How to develop LAL Code

presented by John T. Whelan
The University of Texas at Brownsville
`whelan@oates.utb.edu`

Presented at LDAS Camp
2001 June 7

G010229-00-Z

Outline

1. Structure of a LAL Package
2. LAL Basics
3. Anatomy of a LAL Module
4. Constructing A New Routine & Package

LAL Resources

- LAL Spec: Theory
- LAL Software Documentation: Practice
- LAL Homepage `http://www.lsc-group.phys.uwm.edu/lal/` : Ground Truth

Structure of a Package

Consider a package `mypackage` containing a module `MyModule.c` including a function `LALMyFunction()`

- Header File `packages/mypackage/include/MyModule.h`
- Module `packages/mypackage/src/MyModule.c`
- Test Program `packages/mypackage/test/MyFunctionTest.c`
- Directory `packages/mypackage` and subdirectories `include`, `src`, `test`, `doc` each contain `Makefile.am`
- Auto-documentation generated by `packages/mypackage/doc/mypackage.tex`.

LAL Coding Different from C Coding

- LAL Functions (as opposed to those used internally) must conform to LAL spec
- `LALStatus` Structure & Macros
- LAL Data Types

Some Rules for LAL Functions

- Return void
- Up to 4 args:
  Status structure, [Output], [Input], [Parameters]
- Pass everything but simple integer and floating point inputs and parameters by reference.
- If your function has e.g. multiple inputs, pass it a pointer to a structure containing them.
- Choose a name which will be unique!

LAL Status Structure `LALStatus status`

- Since all LAL functions return void, must return any error information via status structure.
- `status.statusCode` is 0 for normal exit, negative for general LAL problems, positive for function-specific errors
- `status.statusDescription` contains the error message
- other fields (see LSD sec 7.3.4) contain function name, filename, etc.
- `status.statusPtr` points to another status structure which is passed to routines called by this one.

LAL Data Types

- Primitive data types aliased to appropriate C types:
  `CHAR`, `INT2`, `INT4`, `INT8`, `UINT2`, `UINT4`, `UINT8`, `REAL4`, `REAL8`
- Complex structures `COMPLEX8`, `COMPLEX16`:
  `z.re` and `z.im` define a complex number
- Aggregate data types `INT2Vector`, `COMPLEX16Vector`, etc.
  Like dynamically-allocated arrays;
  `vec.length` tells how many elements `vec.data[]` has.
- Structured data types like `REAL4TimeSeries` and `COMPLEX8FrequencySeries` include information about spacing of samples in time or frequency, physical units, etc.

Example: module `StochasticCrossCorrelationStatistic.c`

```
setenv STOCHHOME $LALHOME/src/lal/packages/stochastic/
```

then examine

- `$STOCHHOME/doc/main.pdf`
- `$STOCHHOME/src/StochasticCrossCorrelationStatistic.c`
- `$STOCHHOME/include/StochasticCrossCorrelationStatistic.h`

Breakdown of Module

- Version block (with CVS `$Id$`)
- Documentation block
- Includes
- NRCSID tag (with CVS `$Id$`)
- Function Declaration
- Variable Declarations
- Status Macros for Start
- Checks on & Extraction of Inputs & Params
- Calculation
- Assembly of Output
- Status Macros for End

Breakdown of Header

- Version block (with CVS `$Id$`)
- Documentation block
- Includes
- NRCSID tag (with CVS `$Id$`)
- Error Table
- Structure Definitions
- Prototypes

Adding a New Package

In the afternoon you will add your own
using the tarball `ldascamptemplate.tar.gz`

Here we illustrate with the following:

- Package `ldascamp`
- Module `LDASCampMoment.c`
- LAL Function `LALLDASCampMoment()` to calculate nth moment of distribution

