

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

Technical Note	LIGO-T030286-00-C	12/07/03
ROBO SCIFOM		
Chethan Parameswariah		

This is an internal working note
of the LIGO Project.

LIGO Livingston Observatory
19100 Ligo Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352-0159
Phone (509) 372-8106
Fax (509) 372-8137

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 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

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

1. ABSTRACT

This document describes the working of the automatic capture and web logging system called “ROBO SCIFOM” for figure of merit displays and seismic noise, and logging them into the LLO elog at the end of each shift during science and engineering runs. Robo-scifom is part of the collection of software robots - “*SOFT-ROBOTS*” now working at LLO to ease and improve efficiency.

2. INTRODUCTION

Automation of LIGO CDS Systems is essential to maintain consistency and to minimize human errors, with the advancement of interferometer into low noise commissioning and science runs.

ROBO SCIFOM is a software robot that captures gif images of the two figures of merit displays and the seismic noise plot from the sun workstation monitors displaying them at LLO for previous 12 hours and then elogged to current day’s elog at the end of the shift during the science and engineering runs.

3. OPERATION

ROBO SCIFOM is part of “*SOFT-ROBOTS*” - a collection of software programs and scripts called ‘*software robots*’ that are run at LIGO to automate various duties of the scimons, operators and engineers to ease and improve efficiency while maintaining reliability and consistency on a daily basis.

The main program “autoelog_figofmerits.pl” is a wrapper for gif capture program and auto e-logger program. This is a Perl program that is spawned by a cronjob one minute past the end of shift on LLO2 a sun workstation at LLO which has the software license for ezcapy software. The ezcapy software supplied by Autograph International is used to capture screen shots of displays running on the sun workstations. Command line mode is used to control and capture the gif images. The “autoelog_figofmerits.pl” runs every day at one minute past 00:00 hrs, 08:00 hrs and 16:00 hrs local time (for science run S3) and spawns first, the shell script “autocapture_figmerits.sh” that captures the screen shots of figure of merit displays running on sun workstations control21 and control22. The shell script first clears the directory “/cvs/cds/llo/web/screencapture/figure_of_merit” of all files and then saves the captured images in gif format in the “/cvs/cds/llo/web/screencapture/figure_of_merit” path. Next the “autoelog_figofmerits.pl” script calls the auto-elogger program “figmerits_elog.pl” that elogs each of these captured images with the right title and annotation of the end of the shift. Each time the gif images are captured for the previous 12 hours and e-logged on the today’s page of the e-log. Currently

this cronjob is run on LLO2 in the mass storage room at LLO and the program requires this sun workstation to be 'ON' for it to work.

The program listing for `autoelog_figofmerits.pl`, `autocapture_figmerits.sh` and `figmerits_elog.pl` is in Appendix.

The perl program `autocapture_figmerits.sh` has a set of variables that define the sun workstation display to be captured, hard coded into the program such as the ip address, display number, location to save the captured gif file.

The `figmerits_elog.pl` is an automation script that takes these saved gif images and includes them in the automated elog entry. The information about the Day, Evening or Owl shift is also annotate along with the elog entry. The intelligence to identify between the Day, evening and owl shift is also hard coded by if/else loop and needs to be changed if there is a change in the time of the end of shift.

The `figmerits_elog.pl` perl program also uses various perl modules that help to post forms with authentication to web pages. The location of these modules is defined in the program to append it to the `@INC` variable.

Figure 1 shows the posting of the figure of merits and the seismic noise on the LLO elog during science run S3.

