



Update on LIGO Data Analysis System (LDAS)

6th LIGO Scientific Collaboration Meeting
March 16th-18th, 2000
LIGO Livingston Observatory

Kent Blackburn
LIGO Laboratory
California Institute of Technology



Outline

- Usage Overview
- On-Site Installation Activities
- Development Status
- Recent Software Enhancement Activities
- Current Software Development Activities
- Documentation
- Software Development Collaborations(*A.L.*)



Usage Overview

❑ Clients: User Interfaces

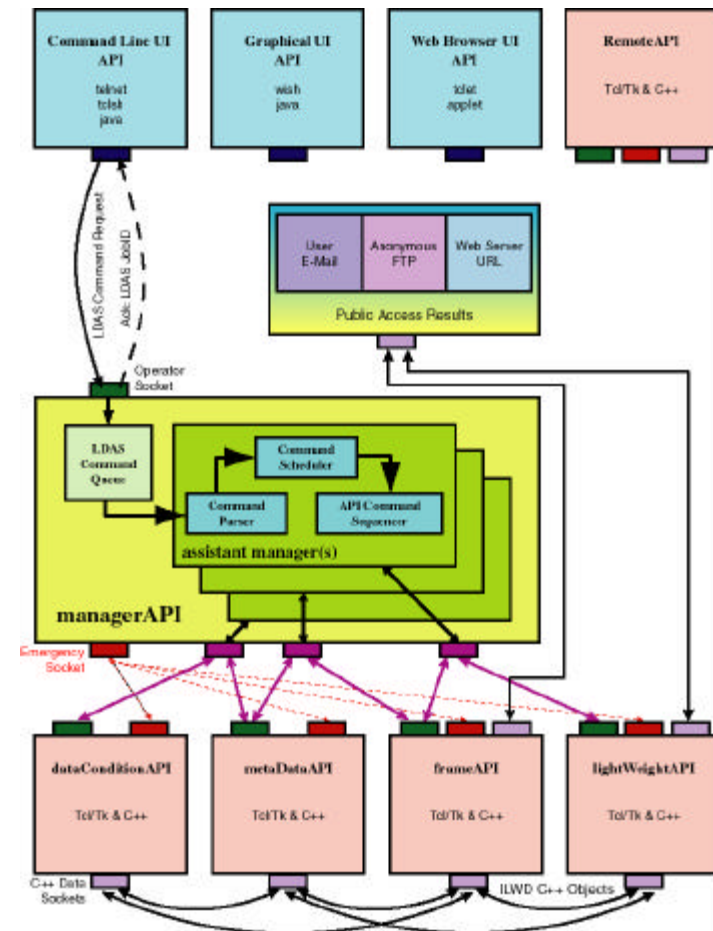
- ⇒ Three types of UIs:
 - ⇒ command line
 - ⇒ graphical
 - ⇒ web browser
- ⇒ Send LDAS Command requests to “Operator Socket” on managerAPI and receive acknowledgement containing “jobID” and other details associated with request.

❑ Server: managerAPI

- ⇒ Uses 3-10 assistant managers to handle each LDAS Command Request independently.

❑ Servlets: LDAS APIs

- ⇒ Internal LDAS APIs & RemoteAPI using LDAS interfaces.
- ⇒ Analysis flow control managed by “meta-scripts” associated with LDAS Commands.
- ⇒ Data communicated internally as ILWD objects
- ⇒ Data shared externally as Frames & LIGO_LW(XML).





LDAS - Hanford

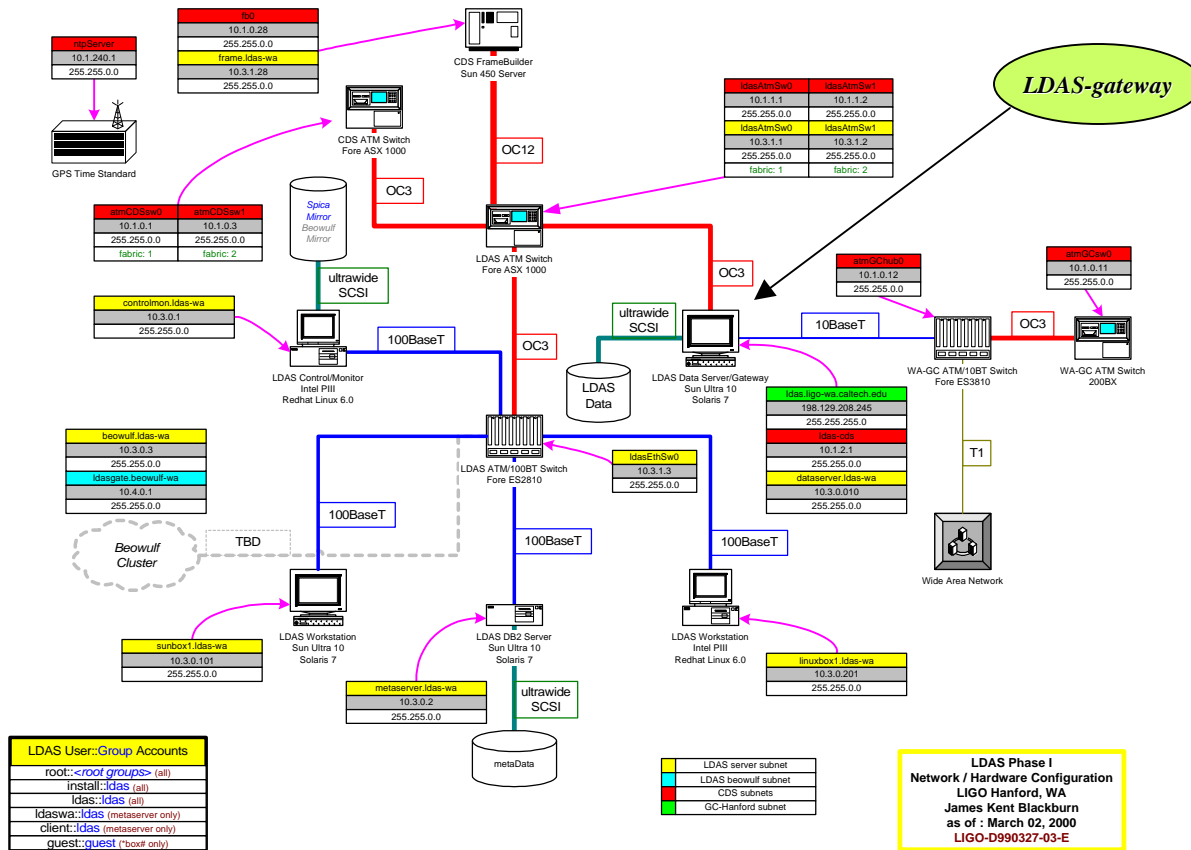
□ Hanford: Jun, Jul, Sep, 99

- ⇒ **dataserver:** *managerAPI / frameAPI*
 - ⇒ Sun Ultra 10 + 18GB HD
- ⇒ **metaserver:** *metadataAPI, DB2 Server*
 - ⇒ Sun Ultra 10 + 18GB HD
- ⇒ **controlmon:** *internal software mirror*
 - ⇒ Pentium III PC + 18GB HD
- ⇒ **Fore 1000 ATM switch**
- ⇒ **Fore 2810 fast ethernet switch**
- ⇒ **1 sun & 1 linux workstations**
 - ⇒ Sun Ultra10
 - ⇒ Pentium III PC
- ⇒ **video/keyboard switch**





