

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

<b>Document Type</b>	<b>LIGO-T030293-00-Z</b>	September 26, 2003
<b>MATLAB "Channel" Class Software Specification</b>		
Keith Thorne		

*Distribution of this draft:*

**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](mailto:info@ligo.caltech.edu)

**Massachusetts Institute of Technology**  
**LIGO Project - MS 20B-145**  
**Cambridge, MA 01239**  
Phone (617) 253-4824  
Fax (617) 253-7014  
E-mail: [info@ligo.mit.edu](mailto:info@ligo.mit.edu)

WWW: <http://www.ligo.caltech.edu/>

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Approach</b>	<b>3</b>
<b>3</b>	<b>Channel Class Specification</b>	<b>3</b>
3.1	Requirements . . . . .	3
3.2	Channel Class Interface . . . . .	4
<b>4</b>	<b>Implementation Plan</b>	<b>5</b>
<b>5</b>	<b>Channel Structure for Matlab Compiler use</b>	<b>5</b>
5.1	Channel Structure Interface . . . . .	5

# 1 Introduction

A Matlab class, termed "Channel", provides a way to accessing the time-series data from gravity-wave interferometers (such as LIGO) from within Matlab which "abstracts" them as objects. In this way, the particular details of how the time-series data are obtained from LDAS, LDR or otherwise are hidden.

## 2 Approach

To implement this new access, a Matlab class called "Channel" will be used. Matlab classes are similar to C++ classes. This "Channel" class will be used to create objects with the properties of that class, which act a lot like structures. A software requirements specification was first created for the class. This specification forms the basis for the component tests to be used to verify an implementation of the software. Each 'shall' is intended to be a testable condition. This specification is followed by an explanation of the class interface to be created.

At present, the Matlab Compiler does not support Matlab object-oriented coding. To support this usage, a Channel structure similar in behavior to the class will also be implemented.

## 3 Channel Class Specification

The Channel class will provide an association to a particular interferometer data stream that hides exactly how that connection is provided. This 'abstraction' will allow the frame data retrieval method to be changed without requiring a change to the main analysis code.

### 3.1 Requirements

The software shall allow creation of Channel objects which are each associated with a specific Interferometric Gravity-Wave Detector (IGWD) channel name. The software will support the following properties for a Channel object: 'name', 'statusCode', 'sys', 'type' and 'gpsStart'. The 'name' property will be a string that the software shall set to the channel name used when creating the Channel object. The 'statusCode' property will be an integer that the software shall set to 0 if the Channel object was created successfully. The software shall not allow 'name' and 'statusCode' to be set externally. The 'sys' property will be a string that corresponds to the LDAS 'sys' parameter. The 'type' property will be a string that corresponds to the LDAS 'type' parameter. The software shall set the 'sys' and 'type' properties to default values of 'LDR' and 'RDS\_R\_L1' when a Channel object is created. The software shall allow the 'sys' and 'type' properties to be set externally.

The 'gpsStart' property will be an integer in units of GPS seconds which is the starting location for all time series derived from this channel. The software shall set the 'gpsStart' property when a Channel object is created if a gpsStart is passed as a parameter. If no gpsStart parameter is passed when creating a Channel object, the software shall set the 'gpsStart' to a default value of 0. The software shall allow the 'gpsStart' property to be set externally.

The software will handle channel names of the format defined in Appendix D of document LIGO-T970130-E. If the channel name is not of this format when a Channel object is created, the software shall indicate the error by setting the 'statusCode' property to 1.

The software shall allow the creation of time-series data vectors from Channel objects. The software will allow time-series data to be created based upon inputs of 'gpsOffset' and either a 'duration' value or a 'gpsEnd' value. The 'gpsOffset' and 'gpsEnd' values will be integers in units of GPS seconds. The 'duration' value will be an integer in units of seconds. The software shall start the time-series data vector at a GPS time of 'gpsOffset' + the 'gpsStart' property of the specific Channel object. The software shall create a time-series data vector which covers a range of 'duration' seconds. If the 'gpsEnd' input is provided instead of 'duration', the software shall set the duration to be 'gpsEnd' - 'gpsOffset'. The length of the time-series data vector will be 'duration' / times sample rate of the data available for the channel. When creating the time-series data vector, the software shall use the 'name' and 'type' properties of the specific Channel object.