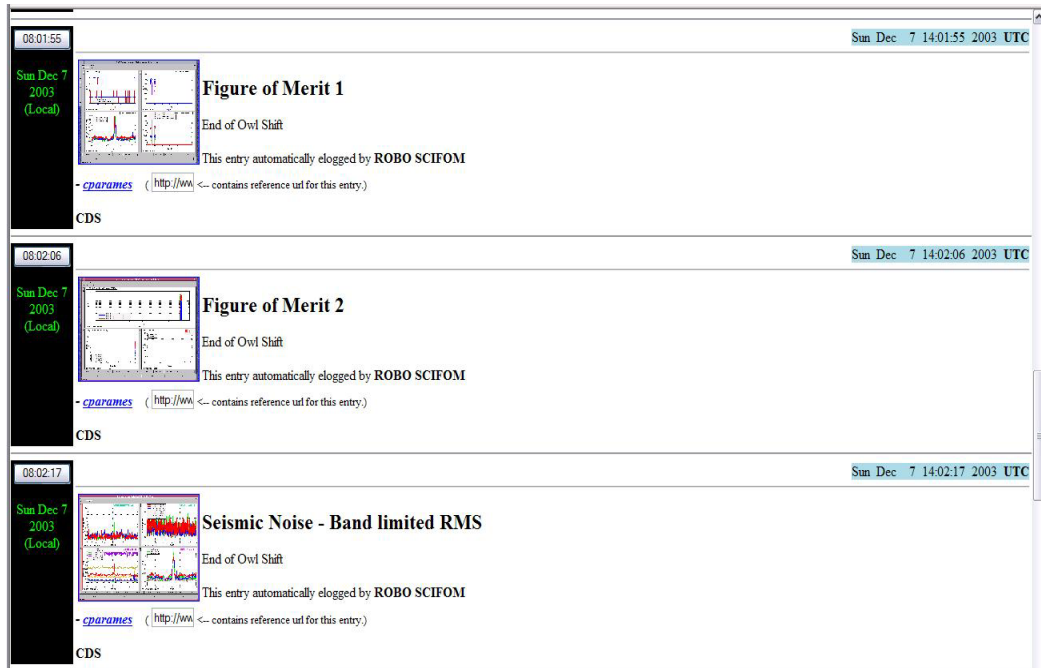


Figure 1: Figure of merit and seismic noise entry automatically posted elogged by ROBO SCIFOM.

Clicking on the thumbnail pictures a link opens up the full-page picture. Figure 2 shows figure of merit 1 plot for a day during S3. Figure 3 shows figure of merit 2 and figure 3 shows the seismic noise - band limited rms plots.

For the description of each of the plots in the figure of merit screens or the seismic band limited rms screen check the DMT monitor information.

