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

Technical Note **LIGO-T000067-00 - D** 6/19/00

Guidelines for representing series data in LIGO lightweight format

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 goal of this document is to standardize the LIGO lightweight representation of the most common complex data types encountered in LIGO. The LIGO lightweight data format standard is described in http://www.cacr.caltech.edu/projects/xsil/xsil.pdf and http://www.cacr.caltech.edu/projects/xsil/xsil_spec.pdf (T990023-01). This document takes this basic elements provided by the standard and builds up representations of time series, Fourier series, coherence spectra, cross-power spectra, transfer functions, coherence functions, and transfer, coherence, harmonic and intermodulation coefficients.

This document does not intend to develop a representation for every possible data object, but rather to provide a standard representation for some of the most common ones, so that they can be easily exchanged between different programs and displayed by the same viewer. To do so it develops a naming convention and eliminates as many ambiguities in format and type as possible.

2 FUNDAMENTAL DATA TYPES

The LIGO light weight data format supports both data object and parameter objects. An optional type specification can be added to the XML head tag. A list of supported type is presented below:

Name	C/C++ data type	Short	type name
8 bit integer	char	c	byte
16 bit integer	short	s	short
32 bit integer	int	i	int
64 bit integer	long long	ll	long
boolean	bool	b	boolean
single precision floating point	float	f	float
double precision floating point	double	d	double
single precision complex number	complex<float>	zf	floatComplex
double precision complex number	complex<double>	zd	doubleComplex
string[1]	char*	st/ch	string
time	long long / double	time	GPS[2]

1. To distinguish channel names from a normal string the unit of the later is set to “channel”.
2. A time parameter has a special format (see below).

3 PARAMETER OBJECTS

Parameter objects as defined by the LIGO-LW standard are represented by:

```
<Param Name=”prm_name” Type=”type_name” Unit=”unit”>value</Param>
```

Parameters have a name, type, an optional unit and a value associated with them. The type name has to be one of names listed in the table above.

A special case its the time parameter which has the following format:

```
<Time Name=“prm_name” Type=”time_type”>time_value</Time>
```

The time type used by the data objects described in this document is “GPS”, and the time value is an integer number representing GPS nano seconds.

4 COMPLEX DATA OBJECTS

A complex data object has the following format:

```
<LIGO_LW Name="object_name" Type="object_type">
  <Param Name="Subtype" Type="int">subtype_num</Param>
  ...
  <Array Type="float">
    <Dim>dim1</Dim>
    ...
    <Stream Encoding="BigEndian,base64">
R8MFzEeJz39DviwlQ5nOFkOP/01Dl9G+Q6ZTTUOYjk5DoLxnQ67sSUN+lApDYUue
Q3fH30NZqRRDb/LgQ4E+8ENvYV1DcSLsQ0Fb10MZTgVDT0qUQzQ/w0MVcvtDFnhz
QwRwSELnUF9C3c2vQtjKwELP+B9CtAiFQqQkcUKuxIZCsrA2QqFDWEKByDtCbzXH
QAUZQj/3wT5ABEVOQBuZHkAPA4NABvLRQARuj0ALsEtADGTi
    </Stream>
  </Array>
</LIGO_LW>
```

Data objects obey the following conventions:

i) The `“object name”` is mandatory but arbitrary,

ii) The `“object type”` for the data objects described in this document are:
`“TimeSeries”` for time series data,
`“Spectrum”` for Fourier series, coherence, amplitude density and, cross-power spectra,
`“TransferFunction”` for transfer functions and coherence functions, and
`“Coefficients”` for transfer, coherence, harmonic and intermodulation coefficients.

iii) A data object can specify a subtype by providing parameter with name `“Subtype”` and type `“int”`. If no subtype is specified the following subtype default values are assumed:
Time Series: real data – 0, complex data – 1,
Spectrum: real data – 1, complex data – 0,
TransferFunction: real data – 5, complex data – 3, and
Coefficients: real data – 7, complex data – 4.
Since both time and frequency series data are most often using equal spacing of x values, these data objects are preferably stored as Y-arrays only. On the other hand, transfer functions and list of coefficients will likely contain unequal x value spacing and are preferably stored as XY-arrays.

iv) An arbitrary number of parameter objects can be specified. However, the above data objects have mandatory parameters to provide the information which is required to make them interchangeable.