Hanford LANs + gateways





LDAS - Livingston

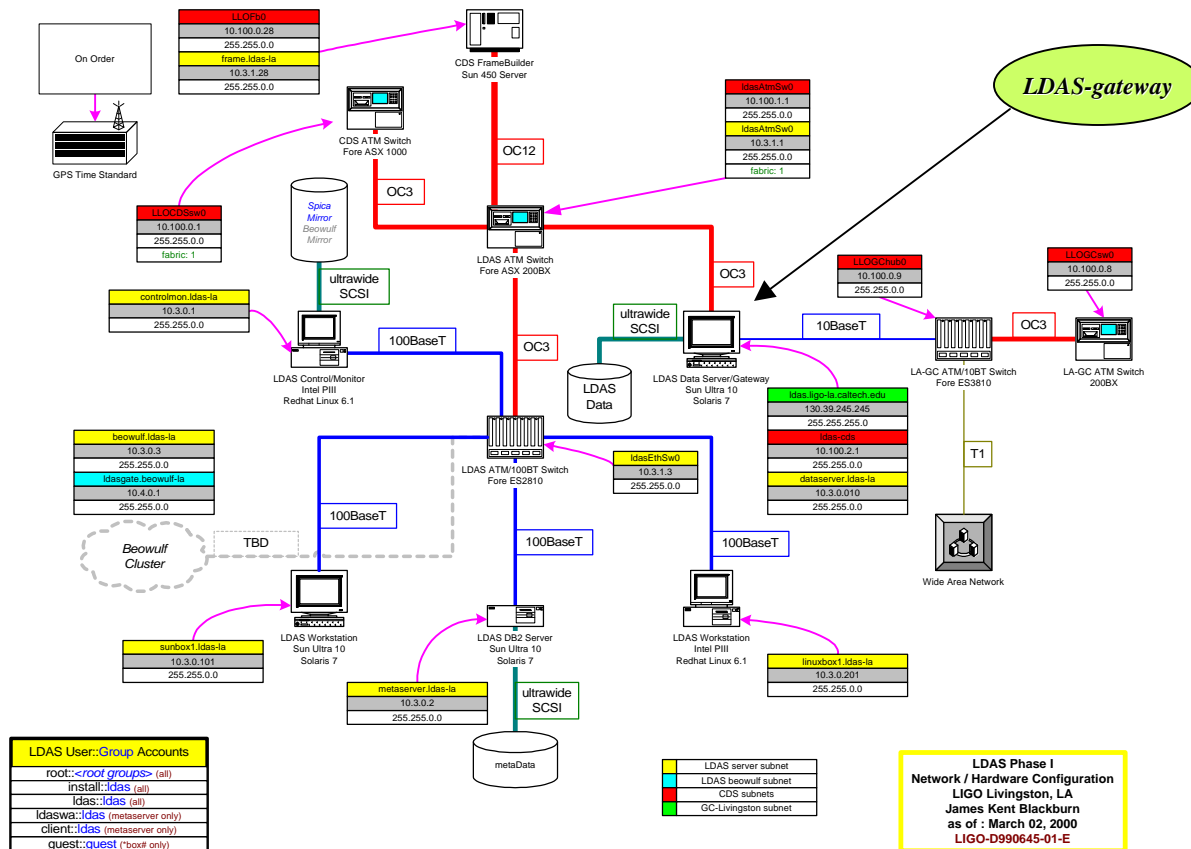
□ Livingston: Oct, 99

- ⇒ **dataserver:** *managerAPI / frameAPI*
 - ⇒ Sun Ultra10 + 18GB HD
- ⇒ **metaserver:** *metadataAPI, DB2 Server*
 - ⇒ Sun Ultra10 + 18GB HD
- ⇒ **controlmon:** *internal software mirror*
 - ⇒ Pentium III PC + 18 GB HD
- ⇒ Fore 200BX ATM switch
- ⇒ Fore 2810 fast ethernet switch
- ⇒ 1 sun & 1 linux workstations
 - ⇒ Sun Ultra10
 - ⇒ Pentium III PC
- ⇒ video/keyboard switch





Livingston LANs + gateways



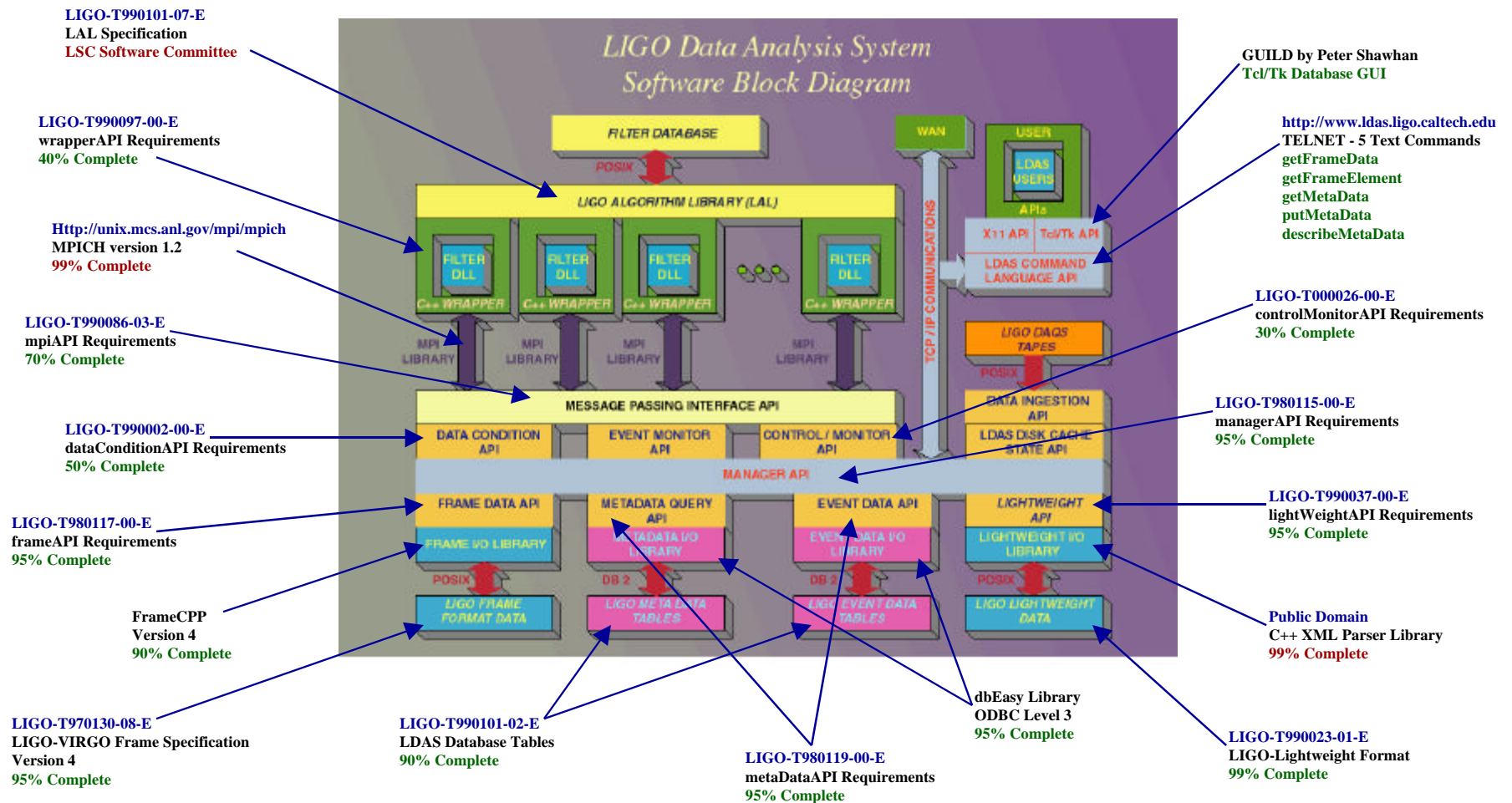


On-Site Software Administration

- ❑ Mirror LDAS Development Server: spica.ligo.caltech.edu
 - ⇒ compilers(*gcc-2.95-2*), interpreters(*Tcl/Tk 8.3*), debuggers(*ddd*), editors(*xemacs*), & many others!
 - ⇒ LDAS **cv**s Repository
 - ⇒ LDAS release installations
 - ⇒ all software managed using *stow* utility
 - ⇒ web based problem tracking report system: *GNU gnats*
- ❑ Unix configurations managed with **cfengine** utility.
- ❑ Limited unix accounts: {*install*, *ldas*, *ldasdb*, *dbclient*, *guest*, *reboot*, & *root*}
- ❑ Gateway provides access to LDAS “private net.”
 - ⇒ LDAS managerAPI, frameAPI, lightWeightAPI
 - ⇒ anonymous ftp
 - ⇒ web service
 - ⇒ ssh services
- ❑ Account logins using **ssh** & **scp** utilities only!