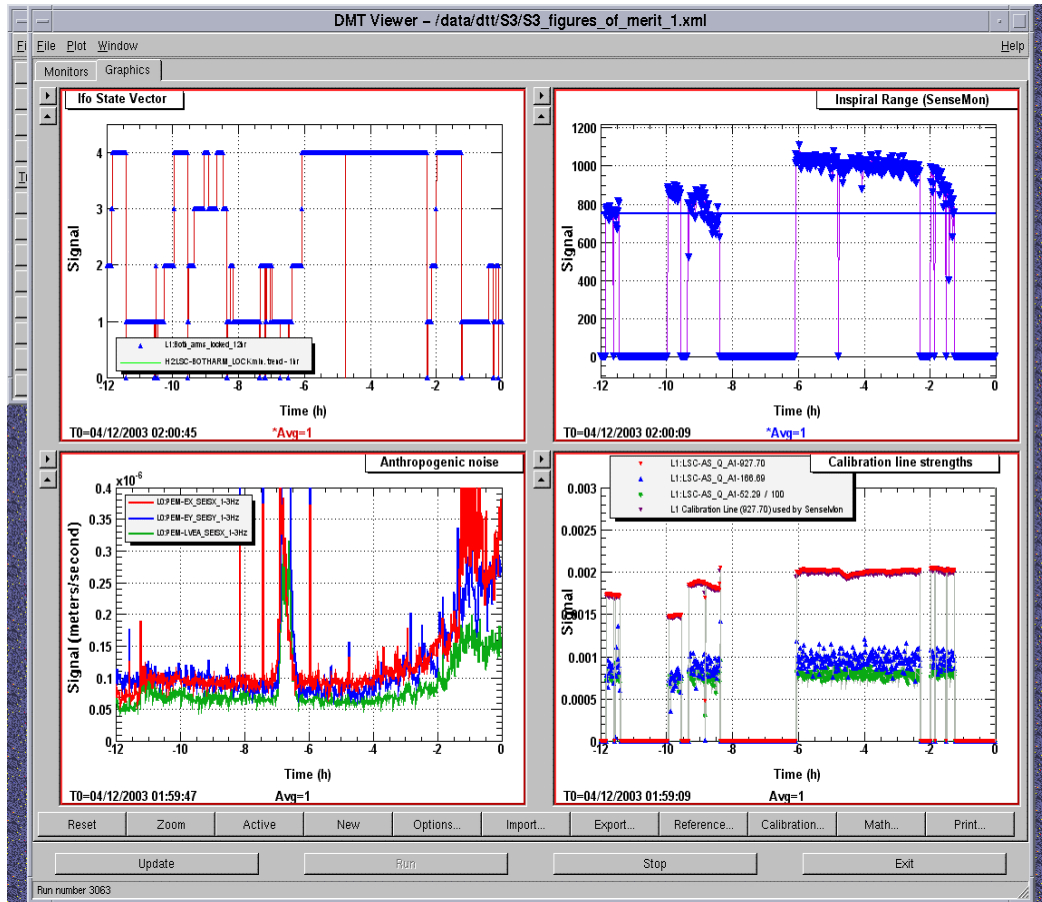


Figure 2: Figure of Merit 1

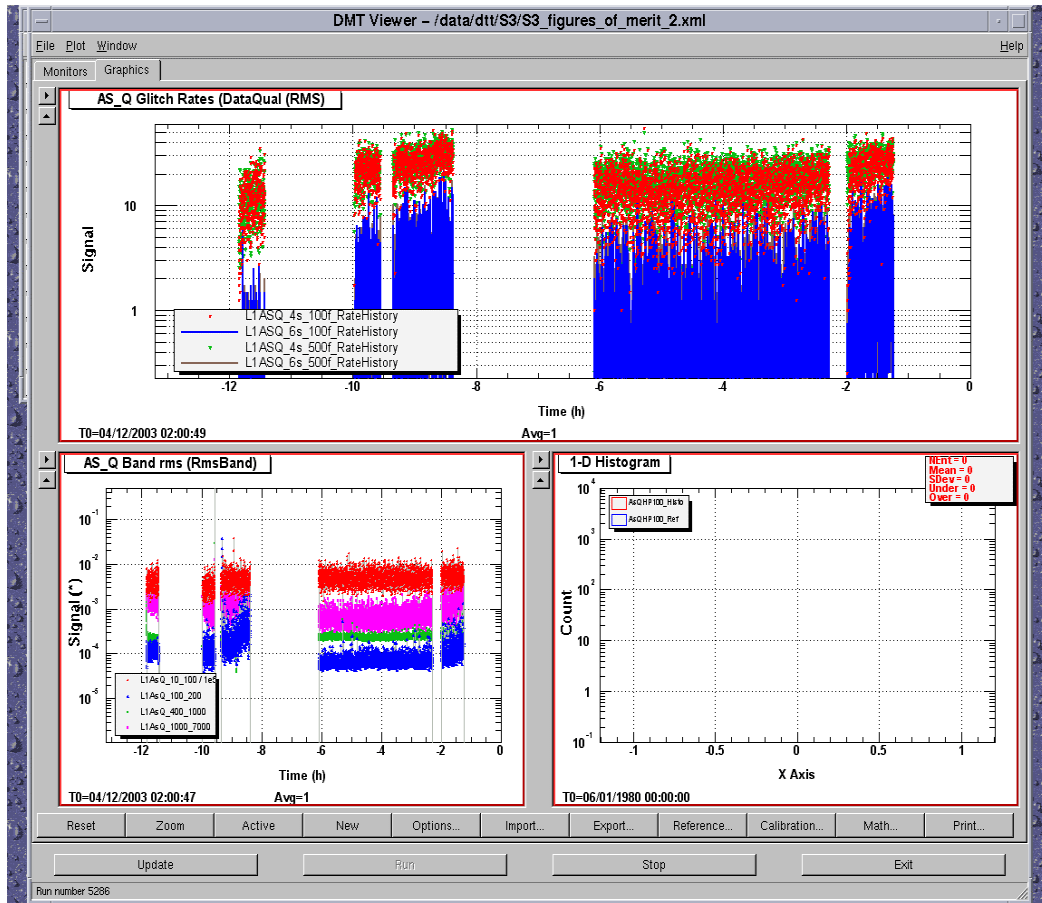


Figure 3: Figure of Merit 2

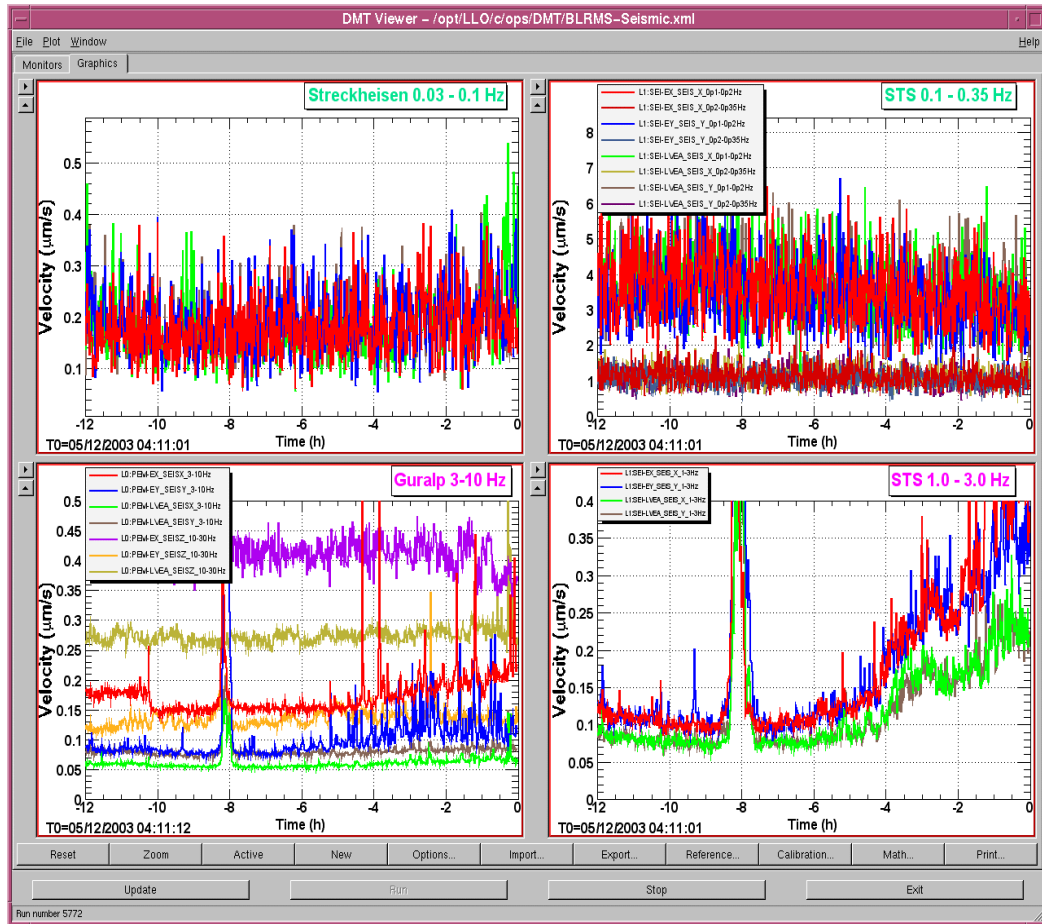


Figure 4: Seismic Noise – Band limited RMS

4. CONCLUSION

The ROBO SCIFOM was initiated for the science run S3 and has been working since. Features such as operator start and stop, best AS_Q plot during a shift, and non-updated figure of merit/seismic noise plot identification is currently either non-existent or is hard-coded in the program. The next version of the Robo-plotter will automate this adding intelligence to the program.

APPENDIX

Note: All the programs and configuration file are located on LLO CDS machines in the directory - */cvs/cds/project/robofom/*. The saved gif files are located in the directory - */cvs/cds/llo/web/screencapture/fig_of_merit/*.

