

---

# Implementation of LIGO/ VIRGO Frame Format

## Status

A. Lazzarini

GWIC/PAC Meeting  
LIGO Livingston Observatory (LLO)  
22 April 199



CALIFORNIA INSTITUTE OF TECHNOLOGY  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

1

LIGO-G980061-00-E

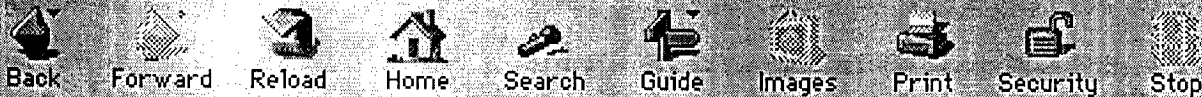
/home/lazz/Presentations/PAC/PAC\_9804\_GWIC\_Frames.fm5

# Frame Format Implementation

## Status

---

- LIGO/VIRGO have been testing Frame format on 40m prototype for ~ 1 yr.
  - ›› Implemented a Frame library software archive site at VIRGO (LAPP)
  - ›› CVS for version control
  - ›› Testing has identified (and repaired or implemented with programming support from VIRGO) numerous bugs, improvements, modifications and
    - static data structures not properly written
    - timing/speedup
    - memory leaks/system crashes
    - data compression (gzip, differential)
    - fixed-length frames for data acquisition
    - etc.
  - ›› Made decision to rewrite Frame library as an API (application programming interface) in C++
    - Joint effort with VIRGO/LAPP
    - Opportunity for LIGO to develop in-house expertise in maintaining Frame code.
  - ›› APIs being developed
    - MATLAB (VIRGO)
    - TRIANA (GEO)
    - DATA ARCHIVE (LIGO)
    - ....



Location: <http://www.cacr.caltech.edu/ligo/>

# Ligo Data Catalogue

Welcome to the Prototype LIGO Data Archive Web Services

The URL for this page is <http://www.cacr.caltech.edu/ligo>

This page explains how to get customized multichannel time-series data from the LIGO data archive that is housed at the Center for Advanced Computing Research at Caltech. The archive is stored on a 30-Terabyte IBM tape robot running the High Performance Storage System (HPSS).

The data available through this temporary prototype site are limited to the datasets acquired May 1 and Sept 3/4 1997, approximately 65GB. At that time, 117 channels of data at various sampling rates were acquired. The data are uncalibrated. The interferometer was marginally operating and often was not locked. Many of the signal names have changed or are being redefined. The frame data were acquired with Frame Library Version 2.36.

- The time stamp is GPS time.
- No calibration data were written to the corresponding frame structures.
- Data are verified to be readable with Frame Version 2.37; compatibility with later Frame Versions is not guaranteed.

Please contact Serap Tilay if you have any questions about the meaning of the data. For questions and feedback on the interface, write to [data@ligo.caltech.edu](mailto:data@ligo.caltech.edu).

If you don't want to read anything more, go straight on to the data by clicking the following link

**[Begin Data Access Wizard](#)**





## Begin Data Access Wizard

The archive consists of a catalog hosted by the Access database. The Wizard below is driven by the WebBase middleware. The archive consists of a collection of *Datasets*. Each of these represents a run of an interferometer over a compact time period, with the machine parameters remaining constant. The first page of the wizard asks you select one of these from a list.

Next you are asked to select the output format for the data you are requesting. For more information, see the web page on LIGO Data Formats. Choices are **Frame**, **SDF**, **netCDF** (not yet implemented), and **audio**. For the latter you will need to set up your browser to accept audio files such as this one.

If you choose Frame format, there is no choice about which channels you want--you get them all. Otherwise, your page listing all the available channels corresponding to the Dataset that you chose. The SDF and netCDF formats contain several channels which are *all the same frequency*, and the audio output contains only one channel. Select the channel(s) that you want and click the "Next" button to continue.