Implementation Status



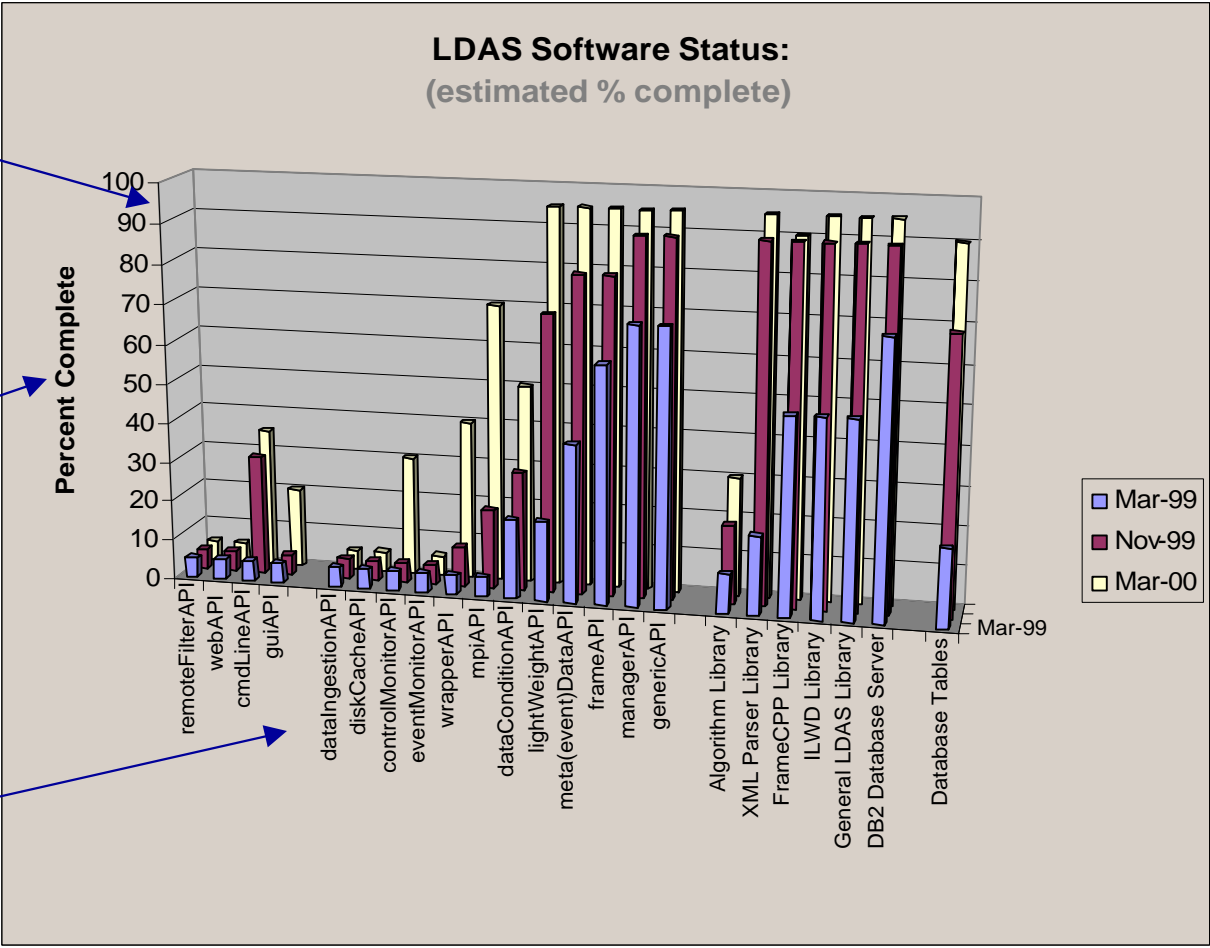


Past Year's Development

I use 95%
to separate
development
from
maintenance

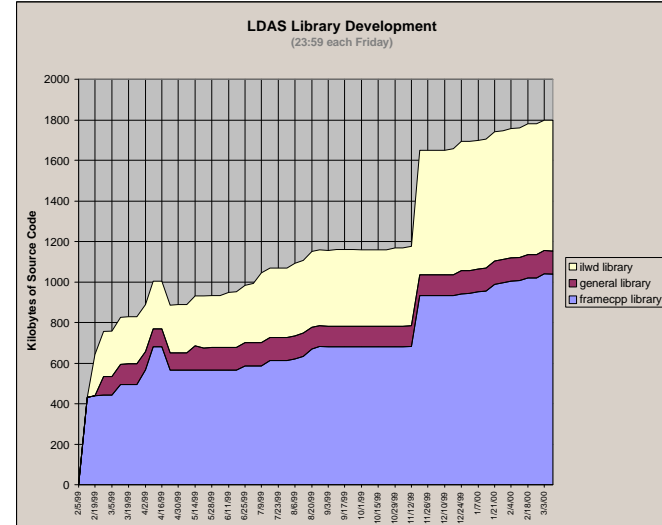
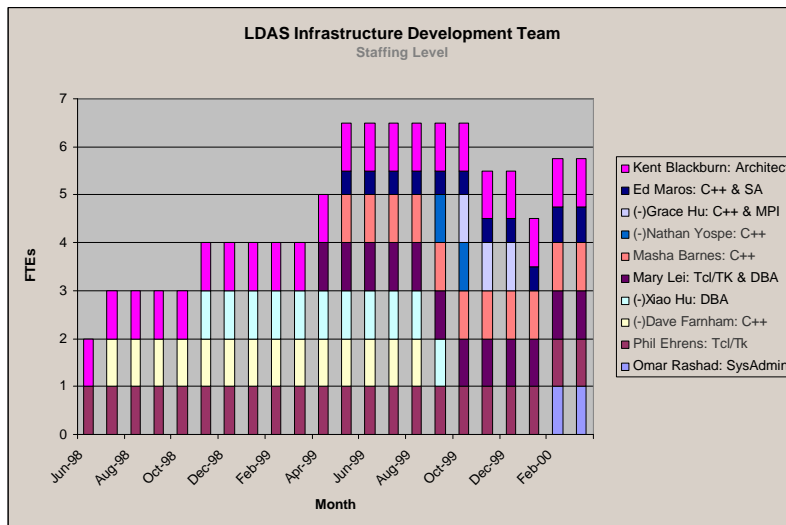
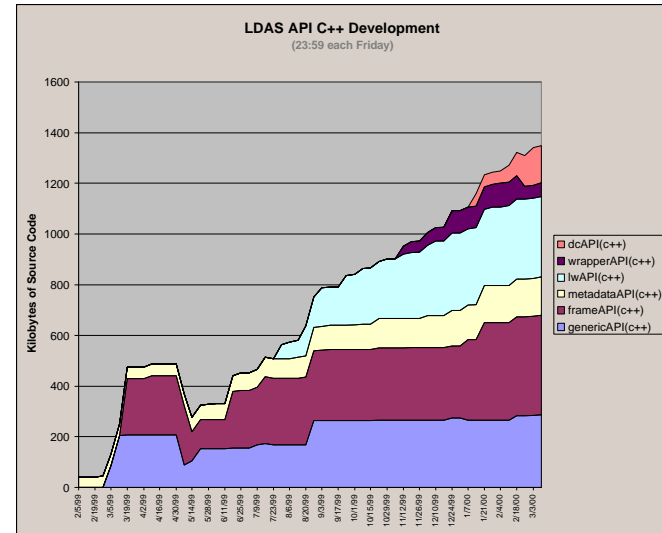
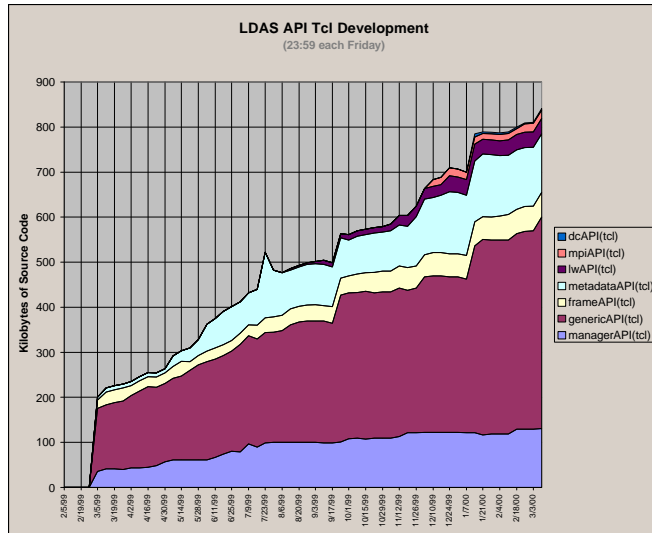
Percent complete
non-linear
with time