v) Data objects have an array tag which describes the data type, the dimensions of the data array and which contains a data stream encoding the actual data.

vi) The array type is either `“float”` or `“complexFloat”` depending on the object type and subtype.

vii) The data is always big-endian, base64 encoded. Little-endian machines have to swap the data before writing and after reading.

viii) To avoid excessive parsing overhead for retrieving the ascii encoded binary data, the ascii stream is formatted as follows:
– The first stream character starts on a new line after the stream tag,
– each line is exactly 64 characters wide with no spaces, except the last line, and
– the end of stream tag follows on the next line after the last stream character.
Following this rule can result in a significant increase in speed when reading large amount of data, since it allows the LIGO lightweight reader to bypass normal XML parsing.

ix) When a data stream consists of complex numbers and both the x and y values are included, the x values are in general real, but they are still stored as complex numbers with their imaginary part set to zero.

x) The dimensions of a data object are specified as part of the array object. Most data objects will have 2 dimensions, e.g., MxN, with N the number of data points and M the number of series or spectra data described by the object. If M=1, only one dimension may be specified. When writing data objects, two additional parameters, M and N, should be associated with the data object. They are redundant information intended for the quick browsing, but should be ignored when reading the data in favor of the dimension arguments of the array object. Also, the number M is one smaller than the one specified by the array if the data object contains x-values (time or frequency).

5 Time Series

Name	Type	Dim	Man.	Description
				TimeSeries
Subtype	i	1	x	0 – normal time series in format (Y), 1 – down-converted time series in format (Y), 2 – averaged time series in format (Y), 3 – averaged time series in format (mean, std. dev., min., max., rms), 4 – normal time series in format (t,Y), 5 – down-converted time series in format (t,Y), 6 – averaged time series in format (t,Y), 7 – averaged time series in format (t, mean, std. dev., min., max., rms),
t0	time	1	x	start time in GPS nsec.
dt	d	1	x	temporal spacing in sec.
tp	d	1		pre-cursor time in sec. The time stamp of the first data point is: t0 – tp.
f0	d	1		modulation frequency in Hz (type 2/5), trigger rate in Hz (type 1/4).
tf0	time	1		start time for modulation signal in GPS nsec.

Name	Type	Dim	Man.	Description
Decimation	i	1		overall decimation factor; default 1.
Decimation1	i	1		decimation factor before modulation (first stage); default 1.
DecimationType	i	1		decimation filter type identifier; default 1 (see Table 1).
DecimationFilter	st	1		description of decimation filter.
DecimationDelay	d	1		delay introduced by the decimation filter in sec.
DelayTaps	i	1		number of taps in the time delay filter (in number of original samples).
TimeDelay	d	1		remaining time delay of time series in sec.
AverageType	i	1		0 – fixed number, 1 – running (exponential weight).
Averages	i	1		number of averages.
Channel	ch	1		channel name
N	i	1	x	number of points.
Unit	st	1		physical unit.
	f / zf	N 2×N 5×N 6×N		time series in format (Y), time series in format (t,Y), time series in format (mean, dev., min., max., rms), time series in format (t, mean, dev., min., max., rms).

6 FFT and (Cross) Power Spectrum

A data object which describes FFT and power spectra can contain multiple spectra. By convention the first one always describes the averaged spectrum. The following table lists the parameters associated with a power spectrum:

Name	Type	Dim	Man.	Description
				Spectrum
Subtype	i	1	x	0 – FFT in format (Y), 1 – power spectral density in format (Y), 2 – cross-power spectrum in format (Y), 3 – coherence in format (Y), 4 – FFT in format (f, Y), 5 – power spectral density in format (f,Y), 6 – cross-power spectrum in format (f,Y), 7 – coherence in format (f,Y).
f0	d	1	x	start frequency in Hz.
df	d	1	x	frequency spacing in Hz.

Name	Type	Dim	Man.	Description
t0	time	1	x	start time in GPS nsec.
dt	d	1		temporal spacing in sec (only useful for averaged power spectrum).
BW	d	1	x	resolution bandwidth
Window	i	1	x	0 – uniform (no window), 1 – Hanning, 2 – Flat-top, 3 – Welch, 4 – Bartlett, 5 – BMH, 6 – Hamming, 7 – Kaiser.
AverageType	i	1		0 – fixed number, 1 – running (exponential weight).
Averages	i	1		number of averages (only useful for power spectra).
ChannelA	ch	1		channel name
ChannelB[M]	ch	1		2nd channel name(s) for cross-power spectra
N	i	1	x	number of points.
M	i	1	x	number of spectra.
Unit	st	1		physical unit.
	f zf	M×N +N M×N +N		power spectral density/coherence in format (Y), power spectral density/coherence in format (f,Y), FFT/cross spectrum in format (Y), FFT/cross spectrum in format (f,Y).