- 1) Perl script run as cronjob at one minute past end of shift on LLO2 - *autoelog_figofmerits.pl*

```
#!/opt/apps/perl_5.6.1/bin/perl
#
# ROBO-SCIFOM - a software robot to automatically capture the
#figure of merit screens
# and then to elog them thrice a day - at LLO - 1 min past
#6.00hrs, 14.00hrs and 22.00hrs.
# For S3 the times were - 1 min past 08:00 hrs, 16:00 hrs and
#00:00 hrs
#
# This program is run at LLO on the machine that has the ezcopy
software - LLO2
#
# First release - Mar 27 2003 - Chethan Parameswariah version 1.0
# Second release - Oct 17 2003 - moved the files to project area
#and fixed the paths
#
# Chethan Parameswariah version 2.0
#
use EZCA
EZCA::AutoErrorMessageOff();
EZCA::SetTimeout(0.01);
EZCA::SetRetryCount(10);
require "ctime.pl";

# Find out where we are
$site = "undefined";
  whereAreWe();

if ($site eq "lho") {
    $dir_path = "/cvs/cds/lho/web/screencapture/fig_of_merit";
} elsif ($site eq "llo") {
    $dir_path = "/cvs/cds/llo/web/screencapture/fig_of_merit";
}
$DEBUG = 0;
$lasttime = 0;
@runtime_array = 0;
$lasttime_checked = 0;

@gps_time = EZCA::Get("L0:DAQ-SC_seconds","ezcaDouble",1);
if ($DEBUG) {
    print " GPS Time = $gps_time[1] \n";
}
#
# remove all files directory
#
$command = "/bin/rm -rf $dir_path/*";
```

```

        if (!$DEBUG) {
            print "Doing $command \n";
            system ($command);
        }
        else {
            print "$command \n";
        }
#
# Create new files
#
        $command =
"/cvs/cds/project/robofom/autocapture_figmerits.sh";
        if (!$DEBUG) {
            print "Doing $command \n";
            system ($command);
        }
        else {
            print "$command \n";
        }
#
# Make the elog entry
#
        $command = "/cvs/cds/project/robofom/figmerits_elog.pl";
        if (!$DEBUG) {
            print "Doing $command \n";
            system ($command);
        }
        else {
            print "$command \n";
        }
# -----
# -----
#
# Subroutine whereAreWe()
#
# find out what site we are on
#
# -----
# -----
sub whereAreWe {

    $temp = "./tempfile";

    # Find out which site we are on, using IP address of this
workstation
    $command = "/bin/rm -rf $temp";
    system($command);

    $command = "/usr/sbin/ifconfig -a > $temp";
    system($command);

    open(TEMP,$temp) || die "Cannot open file $temp\n";
    $site = "undefined";
    while ($line = <TEMP>) {
        if ($line =~ /10\.1\./) {
            $site = "lho";
        }
    }
}

```



```

    } elsif ($line =~ /10\.100\.\/) {
        $site = "llo";
    }
}
if ($site eq "undefined") {
    die "Cannot Determine Which LIGO Observatory this is\n";
}
print "Site is $site\n";
}

```

2) Gif capture script - autocapture_figmerits.sh

```

#!/bin/tcsh
#
# First release - Mar 27 2003 - Chethan Parameswariah. This
# program must be run only on LL02. version 1.0
# Second release - Dec 04 2003 - Chethan Parameswariah Added the
# seismic plot to be
#
#           captured version 2.0
#
/bin/eccapt -fullscreen -display 10.100.0.221:0.0 -capturemode \
xwd /tmp/seismicnoise.xwd
/bin/eccapt -fullscreen -display 10.100.0.221:0.1 -capturemode \
xwd /tmp/figmerit1.xwd
/bin/eccapt -fullscreen -display 10.100.0.222:0.0 -capturemode \
xwd /tmp/figmerit2.xwd
/bin/eccapt -fullscreen -display 10.100.0.222:0.1 -capturemode \
xwd /tmp/figmerit3.xwd
/bin/ecconv -export gif /tmp/seismicnoise.xwd \
/cvs/cds/llo/web/screencapture/fig_of_merit/seismicnoise.gif
/bin/ecconv -export gif /tmp/figmerit1.xwd \
/cvs/cds/llo/web/screencapture/fig_of_merit/figmerit1.gif
/bin/ecconv -export gif /tmp/figmerit2.xwd \
/cvs/cds/llo/web/screencapture/fig_of_merit/figmerit2.gif
/bin/ecconv -export gif /tmp/figmerit3.xwd \
/cvs/cds/llo/web/screencapture/fig_of_merit/figmerit3.gif