Not all APIs
variable complexity
created equally



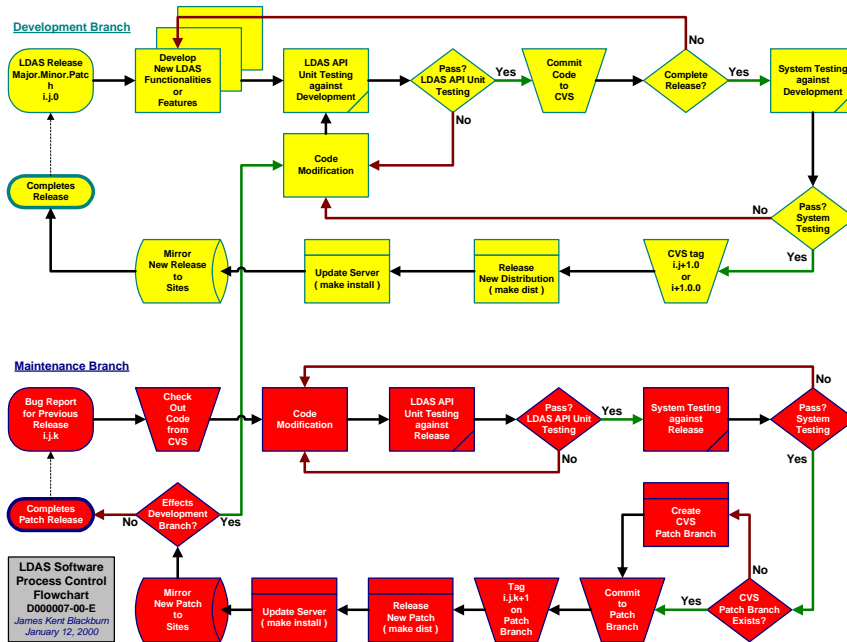


Code Growth Related To Staffing Levels

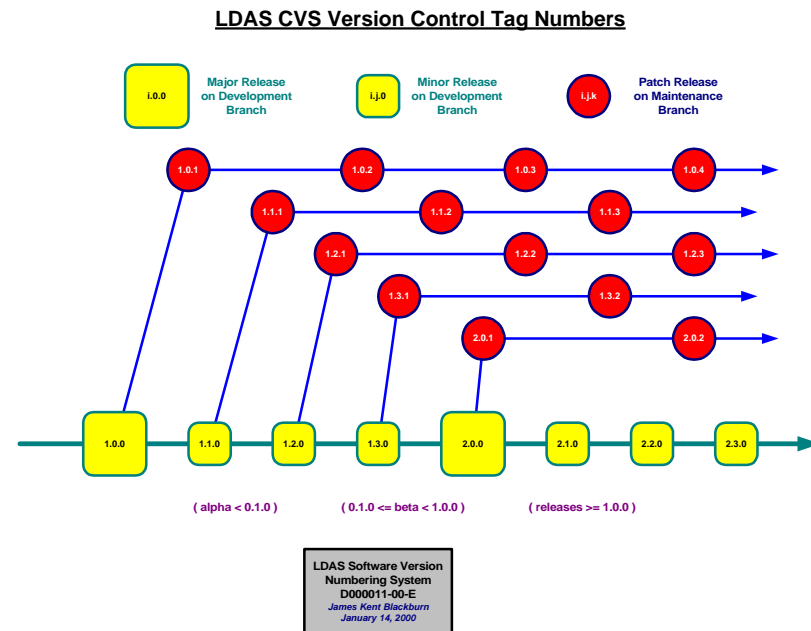




Improved Product Control



Flow chart controls how software development & testing are executed!



Tree diagram controls release tag numbers found in CVS repository!

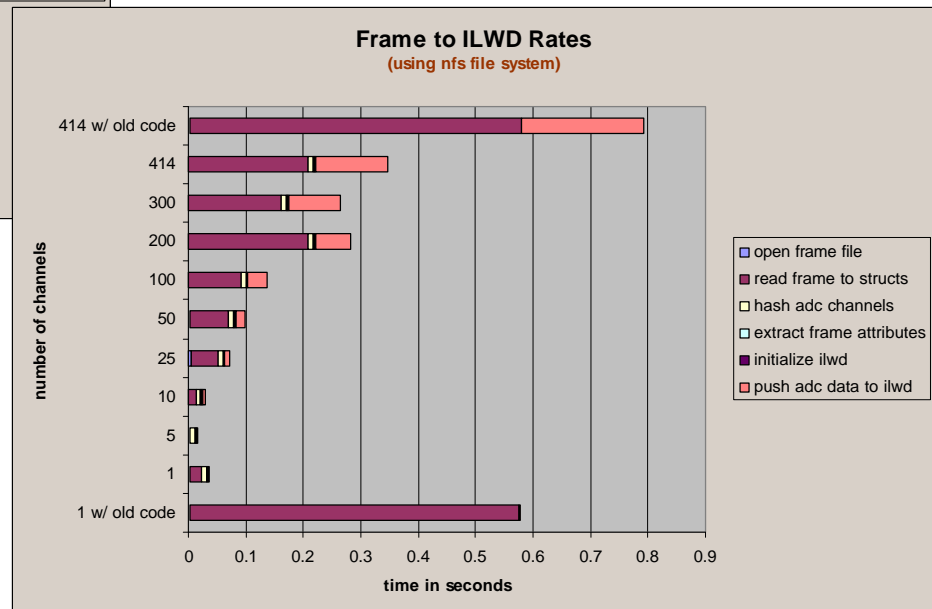
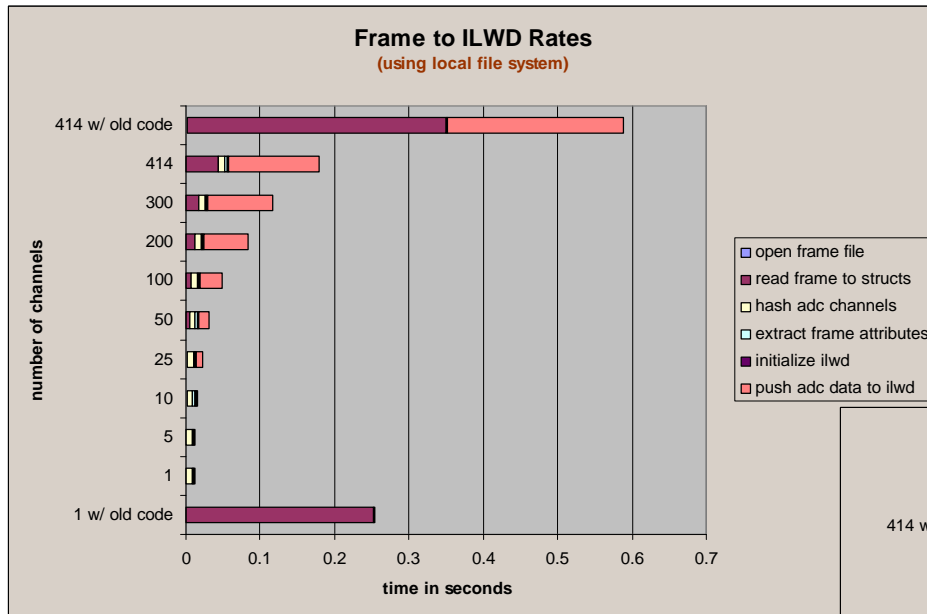


Recent Software Enhancement Activities

- ❑ New LDAS Build Script:
 - ⇒ *New Pre-Build, Build, & Post-Build functionality based on bsh, automake, autoconfig & make*
- ❑ Frame Format:
 - ⇒ *New Frame Specification: version 4.0!*
 - ⇒ *FrTOC “table of contents” indexes into frame for faster data access from file!*
 - ⇒ *This also removes necessity to load full frame!*
- ❑ FrameCPP:
 - ⇒ *Added static index I/O buffers for frames with same run number to remove necessity to parse every LIGO frame increasing read/write speeds ~ **1-10 fold** over previous versions!*
 - ⇒ *Days away from releasing version 4.0 compliant FrameCPP Library!*
- ❑ ILWD C++ Object Socket Transmission:
 - ⇒ *Added more intelligence and buffering to object space code, increase speed of object oriented distributed data transmissions by **10-1000 times** that of previous software!*
 - ⇒ *Now able to send and receive ILWD objects between Sun and Linux!*
- ❑ ODBC Level 3 client library (dbEasy):
 - ⇒ *Upgraded IBM’s DB2 Server: version 6.1*
 - ⇒ *Added ODBC Level 3 calls to dbEasy (multi-row inserts) resulting in a **10-100 fold** improvement in database ingestion rates!*

