

How to Submit an LDAS User Command

Peter Shawhan
LIGO / Caltech

LDAS/DMT Camp
June 5, 2001



Basic Communication Model

Open a socket connection to the LDAS ManagerAPI
(at a particular port on the LDAS gateway machine)

Send the LDAS user command over the socket

```
ldasJob {-user peter -password ***** -email  
shawhan_p@ligo.caltech.edu} {getMetaData ...}
```

ManagerAPI responds with job ID, then closes connection

When job is done (or there is an error), ManagerAPI sends
email to the specified address

You can simply use telnet!

```
telnet ldas.ligo-wa.caltech.edu 10001  
> ldasJob {-user peter -password ***** -email  
shawhan_p@ligo.caltech.edu} {getMetaData ...} <Return>
```

But this would be tedious...



Communication Options

This is discussed in a LIGOTOOLS FAQ:

Go to <http://www.ldas-sw.ligo.caltech.edu/ligotools>

Click on "Frequently Asked Questions"

Click on "How can I submit an LDAS user command?"

Normal "background" execution (notification by email)

- Interactive: `telnet`
- Command-line: Tcl or perl script
- Program interface: C library
`$LIGOTOOLS/lib/libdataflow.a`

"Foreground" execution (notification returned directly to client)

- Interactive: `guild`
- Command-line, specialized: `getMeta`, `getFrames`



Sample Tcl Script

```
#!/usr/bin/env tclsh

# Set the host and port of the LDAS managerAPI
set hostName "ldas.ligo-wa.caltech.edu"
set port 10001

# Set the user command
set ldasCmd "ldasJob\
{-name myname -password blah -email me@ligo.caltech.edu}\
{getMetaData -returnprotocol http:out.xml -returnformat LIGO_LW\
-sqlquery {select tablename from syscat.tables} }"

# Open a socket to the LDAS managerAPI
if { [catch {socket $hostName $port} sockId] } {
    puts "Unable to connect to LDAS manager"
    exit
}

# Send the LDAS command
puts $sockId $ldasCmd
flush $sockId

# Get the response from the managerAPI and print it out
set jobInfo [read $sockId]
puts $jobInfo

# Close the socket connection and exit
close $sockId
exit
```

Password security!



Sample perl Script

```
#!/usr/bin/env perl

# Set the host and port of the LDAS managerAPI
$hostName = "ldas.ligo-wa.caltech.edu";
$port = 10001;

# Set the user command (note that the "@" is escaped)
$ldasCmd = "ldasJob"
. " {-name myname -password blah -email me@ligo.caltech.edu}"
. " {getMetaData -returnprotocol http:out.xml -returnformat LIGO_LW}"
. " -sqlquery {select tablename from syscat.tables} }";

# Open a socket to the LDAS managerAPI
use IO::Socket;
$sockId = IO::Socket::INET->new( Proto    => "tcp",
                                PeerAddr => $hostName,
                                PeerPort => $port )
    or die "Unable to connect to LDAS manager, aborting";

# Send the LDAS command
print $sockId $ldasCmd;

# Get the response from the managerAPI and print it out
@jobInfo = <$sockId>;
print @jobInfo;

# Close the socket connection and exit
close $sockId;
exit;
```

Password security!



Sample C Program

```
#include "DfSocket.h"

/*=====*/
int main(void)
{
    char host[] = "ldas.ligo-wa.caltech.edu";
    int port = 10001;
    char command[] = "ldasJob {-name myname -password blah -email
me@ligo.caltech.edu} {getMetaData -returnprotocol http:out.xml -returnformat
LIGO_LW -sqlquery {select tabname from syscat.tables} }";
    char buf[65536];
    int status;

    status = DfSocketCommand( host, port, command, buf, sizeof(buf) );

    printf( "Status is %d\n", status );
    printf( "Output is:\n%s\n", buf );

    return status;
}
```

To compile this on Solaris, do:

```
gcc mycmd.c $LIGOTOOLS/lib/libdataflow.a -I$LIGOTOOLS/include -L/usr/lib -
lsocket -lnsl -o mycmd
```

To compile it on Linux, do:

```
gcc mycmd.c $LIGOTOOLS/lib/libdataflow.a -I$LIGOTOOLS/include -o mycmd
```



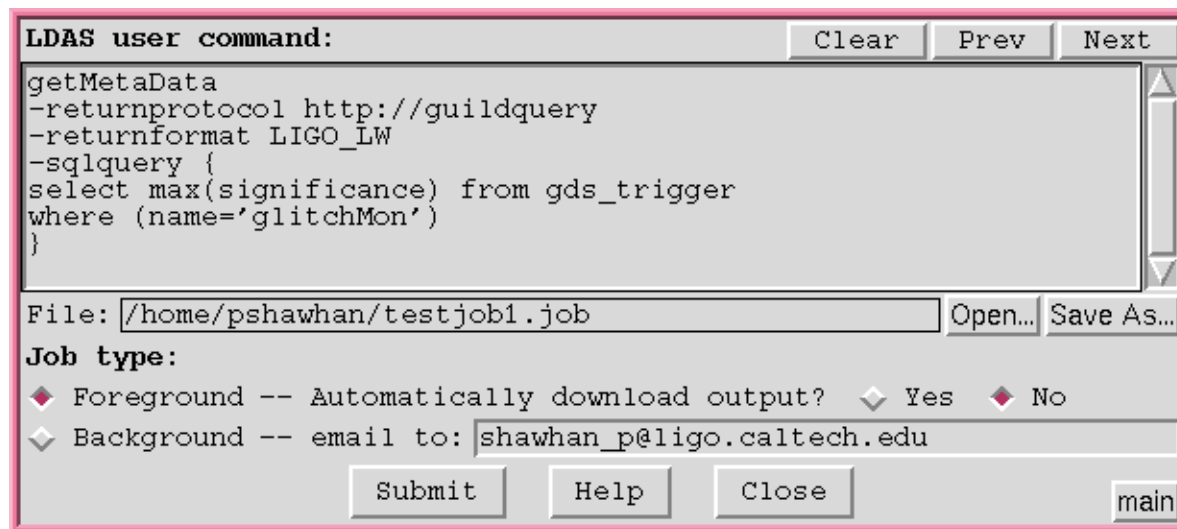
guild “LDAS User Command” Dialog

Allows “foreground” execution

Has a history mechanism to re-run previous commands

Can save / load from a file

Can include comments, newlines for readability





Future Directions

Command-line utility for foreground execution of an arbitrary LDAS user command: `ldasJob`

- e.g. for use in scripts

Secure socket communication

guild customization for other LDAS user commands?

Analysis “driver” tools, to submit a sequence of LDAS jobs

- Analysis-specific, or general-purpose?