Diagnostics tests will generally use subtype 1 for a power spectrum, subtype 2 for the cross-spectra and subtype 3 for the coherence. Power spectra are typically stored individually ($M = 1$), whereas cross-spectrum and coherence are stored as a set of spectra with one A channel and multiple B channels.

7 TRANSFER FUNCTION AND COHERENCE

A transfer function object can contain multiple transfer functions of the same two measurement points. By convention the first one is the average of the following ones. The following list presents the associated parameters of a transfer function:

Name	Type	Dim	Man.	Description
				TransferFunction
Subtype	i	1	x	0 – transfer function B/A in format (Y), 1 – transfer function A in format (Y), 2 – coherence B/A in format (Y), 3 – transfer function B/A in format (f,Y), 4 – transfer function A in format (f,Y), 5 – coherence B/A in format (f, Y).
f0	d	1	x	start frequency in Hz.
df	d	1	x	frequency spacing in Hz.
t0	time	1	x	start time in GPS nsec.
BW	d	1	x	measurement bandwidth in Hz.
Window	i	1	x	0 – uniform (no window), 1 – Hanning, 2 – Flat-top, 3 – Welch, 4 – Bartlett, 5 – BMH, 6 – Hamming, 7 – Kaiser. default: 1
AverageType	i	1		0 – fixed number, 1 – running (exponential weight).
Averages	i	1		number of averages.
ChannelA	ch	1		name of A channel
ChannelB[M]	ch	1		name(s) of B channel
N	i	1	x	number of points.
M	i	1	x	number of transfer/coherence functions.
	zf f	M×N +N M×N +N		transfer function in format (Y), transfer function in format (f,Y), coherence in format (Y), coherence in format (f,Y).

Diagnostics tests will generally use subtype 3 for a transfer function and subtype 5 for the coherence. Typically, they are both stored as a set of functions with one A channel and multiple B channels.

8 LIST OF COEFFICIENTS

A sine response measurement can yield multiple transfer coefficients which are stored in two dimensional arrays. One of the dimension always represents the multiple measurement points, whereas the other dimension may represent the multiple excitation points, the harmonic order or the modulation product terms.

Name	Type	Dim	Man.	Description
				Coefficients
Subtype	i	1	x	0 – transfer coefficients in format (Y), 1 – harmonic coefficients in format (Y), 2 – intermodulation product in format (Y), 3 – coherence coefficients in format (Y), 4 – transfer coefficients in format (f, Y), 5 – harmonic coefficients in format (f, Y), 6 – intermodulation product in format (f, Y), 7 – coherence coefficients in format (f, Y), 8 – transfer matrix in format (Y).
t0	time	1	x	start time in GPS nsec.
BW	d	1	x	measurement bandwidth in Hz.
AverageType	i	1		0 – fixed number, 1 – running (exponential weight), 2 – running (accumulative).
Averages	i	1		number of averages.
ChannelA[M']	ch	1		channel name corresponding to M'-th readback point.
ChannelB[N]	ch	1		channel name corresponding to N-th detection point.
N	i	1	x	number of detection points (A + B channels, except subtype 8 which only counts B channels)
M	i	1	x	number of frequency points: f_1, f_2, f_M: transfer/coherence coefficients, 0, f, 2 f, 3 f, M f: harmonic coefficients (the 0 frequency is used for storing the total harmonic distortion), f_1, f_2, |f_1 – f_2|, f_1 + f_2: intermodulation product
Unit[N]	st	1		physical unit.
	zf zf zf f zf zf zf zf f	M×N M×(N+1)		transfer coefficients in format (Y), harmonic coefficients in format (Y), intermodulation product in format (Y), coherence coefficients in format (Y), transfer matrix in format (Y), transfer coefficients in format (f, Y), harmonic coefficients in format (f, Y), intermodulation product in format (f, Y), coherence coefficients in format (f, Y).