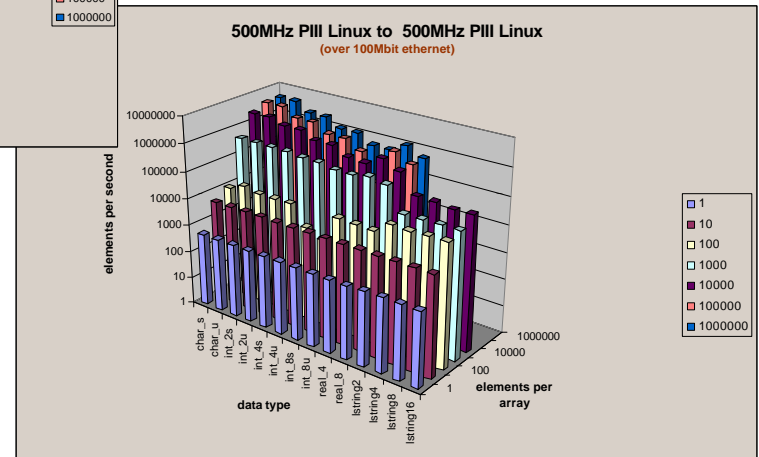
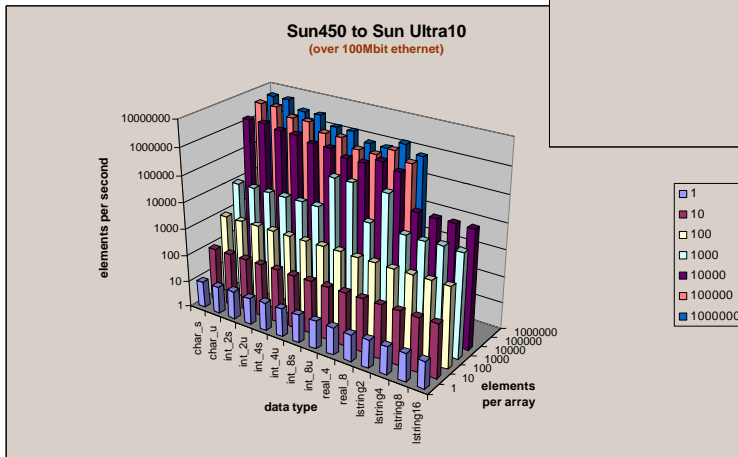
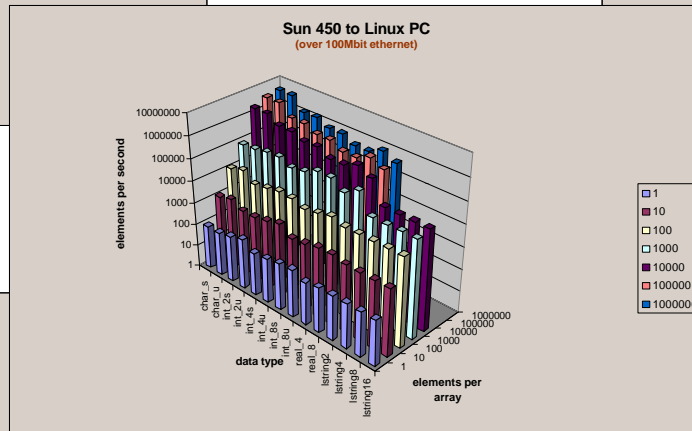
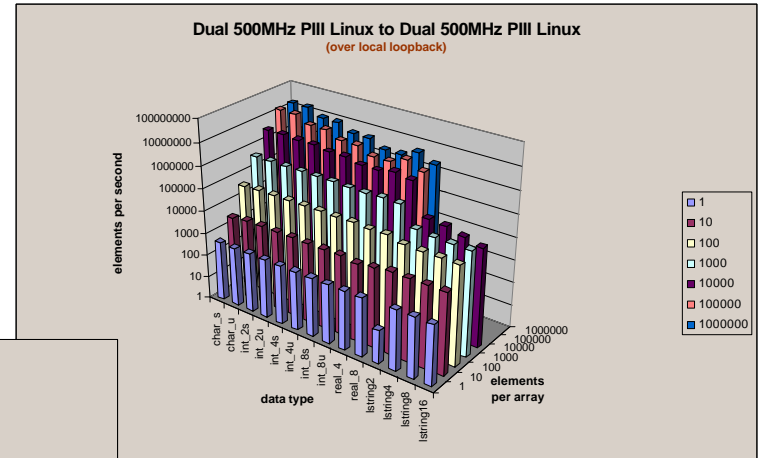
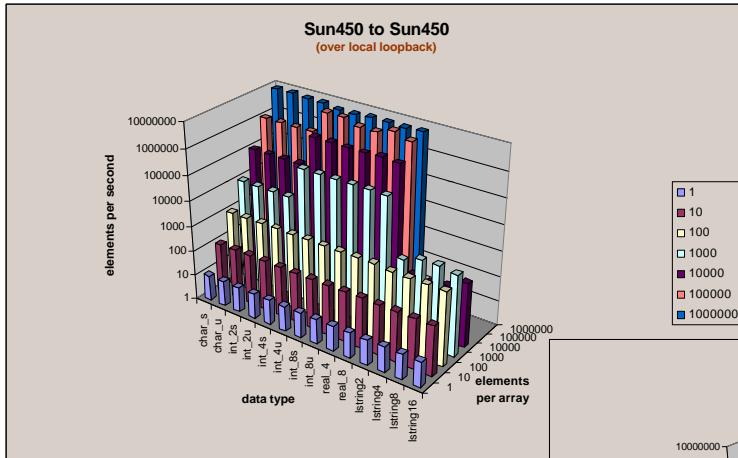


Frame To ILWD Tests



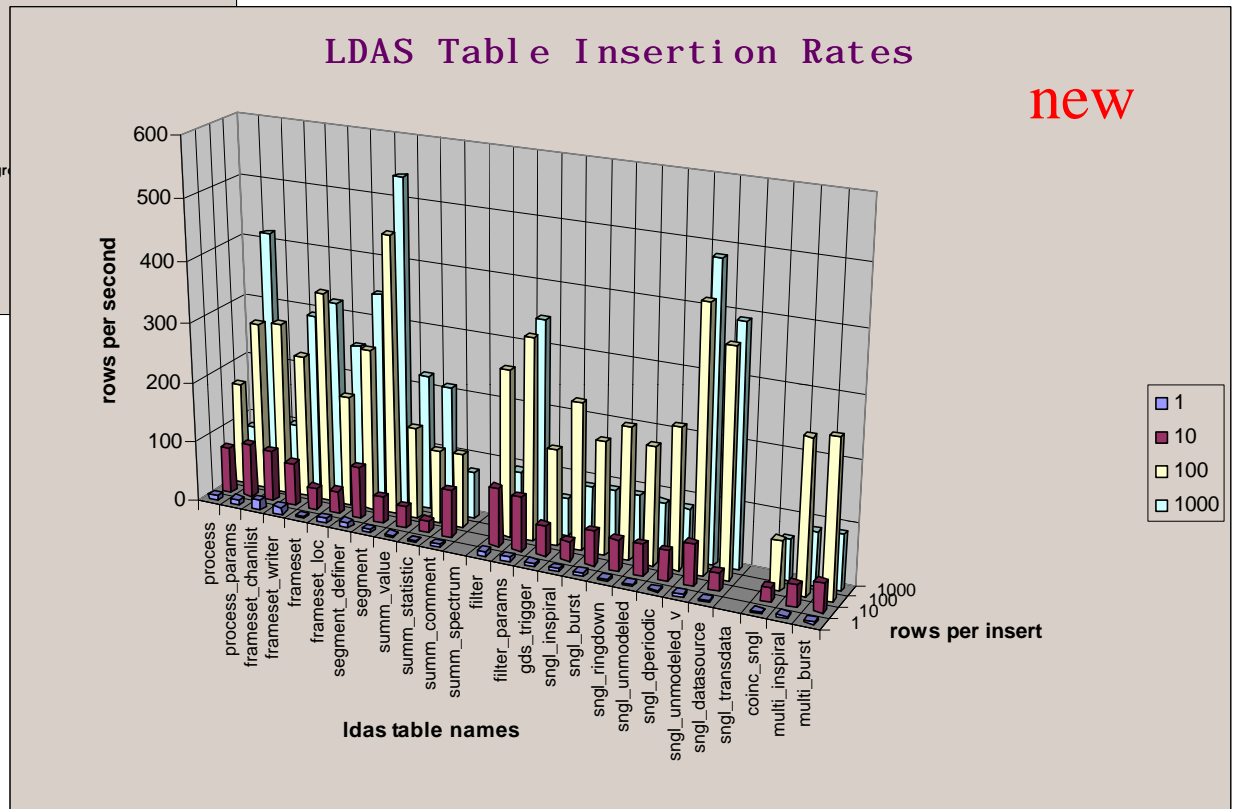
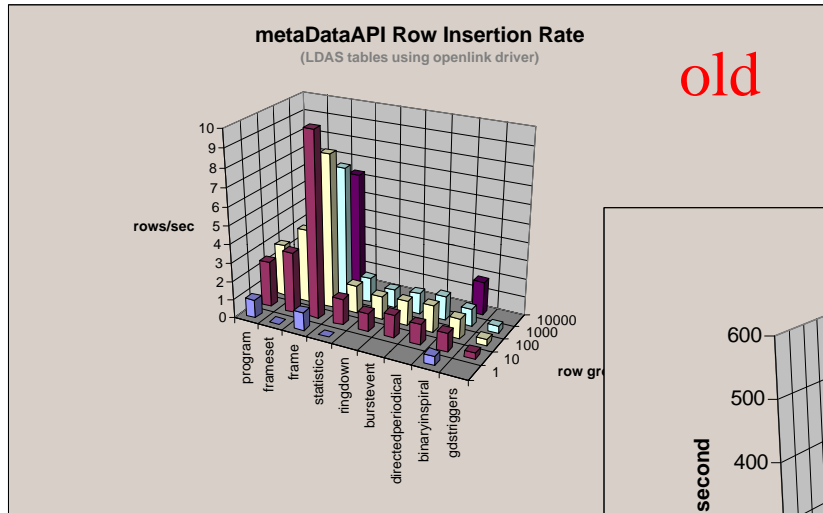