A few error conditions are defined for attempt creation of time-series data vectors. If any of the input values are invalid (non-existent Channel object, gpsOffset missing, duration = 0, etc.), the software shall set the output vector to an empty array. If the software is not able to find frame data files which cover the time-frame requested (based upon gpsStart, gpsOffset and duration) for the values of 'name' and 'type' properties, then the software shall set the output vector to an empty array. If the software is not able to open a frame data file it has the path to, it shall set the output vector to an empty array.

## 3.2 Channel Class Interface

The Matlab syntax for creating a new Channel object will be

```
chanObj = Channel(nameString, <gpsStartTimeValue>)
```

Note that gpsStartTimeValue is an optional input. The properties can be retrieved using the 'get' function.

```
status = get(chanObj, 'statusCode')
```

Some properties can be changed using the 'set' function, which returns the new value of that property.

```
chanObj = set(chanObj, 'type', 'RAW')
```

The properties can also be retrieved using '.' notation.

```
status = chanObj.statusCode
```

Some properties can also be changed using the '.' notation.

```
chanObj.type='RAW'
```

The Matlab syntax for getting a time-series data vector from a Channel object will be in one of two forms. Either the (s,d) format:

```
newSeries = chanObj(gpsOffset, duration)
```

where 'chanObj' is a previously-created Channel object, 'gpsOffset' is the offset of the start of the time-series from the 'gpsStart' property of the channel, and 'duration' is the length of the time-series in whole seconds. Or in the (s:e) format:

```
newSeries = chanObj(gpsOffset:gpsEnd)
```

where 'chanObj' is a previously-created Channel object, 'gpsOffset' is the offset of the start of the time-series from the 'gpsStart' property of the channel, and 'gpsEnd' defining the duration as 'gpsEnd' - 'gpsOffset'.

## 4 Implementation Plan

The initial implementation will support the only the LDR retrieval method using a database query (using MySQL) of the LDR file list. As such, the 'sys' property will not be implemented at this time. If support for the LDAS Frame Cache query retrieval method is needed in the future, it will be implemented at that time.

## 5 Channel Structure for Matlab Compiler use

As of Matlab Compiler 3.0, Matlab object-oriented code (i.e. classes) can not be used in compiled code used for stand-alone libraries and executables. To allow syntaxes similar to that used for the Channel class, the software will implement a Channel structure using the 'chanstruct' and 'chanvector' functions.

The 'chanstruct' function shall return a Matlab structure with the same properties ('name', 'statusCode', 'sys', 'type' and 'gpsStart') as the Channel class. The 'chanvector' function shall return a time-series data vector which is identical to that retrieved from a Channel class object.

### 5.1 Channel Structure Interface

The Matlab syntax for creating a Channel structure will be

```
chanStr = chanstruct(nameString, <gpsStartTimeValue>)
```

Note that gpsStartTimeValue is an optional input. The Channel structure properties can also be retrieved using '.' notation.

```
status = chanStr.statusCode
```

The Channel structure properties can also be changed using the '.' notation.

```
chanStr.type='RAW'
```

The Matlab syntax for getting a time-series data vector from Channel structure will be in one of two forms. Either the (s,d) format:

```
newSeries = chanvector(chanStr, gpsOffset, duration)
```

where 'chanStr' is a previously-created Channel structure, 'gpsOffset' is the offset of the start of the time-series from the 'gpsStart' property of the channel, and 'duration' is the length of the time-series in whole seconds. Or in the (s:e) format:

```
newSeries = chanvector(chanStr, gpsOffset: gpsEnd)
```

where 'chanStr' is a previously-created Channel object, 'gpsOffset' is the offset of the start of the time-series from the 'gpsStart' property of the channel, and 'gpsEnd' defining the duration as 'gpsEnd' - 'gpsOffset'. With the 'chanvector' function, the sample rate (in samples/second) can also be retrieved as an optional output parameter.

```
[newSeries, sampRate] = chanvector(chanStr, gpsOffset, duration)
```