```

3) Auto elogger - figmerits_elog.pl

```

#!/opt/apps/perl_5.6.1/bin/perl
#
# First release - Oct 17 2003 - Chethan Parameswariah - Auto
# elogger for the FOM and seismic display version 1.0
# Second release - Dec 04 2003 - Chethan Parameswariah - Added
# the seismic plots to be elogged version 2.0
#
#
# Need these lib modules - this prepends to @INC at run time
#
use lib "/opt/apps/perl_5.6.1/modules/HTML-Parser-2.22/blib/lib";
use lib "/opt/apps/perl_5.6.1/modules/libwww-perl-5.42/lib";
use lib "/opt/apps/perl_5.6.1/modules/URI-1.02";

```

```

use lib "/opt/apps/perl_5.6.1/modules/HTML-Parser-2.22/lib";
use lib "/opt/apps/perl_5.6.1/modules/MIME-Base64-2.11/blib/lib";

#use lib "/cvs/cds/project/robofom/modules/HTML-Parser-
2.22/blib/lib";
#use lib "/cvs/cds/project/robofom/modules/libwww-perl-5.42/lib";
#use lib "/cvs/cds/project/robofom/modules/URI-1.02";
#use lib "/cvs/cds/project/robofom/modules/HTML-Parser-2.22/lib";
#use lib "/cvs/cds/project/robofom/modules/MIME-Base64-
2.11/blib/lib";

# Tell what modules to use
#
use HTTP::Request::Common qw(POST);
use LWP::UserAgent;
use CGI;

# Set variables
#
$DEBUG = 1;
$dir_path = "/cvs/cds/llo/web/screenshot/fig_of_merit";
$seismicnoise = "$dir_path/seismicnoise.gif";
$figureofmerit1 = "$dir_path/figmerit1.gif";
$figureofmerit2 = "$dir_path/figmerit2.gif";
# Commented out since the AS_Q spectrum may not be good at the
time of capture.
#$figureofmerit3 = "$dir_path/$gps_time/figmerit3.gif";
$username = "cparams";
$password = "wave\$";

# Get hour, today's date, month and year
($HOUR_NUMBER, $DAY_NUMBER, $MONTH_NUMBER, $YEAR_NUMBER) =
(localtime(time)) [2,3,4,5];
$MONTH_STRING = (qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov
Dec))[(localtime) [4]];
$MONTH_NUMBER += 1;
if ($MONTH_NUMBER < 10) {
    $MONTH_NUMBER = "0".$MONTH_NUMBER;
}
if ($DAY_NUMBER < 10) {
    $DAY_NUMBER = "0".$DAY_NUMBER;
}
$YEAR_NUMBER += 1900;

if ($DEBUG) {
print "Hour = $HOUR_NUMBER\tMonth = $MONTH_STRING -
$MONTH_NUMBER\tDay = $DAY_NUMBER,\tYear = $YEAR_NUMBER\n";
}

#use this if you want to test the script for a day when reboot
was made, change the value.
#$DAY_NUMBER = 3;

##### Auto Elog #####

if ($HOUR_NUMBER eq "8") {
    $comment_string = "End of Owl Shift";
}