ILWD Socket Object Tests





Metadata Table Insert Tests

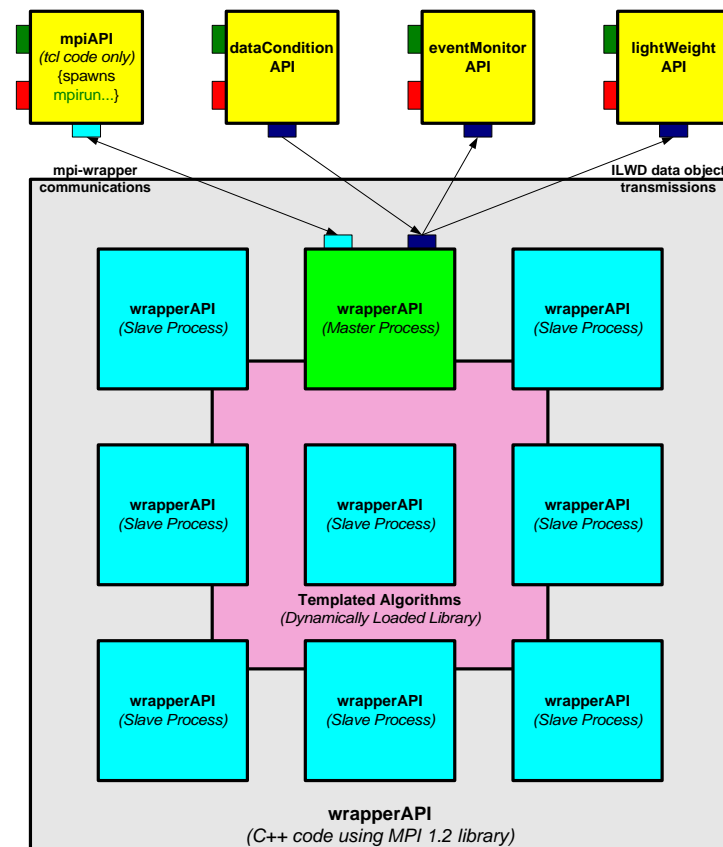




Current Software Activities

□ Parallel Computation!

- ⇒ **dataConditionAPI:**
 - X pre-conditions data prior to parallel processing
 - X collaborating with ANU, PSU, & UTB LSC members
- ⇒ **mpiAPI:**
 - X initiates parallel jobs
 - X manages dynamic load balancing
 - X listens to parallel status reports
- ⇒ **wrapperAPI:**
 - X parallel analysis driver code
 - X dynamically loads algorithm libraries
 - X collaborating with UWM LSC members
- ⇒ **controlMonitorAPI:**
 - X GUI interface to LDAS system
 - X displays status of parallel jobs
 - X allows additional control of parallel jobs after start-up
 - X provides views of all LDAS log files
 - X provides some simple system administration tasks.
- ⇒ **eventMonitorAPI:**
 - X gathers results from parallel jobs
 - X commits database results to metaDataAPI
 - X initiates new post-parallel processing on resultant data





LDAS Web Documentation

Web Pages (*about to receive a face lift!*)

- ⇒ **Caltech:** <http://www.ldas.ligo.caltech.edu>
- ⇒ **Hanford:** <http://www.ldas.ligo-wa.caltech.edu>
- ⇒ **Livingston:** <http://www.ldas.ligo-la.caltech.edu>

User Interfaces

Tcl/Tk - *TclDoc*

C++ - *Perceps*

Log Files - *LDAS APIs*

Problem Tracking System



New LDAS Home Page Preview

LDAS Software Documentation Index
Modification Date: 99.09.15

Developer Documents

Tel Code:

- Generic API
- genericAPI.tcl
- control
- data
- event
- frame
- lightweight
- manager
- mpi
- wrapper

Manager API

- managerAPI.html
- manager.tcl
- control.tcl
- data.tcl
- event.tcl
- frame.tcl
- lightweight.tcl
- manager.tcl
- mpi.tcl
- wrapper.tcl

Metadata API

- metadata.tcl
- data.tcl

Lightweight API

- lightweight.tcl

Generic Wrapper for APIs

- LDASWrapper

C++ Code:

- Generic API
- genericAPI.so source docs
- frame API
- frameAPI.so source docs
- Lightweight API
- lightweightAPI.so source docs
- general library
- general.so source docs
- framewrap library
- framewrap.so source docs
- lib library
- lib.so source docs
- Other Source Libraries

Swig

- Index to All SWIG Documents
- SWIG Docs in HTML
- SWIG Docs in PDF
- SWIG Docs in PostScript

User Commands

- LDAS User Commands
- generic User Commands
- metadata User Commands
- data User Commands

Developer News

- Example .src file with comments
- Example src file with comments
- Example API level in src with comments
- Primary IRL Documents

LDAS Log File Directories By Site

Hanford

- TopLevel
- Manager API logs at Hanford
- Frame API logs at Hanford
- Metadata API logs at Hanford

Livingston

- TopLevel
- Manager API logs at Livingston
- Frame API logs at Livingston
- Metadata API logs at Livingston

CalTech

- TopLevel
- Manager API logs at CalTech
- Frame API logs at CalTech



LIGO Data Analysis System

Welcome to the LDAS web site at Caltech (@ spica.ligo.caltech.edu)

Visit our other web sites:

- [LIGO Hanford Observatory](#)
- [LIGO Livingston Observatory](#)

LDAS Development at Caltech

- [Software Development Index](#)
- [LDAS Bulletin Board](#)
- [Problem Reporting System](#)

Log Files at spica.ligo.caltech.edu

	Current	Previous	Past
controlmanager API	View	View	View
datacondition API	View	View	View
eventmanager API	View	View	View
frame API	View	View	View
lightweight API	View	View	View
manager API	View	View	View
mpi API	View	View	View
wrapper API	View	View	View

[Direct comments to ???](#)



Telnet User Interface

A screenshot of a Microsoft Internet Explorer browser window. The title bar reads "LDAS User Commands - Microsoft Internet Explorer". The address bar shows the URL "http://spica.ligo.caltech.edu/ldas/ldas-0.0/doc/userAPI/MiniIndex.html". The page content includes the LIGO logo in red, a section titled "The LDAS User Commands", and a list of user commands. The first command is "ldasJob" with its syntax: "ldasJob (-name {} -password {} -email {}) { userCmd -opt1 {} ... }". The page also explains the format of a Telnet command and lists three classes of commands: "getData", "putData", and "descData".

LDAS User Commands

All user commands have the form:

```
ldasJob ( -name {} -password {} -email {} ) { userCmd -opt1 {} ... }
```

Which is in the format of a [Tel](#) command, named **ldasJob**, with two required arguments:

1. A Tel list of **user information** consisting of username, password, and e-mail address. All fields must be filled or the command will be rejected.
2. A **user command** in the form of a Tel list, consisting of the name of the user command (userCmd in the example) for which there exists a 'meta' macro file, and the required options with their argument lists.

An argument **must be provided to every option field** for any given command or the command will be rejected. Some option fields will accept a 'null' argument consisting of a matching pair of braces "{}" with no interposed space.

Meta macros consist of a prototype declaration of the arguments for the given user command, and a template describing the calling order of API specific blocks of Tel code which are concatenated into a larger block comprising the complete request, which can then be distributed by the assistant manager for interpretation by the low level APTs.

These API specific blocks are maintained as API specific macro files consisting of immediately interpretable Tel code.

