data Environment
  - Double-sided FSeries
  - FSpectrum
- Version 1.1a (January 22, 2000)
  - Bug fixes in FSeries, display macros.
  - Linux compatibility
- Version 1.1b (February 22, 2000)
  - Incorporate Idas-0.0.9 FrameCPP & build.
  - Minor compatibility improvements.

*<http://www.ligo.caltech.edu/~jzweizig/DMT-Download.html>*

- List of s/w needed for installation.
- Solaris installation notes (thanks to Ed Daw).
- Pointers to distribution kits.

---

## **Data Analysis Environment: DatEnv**

---

### Goals:

- Provide structure for monitor development
- Prevent redundant development of code.
- Insulate users from eccentricities of FrameCPP
- Supply requested data in an easily used form
- Run in background or under root.

### Implementation:

- Monitor environment base class in C++ .
- Environment provides setup/setdown, Data I/O, interrupt handling.
- Requested data stored automatically in TSeries.
- Monitor functionality provided in ProcessData().

---

## Data Analysis Environment (cont'd)

---

Monitor Recipe:

- Start with templates `DatTemplate.hh`, `DatTemplate.cc`
- Define global data in private section of your monitor class.
- Fill in methods in `DatTemplate.cc`
  - **Monitor(int, char\*\*)**: Monitor initialization, *e.g.* configure the monitor, specify time stride, request needed channels.
  - **~Monitor(void)**: Release allocated storage, print results.
  - **ProcessData(void)**: Perform all monitor specific analysis on one time segment.
  - **Attention(void)**: User defined SIGUSR1 signal handling, *e.g.* print results, reconfigure, re-set calculations.

---

## Frame Writing

FrWriter API:

- Writes designer frames from TSeries data.

FrWrite Function:

- Copies a specified subset of channels to output frames.
- User selectable number of frames per file, number of files.
- Script interface for data writing to
- In use at LHO - Only current data storage.
- data servers by mini-NDS.

Frame Directory Utilities:

- FrameDir class tabulates frames specified directory(s).
- Uses quick scan to find number, length of frames in file.
- Will write frame directory utility functions based on FrameDir.



---

## Changes to Container Classes

### Data Vector:

- Now based on Copy-on-Write vector.
- Many copies are transparent (*e.g.* when series returned from a function).
- Min, Max, NGreater, NLess (version 1.2).

### Time Series:

- Uses copy-on-write Vector (see above).
- Min, Max, NGreater, NLess (version 1.2).
- Inverse FFT constructor (version 1.2).

### Frequency Series:

- Uses copy-on-write Vector (see above).
- Contains double-sided (and single-sided) series.
- Correct normalization of spectral density.

---

# **Communication with Background**

---

## Goals:

- Status interrogation, display facility.
- Displays generated only upon request.
- Simple interface for monitor developers.
- Low overhead, Flexibility.
- Possible extension to command interface.