```

```

}
elsif ($HOURL_NUMBER eq "16") {
    $comment_string = "End of Day Shift";
}
elsif ($HOURL_NUMBER eq "0") {
    $comment_string = "End of Evening Shift";
}

# Create a new user agent
#
$ua = LWP::UserAgent->new();

# Since I need a proxy at LLO, set the proxy here
#
$ua->proxy('http','http://london.ligo-la.caltech.edu:80/');

# set url of elog
#
my $URL = 'http://www.ligo-la.caltech.edu/ilog/pub/ilog.cgi?';

# Elog file date
#
$log_file_date = "$MONTH_NUMBER/$DAY_NUMBER/$YEAR_NUMBER";
#$log_file_date = "12/23/2001";

$dispnum = 1;
$comment_string_header = "<H2>Figure of Merit ".$dispnum."</H2>";
$comment_string_footer = "<P>This entry automatically elogged by
<b>ROBO SCIFOM </b><BR>";

$comments =
$comment_string_header."<P>".$comment_string.$comment_string_foot
er;
if ($DEBUG) {
    print $comments."<n";
}
# POST it with contents and values
#
my $request = POST $URL,
    Content_Type => 'multipart/form-data',
    Content =>
    [
        group => 'detector',
        task => 'makeEntry',
        log_file_date => $log_file_date,
        comments => $comments,
        keywords => 'CDS',
        priority => 'normal',
        'image_to_include' => [$figureofmerit1],
        entry_author => 'cparams',
        'submit' => 'Submit Log Entry'
    ];

# Dont forget the authorization
#

```

```

$request->authorization_basic($username, $password);

# And finally make the call.
$content = $ua->request($request)->as_string;

if ($DEBUG) {
print $content;
};

sleep 10;

$dispnum = 2;
$comment_string_header = "<H2>Figure of Merit ".$dispnum."</H2>";
$comments =
$comment_string_header."<P>".$comment_string.$comment_string_footer;
if ($DEBUG) {
    print $comments."\n";
}
# POST it with contents and values
#
my $request = POST $URL,
    Content_Type => 'multipart/form-data',
    Content =>
    [
        group => 'detector',
        task => 'makeEntry',
        log_file_date => $log_file_date,
        comments => $comments,
        keywords => 'CDS',
        priority => 'normal',
        'image_to_include' => [$figureofmerit2],
        entry_author => 'cparams',
        'submit' => 'Submit Log Entry'
    ];

# Dont forget the authorization
#
$request->authorization_basic($username, $password);

# And finally make the call.
$content = $ua->request($request)->as_string;

if ($DEBUG) {
print $content;
};

sleep 10;

$comment_string_header = "<H2>Seismic Noise - Band limited
RMS</H2>";
$comments =
$comment_string_header."<P>".$comment_string.$comment_string_footer;
if ($DEBUG) {
    print $comments."\n";
}

```

```

# POST it with contents and values
#
my $request = POST $URL,
    Content_Type => 'multipart/form-data',
    Content =>
    [
        group => 'detector',
        task => 'makeEntry',
        log_file_date => $log_file_date,
        comments => $comments,
        keywords => 'CDS',
        priority => 'normal',
        'image_to_include' => [$seismicnoise],
        entry_author => 'cparams',
        'submit' => 'Submit Log Entry'
    ];
#
# Dont forget the authorization
#
$request->authorization_basic($username, $password);

# And finally make the call.
$content = $ua->request($request)->as_string;

if ($DEBUG) {
print $content;
};

#sleep 10;

#$comment_string_header = "<H2>Noise Spectrum - LLO</H2>";
#$comments =
$comment_string_header."<P>".$comment_string.$comment_string_footer;
#if ($DEBUG) {
#    print $comments."\n";
#}
## POST it with contents and values
#
#my $request = POST $URL,
#    Content_Type => 'multipart/form-data',
#    Content =>
#    [
#        group => 'detector',
#        task => 'makeEntry',
#        log_file_date => $log_file_date,
#        comments => $comments,
#        keywords => 'CDS',
#        priority => 'normal',
#        'image_to_include' => [$figureofmerit3],
#        entry_author => 'cparams',
#        'submit' => 'Submit Log Entry'
#    ];
#
# Dont forget the authorization
#
#$request->authorization_basic($username, $password);

```

```
#  
# And finally make the call.  
#$content = $ua->request($request)->as_string;  
#  
#if ($DEBUG) {  
#print $content;  
##};
```

***** **END** *****