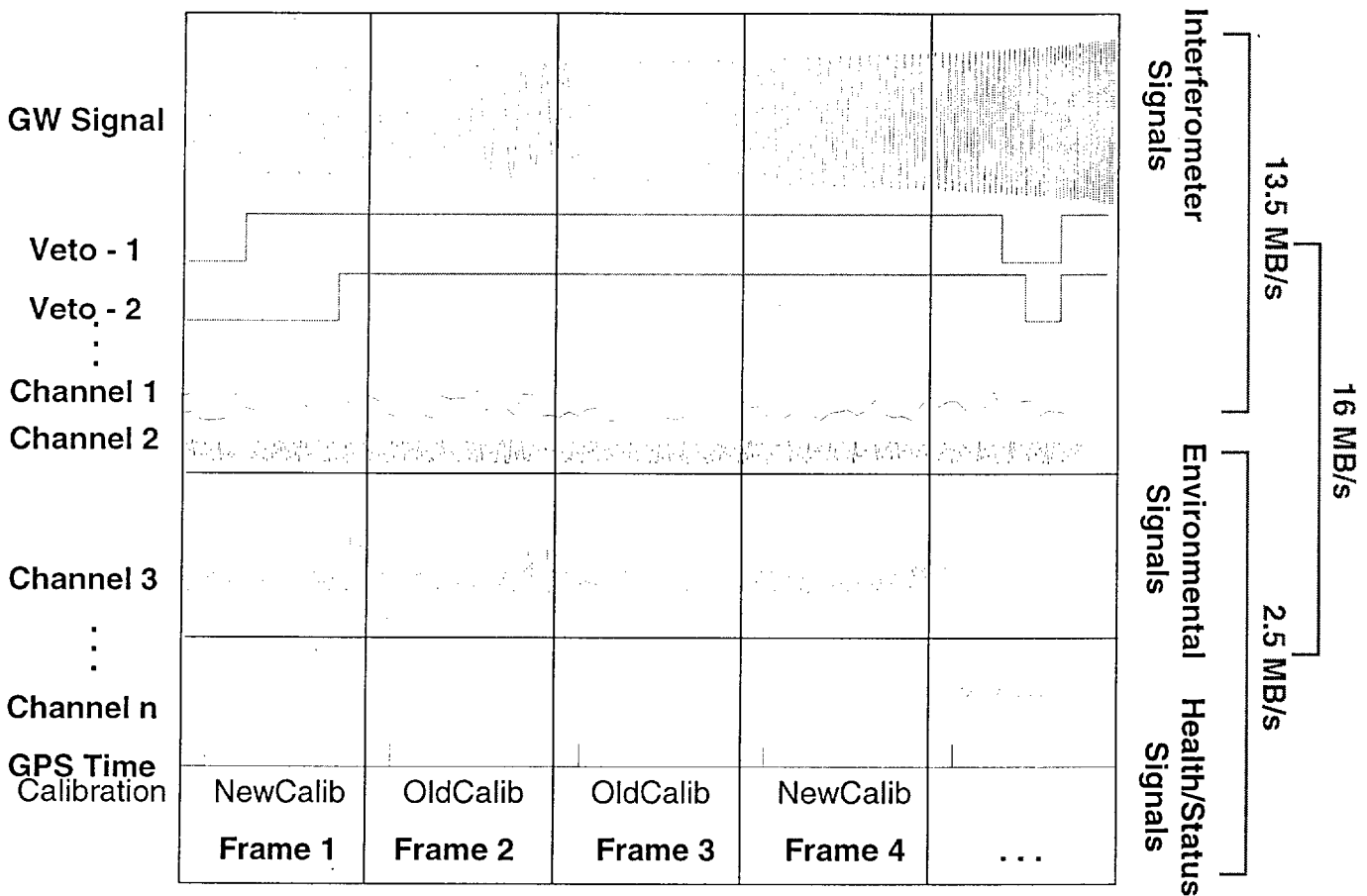
Next you get a list of data segments to choose from, each a few minutes long. Select one of these, then choose what part of that segment that you would like, by typing in the offset, in seconds, from the beginning of the segment, and number of seconds you would like sent to you. Click the "Next" button to continue.

The final page shows the structure of the query that you have generated. When you click on the "Get Data" button, server is activated and data will begin to flow. Since the data is on a tape robot, it may take a couple of minutes to start. The data will come with a MIME type that depends on the format you chose:

Frame	application/x-igwd-frame
SDF	application/x-SDF
netCDF	application/x-netcdf
audio	audio/basic

Of course, your browser may not know what to do with MIME type "application/x-SDF", and you will be asked if you want to save the data to disk. Now you can read it with your favourite software.