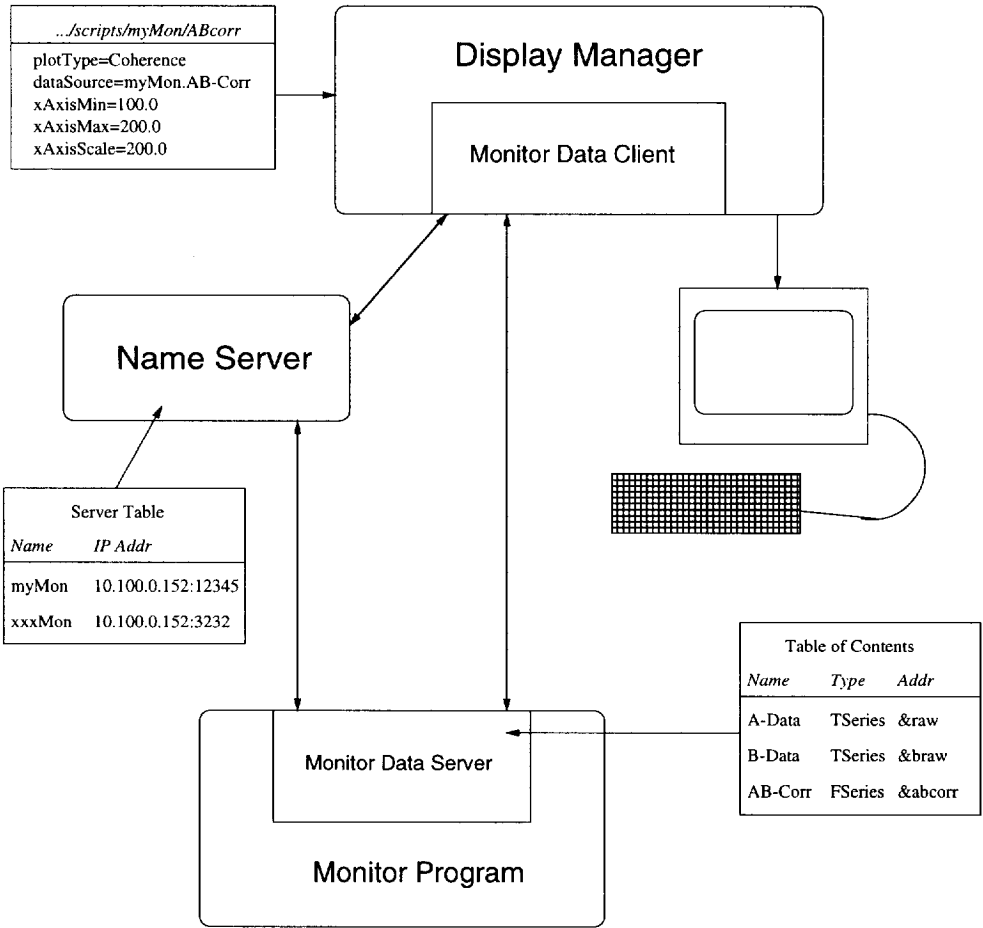
## Data Server:

- Small data server incorporated automatically into monitors.
- User describes data objects to served to external clients.
- Light-weight UDP-IP protocol used to send/receive datagrams.

## Display Manager:

- User has choice of predefined displays.
- Script tells display manager where to get data for display, how to present it.
- Uses GDS display facilities to present data.

# Background Communication Scheme



---

## Miscellaneous Plans (Future)

---

### Audio:

- Audio API has been written (Solaris Only)
- Listen, DAQListen monitors and GUI.
- Possible future expansion using OSS.

### Trend Frames:

- API for writing standard trend frames.
- For use as monitor summaries.
- Contain  $N_{SAMPLE}$ ,  $\langle x \rangle$ ,  $\langle x^2 \rangle$ ,  $min$ ,  $max$ .

### Efficiency Improvement - DaqReader:

- Address greatest concern: monitors read full frame.
- Unpacks only requested data.
- Knows where to look in frame file.

### More/Better/Standard Functionality:

- Resampling, deskewing of time series.
- Interface dataAPI code: construct TSeries(valarray).
- Move to LIGO/LDAS standards (FFTW, dataAPI, LAPACK, ...)

---

## **Installation procedure**

1. Add monitor sources & documentation to cvs
  2. Link to documentation from GDS/DMT web page
  3. Compile from CVS into production directory
  4. Add monitor to process manager table
  5. Enable trigger logging
-

---

## **Delivery requirements**

1. Full (html?) documentation
  - Purpose of monitor
  - Algorithm
  - Description of all inputs
  - Description of all outputs (triggers, reports, trend data, served data)
  - List of required packages
2. Full source(s)
3. Sample make-file

*Note 1, Linda Turner, 05/09/00 09:42:48 AM*  
LIGO-G000080-00-D