There are a number of *LDAS User Commands* for manipulating and retrieving LDAS data:

- [getData](#) class of commands
- [putData](#) class of commands
- [descData](#) class of commands



Telnet UI Examples

```
http://psds.ligo.caltech.edu/ldas/ldas-0.8/docs/psdsAPI/ldas/getData.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://psds.ligo.caltech.edu/ldas/ldas-0.8/docs/psdsAPI/ldas/getData.html#psdsdata

• getRawData

Retrieves raw data stored in the LDAS database in response to the query given by the user command. Returns in several
formats either by attaching the resulting files to an e-mail (-returnformat email) or by returning an e-mail with a URL pointing
to the location of the results (-returnformat http or ftp). It is also possible to have the data directly delivered to a local port,
which will be the default method for the get to be completed user API.

ldasJob [-name "username" -password ***** -email "user@bobbar.edu"] [getRawData -
returnprotocol URL -returnformat data_format -sqlquery 'SQL text']

Option Descriptions

-returnprotocol: http ftp mailto file port

The argument to the -returnprotocol option conforms to the usual browser
conventions for URI's for determining the location of the results of the user
request.
The mailto and port options result in an active return of data to the user.
The http, ftp, and file options cause a URL pointer to the location of the results
to be returned via e-mail.
The possible formats of the argument are:

• http://host:port/dir/...
• ftp://host:port/dir/...
• mailto:user@address.dom
• file:/path/... (note, only one "/" )
• port:hostname:portnumber

NOTE: Embedded spaces in the argument to the -returnprotocol option will cause the
request to fail.

-returnformat: raw raw_xml

The argument of the -returnformat option is the name of the data type to use in
formatting the result of the user request.
The possible output formats which the system can produce, which include binary frames,
shred text, and LIGO_XML (an XML format) for the result data.

-sqlquery: (any valid SQL statement)

The argument of -sqlquery option is any valid sql statement supported by the LDAS
database server. If the query turns out to be invalid, a database error
message is returned.

Example of getRawData command

o For LDAS registered user job to get all data from table rawframe in shred format and have results delivered by mail,
return.

ldasJob [-name "job" -password ***** -email "job@bobbar.edu"] [getRawData -
returnprotocol "job@bobbar.edu" -returnformat shred -sqlquery "select * from rawframe"]
```

```
http://psds.ligo.caltech.edu/ldas/ldas-0.8/docs/psdsAPI/ldas/getData.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://psds.ligo.caltech.edu/ldas/ldas-0.8/docs/psdsAPI/ldas/getData.html#psdsdata

• getFrameData

Retrieves HD or reduced raw frame data in several formats either by attaching the resulting files to an e-mail (-returnformat email)
or by returning an e-mail with a URL pointing to the location of the results (-returnformat http or ftp). It is also possible to have the data directly delivered to a local port, which will be the default method if the user to be
completed user API.

Calling convention (all on a single line):

ldasJob [-name "username" -password ***** -email "user@bobbar.edu"] [getFrameData -returnprotocol URL -returnformat data_format -time
(T1:tm) -interferometer (T1:tm) -dataquery (T1:tm)]

Option Descriptions

-returnprotocol: http ftp mailto file port

The argument to the -returnprotocol option conforms to the usual browser conventions for URI's for determining the location
of the results of the user request.
The mailto and port options result in an active return of data to the user.
The http, ftp, and file options cause a URL pointer to the location of the results to be returned by e-mail.
The possible formats of the argument are:

• http://host:port/dir/...
• ftp://host:port/dir/...
• mailto:user@address.dom
• file:/path/... (note, only one "/" )
• port:hostname:portnumber

NOTE: Embedded spaces in the argument to the -returnprotocol option will cause the request to fail.

-returnformat: frame raw_xml

The argument of the -returnformat option is simply the name of the data type to use in formatting the result of the user
request.
The possible output formats which the system can produce, which include binary frames, shred text, and LIGO_XML (an XML
format).

-time: ( S2345)789 S2347894-S2367898 )

The argument of the -time option is a Tm list of times, which may consist of mixed individual times and ranges of times.
The times in the list MUST be valid gpc times! (Always 9 digits in length, and greater than 600000000 in any case.)

-interferometers: ( H1 H2 L1 C1 U1 G1 T1 )

The argument of the -interferometers option is a Tm list of interferometers (ifso) corresponding to the file name prefix
of the frame files created from that interferometer.

-dataquery: ( I J 210-217 999 and_000-10 )

The argument of the -dataquery option is a Tm list of adc channel names and/or a numerical list (including ranges) of
channel indices, which are NOT requested to point to specific adc channels.
An argument of "all" will result in either a copy of the frame if the return format is specified as "frame", or a shred text
dump of the frame if the return format is specified as "shred".
```




On-Line Tcl Source Code

LDAS

The frame.tcl Library Module

Modification Date: 99.10.15

Table of Procedures

frame::init	frame::container::test
frame::mapToMsg	frame::current
frame::ptr2test	frame::find
frame::testData	frame::cache::init
frame::file2ptr	frame::cache::date
frame::ptr2file	frame::history
frame::chanlist	frame::names
frame::copy	frame::next
frame::create	frame::dump::edit
frame::intra::I,W,D	frame::shutdown
frame::change	

frame.tcl Version 1.0 Wraps the *frameAPI.so* and the *genericAPI.so* and the *genericAPI.tcl* for use by the frame API.
frame::cache is a list of the last 'n' records of frame file names with absolute paths to the files.

```
package provide frame 1.0
package requires frameAPI
namespace eval frame {
  set errlvl 1
  set occlude {}
  set cachelen 0
  set builderbroken 0
  set trigger {}
}
```

Name: *frame::chanlist*

Description:
Returns a tcl list of the channel names in the frame.
Will work on a frame file, if given the filename, or a frame_p to data from *readFrame*.

Parameters:

- target - the name of a frame, or a frame data pointer.

Usage:

```
set chanlist [ frame::chanlist $target ]
```

Comments:
A frame data pointer looks like _XXXXX_Frame_p.
The regular expression must be "inline" for some reason.

```
proc frame::chanlist { | target ** } {
  switch -regexp -- $target {
    (^|) ; ## a null string
      return {}
    }
    {^[_0-9a-z]+_Frame_p$} ; ## a frame pointer
      if { ! catch {
          set chanlist [ getChannelList $target ]
        } err } {
        return -code error "frame::chanlist: $err"
      }
      default ; ## a file name, or garbage
        if { ! [ file exists $target ] } {
          set msg "frame::chanlist: File not found |"$target|"**"
          return -code error $msg
        }
        if { ! catch {
          set target [ frame::file2ptr $target ]
          set chanlist [ getChannelList $target ]
          destructFrame $target
        } err } {
          return -code error "frame::chanlist: $err"
        }
      }
    }
  return $chanlist
}
```



On-Line C++ Source Code

http://pscc.ligo.caltech.edu/ldas/ldas-0/Doc/gwcode/PT/ptElem/ptElem.html - Microsoft Internet Explorer

Name: `sendElementObject` Send an Element as binary

Description:
This command sends an Internet LDAS Light-Weight Data set, called an Element because of its relationship to XML elements, through a Data Socket as a C++ Object. This method of sending Elements is used often because of efficiency.

Usage:
`sendElementObject ptSock ptElem`

Parameters:

Parameter	Description
<code>ptSock</code>	a pointer to a Data Socket which has previously been opened with the <code>createDataSocket</code> command
<code>ptElem</code>	a pointer to an Element object that has previously been instantiated in the C++ layer

Return value:
none

Exceptions:

Exception	Description
<code>invalid_socket</code>	the socket doesn't exist
<code>unconnected_socket</code>	the socket isn't connected
<code>invalid_element</code>	the element doesn't exist

[6 6 6](#)

Name: `sendElementObject_t` Send element as binary - Threaded

Description:
This command runs the `sendElementObject` command in a thread.

Usage:
`std::t1 [sendElementObject_t] ptSock ptElem`

Parameters:

Parameter	Description
<code>ptSock</code>	a pointer to a Data Socket which has previously been opened with the <code>createDataSocket</code> command
<code>ptElem</code>	a pointer to an Element object that has previously been instantiated in the C++ layer

Return value:
`t1d` - A pointer to the thread which was started.

[6 6 6](#)

Description of Comment - Microsoft Internet Explorer

```
class Comment : public Base {
public:
    XML Comment;
    This class represents an XML comment.

    Members

    Public
    Comment ( const Comment& c ) throws ( bad_alloc )           : Copy constructor.
    Comment ( const Comment& c ) throws ( bad_alloc )           : Copy constructor.
    Comment ( Reader& r ) throws ( bad_alloc, StreamException, FormatException ) : Input constructor.
    virtual ~Comment () throws ()                               : Destructor.
    const Comment& operator= ( const Comment& c ) throws ( bad_alloc ) : Assignment operator.
    bool operator== ( const Comment& c ) const throws ()       : Equal comparison operator.
    bool operator!= ( const Comment& c ) const throws ()       : Inequality operator.
    const string& getComment () const throws ()                 : Returns the comment.
    ClassType getClassid () const throws ()                    : Sets the comment.
    void setComment ( const string& c ) throws ( bad_alloc, FormatException ) : Writes a comment.
    virtual void write ( ostream& stream ) const throws ( exception ) : Input constructor.
    virtual void read ( Reader& r ) throws ( StreamException, FormatException ) : Input constructor.

    Friends
    friend void os_read ( os_bstream&, Comment& );

    Comment ( const Comment& c ) throws ( bad_alloc )

    Copy constructor.

    Parameters:
    Parameter      Description
    const Comment& c  The object to copy from.
```



