



# Storing time-domain calibration information in frame files

Martin Hewitson, Gabriela Gonzalez,  
Mike Landry, Brian O'Reilly  
Albert Lazzarini, Xavier Siemens

# Documentation

- This presentation is based on
  - Technical note T040026-01-E
    - in writing
  - Telecons and discussions

# Current status

- Both LIGO and GEO have time-domain calibrated data
- So far both calibrated data and info are stored in a dispersed way
  - web pages, logbook pages, frame files, other...
- Both LIGO and GEO currently store double precision  $h(t)$
- We have no coherent final calibration product as yet

# Philosophy

- Capture sufficient information to reproduce the calibration at any time in the future
  - code version, raw data, filter/processing data
  - Calibration code must be under version control (some already is)
  - Have various reduced data set levels
  - Store double and single precision version of  $h(t)$
  - Keep data quality information with calibrated data
  - Have a generic storage system across detectors

# Frame file naming convention

<SITE>-<IFO>\_RDS\_C##\_L#-<GPSTimeSec>-<DeltaT-sec>.gwf

<SITE>: G, H, L

<IFO>: Interferometer ID = H1, H2, L1, G1, etc.

C##: Version of calibrated data, ## = 01-99

L#: RDS level (see later), # = 0,1,2,...

# Proposed RDS Levels

Level 0
<ul style="list-style-type: none"><li>• <b>Detector info</b></li><li>• <b>Calibration version number</b></li><li>• <b>High-level data quality</b></li></ul>
<ul style="list-style-type: none"><li>• raw data</li><li>• calibration filters</li><li>• detailed data quality info</li><li>• band-limited <math>h(t)</math><ul style="list-style-type: none"><li>• single and double precision</li></ul></li></ul>

Level 1
<ul style="list-style-type: none"><li>• <b>Detector info</b></li><li>• <b>Calibration version number</b></li><li>• <b>High-level data quality</b></li></ul>
<ul style="list-style-type: none"><li>• calibration filters</li><li>• detailed data quality info</li><li>• band-limited <math>h(t)</math><ul style="list-style-type: none"><li>• single and double precision</li></ul></li></ul>

Level 2
<ul style="list-style-type: none"><li>• <b>Detector info</b></li><li>• <b>Calibration version number</b></li><li>• <b>High-level data quality</b></li></ul>
<ul style="list-style-type: none"><li>• calibration filters</li><li>• detailed data quality info</li><li>• single precision <math>h(t)</math></li></ul>

# Frame header (`FrameH`)

- Standard use of most fields
- **dataQuality** field to store high-level data quality info

# ADC Structures (`FrADCData`)

- Store raw (input) data
  - `AS_Q`, `DARM_CTRL` channel for LIGO
  - `LSC_MID_EP-P_HP`, `LSC_MID_EP-Q_HP`, `LSC_MID_CAL` for GEO

# Detector structure (`FrDetector`)

- Detector specific details
  - site, orientation, etc.
- Table of (detailed) data quality flags

# History information (`FrHistory`)

- Store information about this version of the calibrated data
  - Production site, code version, production date, production machine(s), etc.



# Processed data (`FrProcData`)

- Storage of intermediate data products
  - Estimated detector parameters
    - $\alpha(t), \beta(t)$  for LIGO
    - $P0-n(t)$  for GEO
- Storage of  $h(t)$  time-series
  - vector of data (single and double precision)
  - valid frequency range

# Static data structures (FstatData)

- To store all filters and quasi-static information used in calibration process (IIR, FIR)
  - coefficients, sample rate, etc.
- Filter data to be stored in each frame file
- New (internal) structures need defined for storing this data

# Time-scale

- First version of documentation to be in place in the next month or so
- S3 data in reproduced by mid august
- Fully tested implementation in place for S4 (October)

# Open questions

- What size of frame files do we want?
  - examples
    - 16x1 second frame per file (~4 Mb per file)
    - 60x1 second frames per file (~15 Mb per file)
    - 3600x1 second frames per file (~900 Mb per file)
    - .....
  - Static data (filters etc.) to be stored in each file
- Is single precision needed in the long term?

# Summary

- Aim to have a fully self-consistent format for storing calibrated  $h(t)$  data
  - consistent across detectors
  - various RDS levels
  - full calibration version history
- *Calibrated data sets should be reproducible from information in the frames*