* `LAL` prefix required by spec to avoid namespace collisions w/other libraries

* `LDASCamp` prefix avoids namespace collisions w/other LAL packages

```
setenv MYPKGHOME $LALHOME/src/lal/packages/ldascamp
mkdir $MYPKGHOME
cd $MYPKGHOME
tar -xvzf ldascamptemplate.tar.gz
cd $MYPKGHOME/src
emacs Makefile.am # replace LDASCampTemplate with LDASCampMoment
                  # and ldascamptemplate with ldascamp
mv LDASCampTemplate.c LDASCampMoment.c
cd $MYPKGHOME/include
emacs Makefile.am # replace LDASCampTemplate with LDASCampMoment
mv LDASCampTemplate.h LDASCampMoment.h
cd $MYPKGHOME/test
emacs Makefile.am # replace LDASCampTemplate with LDASCampMoment
mv LDASCampTemplateTest.c LDASCampMomentTest.c
```

```
cd $MYPKGHOME/doc
emacs Makefile.am # replace ldascamptemplate with ldascamp
mv ldascamptemplate.tex ldascamp.tex
emacs $MYPKGHOME/src/LDASCampMoment.c # see below
emacs $MYPKGHOME/include/LDASCampMoment.h # see below
cd $LALHOME/src/lal
emacs 00boot          # add ldascamp to lal_pkg_list_base
./00boot
./configure --prefix=$LALHOME --enable-shared --enable-mpi \
--with-extra-cflags=-fexceptions \
--with-extra-cppflags=-I/ldcg/include
make
```

On first pass through header and source file, replace

- `LDASCAMPTEMPLATE` with `LDASCAMPMOMENT` (MODULE NAME)
- `LDASCampTemplate` with `LDASCampMoment` (Module Name)

LAL Mailing Lists

Anyone writing LAL code should be subscribed to the LAL-announce and LAL-discuss mailing lists. While LAL recompiles, go to the websites below and subscribe.

http://www.lsc-group.phys.uwm.edu/mailman/listinfo.cgi/lal-announce
http://www.lsc-group.phys.uwm.edu/mailman/listinfo.cgi/lal-discuss

(These are also linked from the LAL home page)

Customizing the Template Files for Our New Package

In each file of our package (the remaining ones are `$MYPKGHOME/test/LDASCampMomentTest.c` and `$MYPKGHOME/doc/ldascamp.c`) replace

- `LDASCAMPTEMPLATE` with `LDASCAMPMOMENT` (MODULE NAME)
- `LDASCampTemplate` with `LDASCampMoment` (Module Name)
- `ldascamptemplate` with `ldascamp` (package name)

At this stage,

```
grep -i template $MYPKGHOME/*/*.c $MYPKGHOME/*/*.h \
$MYPKGHOME/*/*.tex
```

should not return anything.

Customizing the Template Files for Our New Package (cont'd)

Finally we customize `$MYPKGHOME/src/LDASCampMoment.c`, `$MYPKGHOME/include/LDASCampMoment.h`, and `$MYPKGHOME/test/LDASCampMomentTest.c` for our algorithm and recompile with

```
pushd $LALHOME/src/lal/lib
make clean
cd ..
make
popd
```

The resulting package should look something like the contents of the tarball `ldascamp.tar.gz`

Caveats

- If you make a change to a source file, you need to do

  ```
  pushd $LALHOME/src/lal/lib
  make clean
  cd ..; make
  popd
  ```

  before compiling your test routine.
- The executable `$MYPKGHOME/test/LDASCampTemplateTest` is a shell script; if you want to run a debugger on the binary, use `$MYPKGHOME/test/.libs/lt-LDASCampTemplateTest`
- If `make dvi` fails mysteriously, try

  ```
  cd $MYPKGHOME/doc/.adoc
  pdflatex main.tex
  ```

  to find out what the TEX error was.

Caveats (cont'd)

- If `make check` in `$LALHOME/src/lal` fails on `LALVersionTest` in the `support` package, you probably configured with different arguments at some point. Do the following

  ```
  cd $LALHOME/src/; make distclean
  ./configure --prefix=$LALHOME --enable-shared --enable-mpi \
  --with-extra-cflags=-fexceptions
  make
  ```

- If all else fails, you can redo the compile from scratch with

  ```
  cd $LALHOME/src/; make cvs-clean
  ./00boot
  ./configure --prefix=$LALHOME --enable-shared --enable-mpi \
  --with-extra-cflags=-fexceptions
  make
  ```