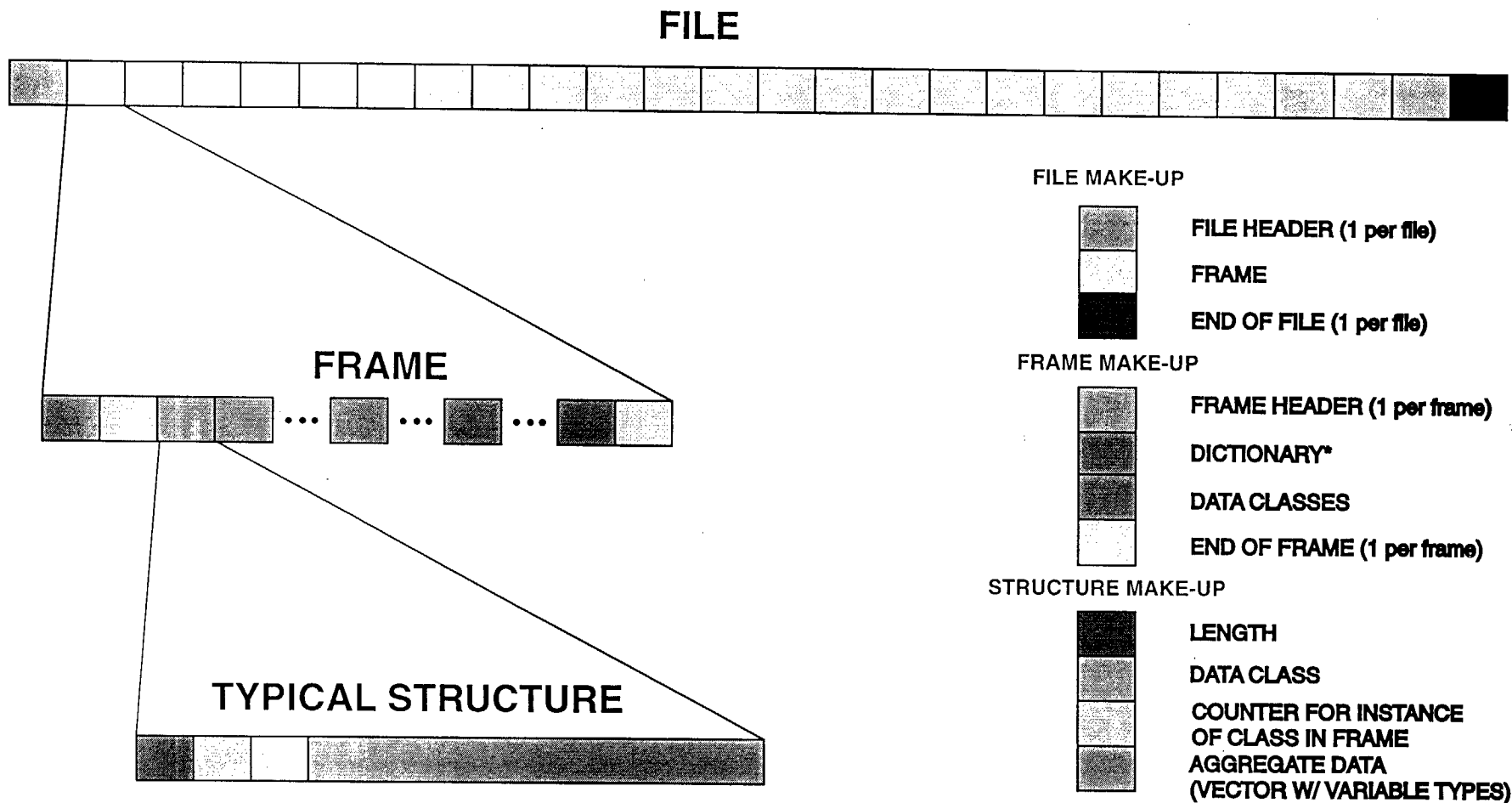
# LIGO Data Stream and Data Frame Design



- Frame is (structured) self-contained snapshot of data for a period of time
  - GW channel & ancillary IFO channels
  - Environmental monitoring (veto) channels
  - Facilities/Vacuum health & status
  - Hierarchical organization of data reflects IFO subsystems for more efficient veto utilization

# Frame Format Implementation

## Frame Composition



- \* Dictionary structure behavior is unique in that:
1. It precedes header for first frame of file;
  2. Dictionary is built up incrementally as additional structures are incorporated into frame
  3. It is valid for entire file (persistent)

# Frame Class Library and Frame API Development and Status

---

- Motivation

- ›› Develop a true C++ Frame Class Library (Fcl)

- ›› Develop a true C++ Application Programming Interface (API)

- ›› Object Oriented Technology

- Modular

- Extensible

- Portable

- Consistent API

- ›› C++ Features

- Class hierarchy

- Multiple inheritance

- Polymorphism

- Pure Virtual methods

- Function Templates

- Strong type checking

- Access Control

# Frame Class Library and Frame API Development and Status

---

- Approach - joint effort with VIRGO
  - ›› Start from the Frame Specification
  - ›› Design Classes
    - Define public methods
  - ›› Layout of class hierarchy and relationship
  - ›› Implement Design
    - Frame class library (Fcl)
  - ›› Design consistent set of Frame API
  - ›› Implement Frame API in C++



# Frame Class Library and Frame API Development and Status

---

- Status

- ›› Fcl Classes Defined and Implemented

- Constructors
    - Copy constructors
    - Destructors
    - Operators: assignment (=), concatenation (+)
    - Member access methods: Set and Get

- ›› Dictionary Classes Defined and Implemented

- ›› API Defined and Partly Implemented

- Generic methods
    - File level
    - Frame level
    - Module level

# Frame Class Library and Frame API Development and Status

---

- To-Do List

- ›› I/O Methods - VIRGO to supply
- ›› Complete API
- ›› Exception Handling
- ›› Comprehensive Debugging of Library and API
- ›› Documentation
  - Users Manual
  - Examples for every function/application

# Frame Format Implementation Status

---

- TAMA Plans for Frame Format Implementation

(information provided by Prof. Kanda)

- ›› Presently constructing final data acquisition system - will be writing Frames.
- ›› Prototyped Windows-NT data storage host for Frames.
- ›› Changed to a UNIX (Sun/Solaris 2) host
  - Testing of main program to be completed by middle of May.
- ›› TAMA is presently using Frame lib v 3.40.
  - Data from high speed ADC will be reformatted in Frame format.
- ›› Also acquiring EPICS data (slow) from environmental monitoring system
  - EPICS is also reformatted, and merged with high speed ADCs.
- ›› Final merged data will be storage on disk server and archived to magnetic tape (DLT)

# Frame Format Implementation Status

---

- GEO Activities

- ›› GEO will archive data in Frame format
- ›› Visit in February by Schutz, Robertson, Taylor
  - Reviewed ~~mutual~~ data analysis system design plans
  - GEO reviewed in depth LIGO data acquisition system design
  - Considering adopting design for GEO
- ›› Subsequent visit by Taylor to install GEO on-line analysis/visualization code (TRIANA/Java) at 40m
  - Environment presently being tested/debugged at Caltech
  - Attractive to LIGO as a (low bandwidth) on-line data visualization system
  - Developed a Java API to TRIANA to read/display Frame data

# Frame Format Implementation

## Status

---

- Issues

- ›› Concern (within LIGO) on how fast frame data access will be
- ›› Recognized need for a less structured data type for a single (or very few) channels; processed data (spectra, decimated/filtered data, etc.)
- ›› LIGO will implement a LIGHTWEIGHT DATA format
  - Useful for researchers working on small datasets
  - Diagnostics/real-time data distribution
  - Keeping data snippets associated with events
  - etc.
- ›› LIGO has identified a two-fold LW format design (existing)
  - SDF (structured data format, developed at Caltech/CACR for data storage/passing among large computers) for ASCII representation
  - notify for binary representation (commercial standard, recognized; many APIs available to visualization & data analysis environments)
- ›› No cooperation at this stage among different groups
  - Anticipate that similar realizations/needs will be identified as use matures
- ›› Need to identify metadata convention - work with VIRGO just starting
  - events
  - catalogs
  - etc.