API Log Files

```
The manager API Log File
614566651 openLog /usr1/pehrens/garbage/log/LDASmanager.log.html (file5)
614566652 openListenSock port 10001 (operator) opened on mainin as sock6
614566652 openListenSock port 10002 (emergency) opened on mainin as sock7
614566652 kgLoop looping process watchlogs started
614566652 kgLoop looping process framing started
614566652 kgLoop P1001 manager (614566652) -> frame (614566652): 119809 as
614566652 kgLoop looping process leak started
614566652 leakLogger initial size of manager API: 4008 Kb
614566652 leakLogger total (4008 Kb) heap 1416K stack 72K libmanager () generic ()
lived () libcol 544K
614566654 openLog /usr1/pehrens/garbage/log/LDASdebug0.log.html (file11)
614566654 createAssistant Namespace "debug0" created
614566655 debug0:getsid opened channel sock12 to mainin port 10003
614566655 debug0:sockhandler Your results are available as files: C1-624566594.R
at: http://ligo.caltech.edu/~pehrens/ldas_outfiles/debug0/
614566655 debug0:seq Stack "debug0:cmd" exhausted.
614566655 debug0:delete closed channel sock12
614566657 closeLog /usr1/pehrens/garbage/log/LDASdebug0.log.html (file11) closed
614566658 debug0:delete assistant manager "debug0" destroyed.
614566696 openLog /usr1/pehrens/garbage/log/LDASdebug1.log.html (file11)
614566696 createAssistant Namespace "debug1" created
614566696 debug1:getsid opened channel sock12 to mainin port 10003
614566699 debug1:sockhandler Your results are available as files: C1-624566596.R
at: http://ligo.caltech.edu/~pehrens/ldas_outfiles/debug1/
614566699 debug1:seq Stack "debug1:cmd" exhausted.
614566699 debug1:delete closed channel sock12
614566699 closeLog /usr1/pehrens/garbage/log/LDASdebug1.log.html (file11) closed
614566699 debug1:delete assistant manager "debug1" destroyed.
614566726 openLog /usr1/pehrens/garbage/log/LDASdebug2.log.html (file11)
614566726 createAssistant Namespace "debug2" created
614566727 debug2:getsid opened channel sock12 to mainin port 10003
614566729 debug2:sockhandler Your results are available as files: C1-624566726.R
at: http://ligo.caltech.edu/~pehrens/ldas_outfiles/debug2/
614566729 debug2:seq Stack "debug2:cmd" exhausted.
614566729 debug2:delete closed channel sock12
614566729 closeLog /usr1/pehrens/garbage/log/LDASdebug2.log.html (file11) closed
614566730 debug2:delete assistant manager "debug2" destroyed.
614566744 openLog /usr1/pehrens/garbage/log/LDASdebug3.log.html (file11)
614566744 createAssistant Namespace "debug3" created
614566744 debug3:getsid opened channel sock12 to mainin port 10003
614566747 debug3:sockhandler Your results are available as files: C1-624566743.R
at: http://ligo.caltech.edu/~pehrens/ldas_outfiles/debug3/
614566747 debug3:seq Stack "debug3:cmd" exhausted.
```

```
The frame API Log File
614566614 openLog /usr1/pehrens/garbage/log/LDASframe.log.html (file5)
614566639 frame::cache::init pushed 10001 files into frame::cache last frame created
at: 614566631 time now is: 614566639
614566639 openListenSock port 10003 (operator) opened on mainin as sock6
614566639 openListenSock port 10004 (emergency) opened on mainin as sock7
614566639 kgLoop looping process leak started
614566640 leakLogger initial size of frame API: 20024 Kb total (20024 Kb) heap
6208K stack 64K libframe 2432K generic 480K lived 1088K libcol 544K
614566640 kgLoop looping process horiframe started
614566640 kgLoop looping process frame_diag started
614566640 frame::diagnose frame::cache has 10001 elements, element
(1): /home/pehrens/frames/Data5/C1-614566639.F element
(10001): /home/pehrens/frames/Data5/C1-614566639.F delta_t: 10000
614566651 emergency executed: "9(=API)::reply fcid [ pongAPI ]"
614566655 metaOpts metaOpts debug0 : -dataquery all -jobid debug0 -returnformat
lived -interferometers C1 -times 614566654 -returnprotocol file/...
614566655 frame::find frame::find: /home/pehrens/frames/Data5/C1-624566594.F
614566657 frame::container::dump Dumped 293368 LdasContainer_p as ILSD
to /home/pehrens/public_html/ldas_outfiles/debug0/C1-624566594.F
614566657 leakLogger LEAKED 3536 Kb in 28 seconds (net) total (23592 Kb) heap
9744K stack 144K libframe 2432K generic 480K lived 1088K libcol 544K
614566696 metaOpts metaOpts debug1 : -dataquery all -jobid debug1 -returnformat
lived -interferometers C1 -times 614566696 -returnprotocol file/...
614566696 frame::find frame::find: /home/pehrens/frames/Data5/C1-624566596.F
614566699 frame::container::dump Dumped 208208 LdasContainer_p as ILSD
to /home/pehrens/public_html/ldas_outfiles/debug1/C1-624566596.F
614566699 leakLogger LEAKED 3568 Kb in 40 seconds (net) total (23592 Kb) heap
9768K stack 144K libframe 2432K generic 480K lived 1088K libcol 544K
614566727 metaOpts metaOpts debug2 : -dataquery all -jobid debug2 -returnformat
lived -interferometers C1 -times 614566726 -returnprotocol file/...
614566727 frame::find frame::find: /home/pehrens/frames/Data5/C1-624566726.F
614566729 frame::container::dump Dumped 258608 LdasContainer_p as ILSD
to /home/pehrens/public_html/ldas_outfiles/debug2/C1-624566726.F
614566729 leakLogger LEAKED 3568 Kb in 30 seconds (net) total (23592 Kb) heap
9768K stack 144K libframe 2432K generic 480K lived 1088K libcol 544K
614566744 metaOpts metaOpts debug3 : -dataquery all -jobid debug3 -returnformat
lived -interferometers C1 -times 614566741 -returnprotocol file/...
614566744 frame::find frame::find: /home/pehrens/frames/Data5/C1-624566743.F
614566747 frame::container::dump Dumped 208498 LdasContainer_p as ILSD
to /home/pehrens/public_html/ldas_outfiles/debug3/C1-624566743.F
614566747 leakLogger LEAKED 3568 Kb in 108 seconds (net) total (23592 Kb) heap
9788K stack 144K libframe 2432K generic 480K lived 1088K libcol 544K
614566759 metaOpts metaOpts debug4 : -dataquery all -jobid debug4 -returnformat
lived -interferometers C1 -times 614566759 -returnprotocol file/...
```




Problem Tracking System

Query Results - gnatweb - Microsoft Internet Explorer

Address: http://ligo.caltech.edu/gnat/gnatweb.cgi?table=4db

gnatweb

Query Results

15 matches found

PR	Originator	Arrival-Date	Closed-Date	Synopsis
134 edit	Peter Shawhan	2000-03-10 14:19:01		hwAPI should make use of masks to indicate null values
133 edit	Peter Shawhan	2000-03-10 14:19:00		metadatabaseAPI should make use of masks to indicate null values
132 edit	Peter Shawhan	2000-03-10 13:19:01	2000-03-10 13:26:58	hwAPI should handle empty lwd container with size=0
131 edit	Kent Blackburn	2000-03-10 12:49:01		Add support for main type in database tables and L/W/D external class
130 edit	mbames	2000-03-09 17:04:00		getElementAttribute is not implemented for all attributes in some elements
129 edit	Stuart Anderson	2000-03-08 17:49:00	2000-03-10 14:31:07	m27 fixing to mount /home/stuarterson
128 edit	Phil Ehrens	2000-03-08 16:49:00		attempt to run "sudo reboot" on maxim produces error message: Can't read /etc/passwd: Bad file number "age"
127 edit	Phil Ehrens	2000-03-08 16:34:01	2000-03-08 16:50:11	attempting to ssh anywhere as install causes terminal to hang and generates huge network load
126 edit	Phil Ehrens	2000-03-08 15:34:00	2000-03-08 16:10:25	password for user "install" on m28 is still "l_soff"
125 edit	mleu@ligo.caltech.edu	2000-03-08 15:04:00		m28-m71 latest data socket transfer very slow for dms 1000000
124 edit	Peter Shawhan	2000-03-08 14:34:01		getMetaData can be used to MODIFY the database
123 edit	Peter Shawhan	2000-03-08 14:04:00	2000-03-09 09:54:20	Error converting lwd containing an empty table to LIGO_LW
122 edit	mleu@ligo.caltech.edu	2000-03-07 16:49:00	2000-03-10 14:29:44	unable to mount /home/mleu directory on m71
121 edit	Stuart Anderson	2000-03-06 14:49:00		Change lds lgo-ha/etc/auto_imports
120 edit	Kent Blackburn	2000-03-06 10:34:00		Redhat versions and patch level inconsistent

Remember this query as:

