

Cmlog Viewer

The cmlog viewer is an interactive Eclipse plugin for viewing and querying [cmlog](#) messages. It can be run within SEAL, or as a stand-alone application.

What does it do?

The viewer allows you to browse control system messages, receiving updates from the live system as well as those previously logged. You can specify keyword filters to narrow messages to only those that interest you. It also supports printing, saving to a file, cutting and pasting, sorting, pausing and resuming.

How do I use it?

Starting

Within SEAL: Start the cmlog viewer by starting its Eclipse View. That is, Window -> Show View (you may need to select Other ... at this point) SEAL -> cmlog. Once started, use the File -> Get Messages dialog to connect to a cmlog server.

Stand-Alone viewer:

- Option 1: On the CA network, run `$LCLS_ROOT/bin/cmlogviewer` from the command-line.
- Option 2: On the CA network, from the lcls home screen, select "Message Viewer".
- Option 3: From a linux workstation with access to AFS, run `/afs/slac/g/lcls/physics/bin/cmlogviewer`

Getting Some Messages

Under the "File" menu, select "Get Messages...". Doing so will open a dialog box that allows you to specify the message server, filter messages by content, and indicate whether to listen for live messages, or only to search through old messages.

Server: Usually you will want to select "default", since this provides messages from the machine-specific system. You can choose another server if you have a specific need. You can also manually enter a server by specifying its network name and port separated by a colon (e.g. `opi00gtw00.slac.stanford.edu:8093`).

Filter: If you are interested in only certain messages, you can indicate this with keywords. Only messages that contain all of the whitespace-separated words you enter in this text box will be delivered. If you leave "No Filter" selected, all messages will be delivered. Note that you can also change the filter after a connection has been established. Those who wish to do more specialized filtering may use the CMLOG query syntax. To do so, enter ':' as the first character of the filter, and follow that with a valid CMLOG query. See examples in section [#Retrieved Message Filter Pattern Examples](#) below.

Live / Old Messages: If you choose to listen for live messages (the default), you must specify how much time's worth to keep. Do so by entering a number and a unit of time in the text box (e.g. `10 minutes`, or `3 hours`, or `2 days`, ...). When the connection is first established, the entire timespan will be filled in with previously logged messages. They will disappear from the viewer when they become too old.

If you choose to retrieve old messages only, then you specify a time range in the text box. Only messages logged within that time range will be retrieved by the viewer. You may select a pre-defined time range, or enter one yourself in the format of "begin - end". In addition to the standard `mm/dd/yy hh:mm:ss` format, you may also use several special phrases as shorthand. (e.g. `11/26/07 08:00 - 11/26/07 12:30`, or `current day - now`, or `current hour - current minute`, or `current day - current shift`). If instead of using a pre-defined date or typing one manually, you select "Assist..." from the drop down box, you'll get a little wizard that will help you construct a time range by specifying an anchor time and a duration.

Using the Viewer

Sorting: Normally, messages are sorted by the time field, with newer ones at the bottom. But you can sort by any of the fields. To do so, simply click on the desired column heading. Clicking the currently selected column will reverse the sort order. Note that the black triangle in the sort column header designates the direction: the vertex indicates the direction of descending values.

Pausing updates: Because the message view is continuously updating, it can be useful at times to freeze the display. You can do this by clicking the "Pause Updates" button in the bottom right corner of the window. When you pause updates, you prevent the window contents from changing, but things are still happening under the hood. When you un-click the button, the window contents will look the same as they would have, had you never paused in the first place.

Changing the message retrieval parameters: At the top of the message view are two text boxes. The one on the left indicates the current filter and, and the one on the right indicates the time duration or the time range, depending on whether you selected to view live updates or just browse through old ones. You can update these fields in the same manner as in the initial connection dialog box described above, in "Getting Some Messages". When you modify a text box, the background will turn yellow to indicate that the modifications are unapplied. You may either hit the enter key to apply the changes or the escape key to revert. Note that you may modify both the filter and time specification. Hitting the enter key in either box will apply the changes from both.

Copying messages to the clip board: You can copy messages to the system clipboard so that you can paste them into other applications such as text editors, spreadsheets, etc. To select one message, just click on it in the viewer and select "Copy" from the "Edit" menu. You can select all messages by using the corresponding menu entry. To select multiple messages, select one message then hold the shift key and click on another message. Doing this will cause all message in between to be selected in addition to the two endpoints. To select non-contiguous messages, hold the control key while clicking on individual messages.

Saving to a file: You can save the contents of the message viewer to a text file by selecting "Save As..." from the "File" menu.

Printing: Select "Print..." from the "File" menu.

Clearing messages: You can clear all messages from the view by selecting "Clear Messages" from the "Edit" menu. To delete specific messages, select them as though you were going to copy them, then choose "Delete" from the "Edit" menu.

Retrieved Message Filter Patterns

The Retrieved Messages filter dialog is in the top left hand corner of a message window. This dialog is normally set to "No Filter," meaning that all messages are retrieved and displayed by the viewer. However, you can specify that only those messages conforming to some filter pattern are retrieved. See the example in the screenshot below (which was to retrieve all messages except those whose text contains SIPMAIN).

Searching for Messages

There are two modes of searching (aka filtering). First is the simple search. In this case, just type the text you're interested in into the Filter dialog box. Second allows more complex searches, which use search expressions (see below).

Note that both kinds are **CASE SENSITIVE**

Simple Search

If you simply want to retrieve only those messages containing some text, then just put the text in the dialog box and hit return. For instance, if you want to retrieve only messages with QUAD in them.

```
QUAD
```

Retrieve only severity ERROR messages (note that you have not had to specify the Severity field, since these simple searches are on all fields).

```
ERROR
```

Expression Search

More complicated searches, such as those on specific fields other than the Text field, or those involving negation, must be preceded by ":".

Note - there is a bug here: in the expression search, the column names must be given all Lower Case (eg text), even though the actual column heading as printed in the GUI, is Capitalized First Letter (Text).

Retrieve only messages where the message Text includes 'failed'.

```
:text like 'failed'
```

Retrieve all the messages *except* those where the text contained some string - this uses the '!' (not) operator. Eg NOT containing DpSlc or physicsMatlab:

```
:!(facility like 'DpSlc' || facility like 'physicsMatlab')
```

Another even more restrictive filter cuts out talkative systems leaving many SLC messages:

```
:!(facility like 'DpSlcKlys' || facility like 'Alarm' || facility like 'physics') || facility like 'ChannelWatcher')
```

It's also useful to cut out messages whose text itself contains some pattern. Here's an example having cut out messages whose text contains "SIPMAIN".

Facility	Severity	Text	Time	Code	Host
SLC	EROR	TA00 TRS2:pwrold=110.55, pwr=110.66, diff= 0.34	00:00:03 2008-01-09	CAU-E-EPIC	RFS00
SLC	EROR	TA00 TRS2:abs(pwr-trgt)= 0.34, lims= 1.00, 40.00	00:00:04 2008-01-09	CAU-E-EPIC	RFS00
SLC	EROR	LI11FBCK: Calculated actuators exceed limits in CTL task.	00:00:04 2008-01-09	FBCK-E-ACT	LI11
LCLS		QUAD:LI27:601:STRG changed to , MINOR	00:00:07 2008-01-09		lcls-dat
LCLS		QUAD:LI27:401:STRG changed to , MINOR	00:00:07 2008-01-09		lcls-dat
LCLS		QUAD:LI27:801:STRG changed to , MINOR	00:00:07 2008-01-09		lcls-dat
SLC	EROR	LI22 VACG 8230 FILAMENT is OFF	00:00:17 2008-01-09	STAT-E-O\$D	V016
SLC	EROR	BIVSC - MC00(153) had 1 timeout(s) in the last 30 seconds!	00:00:19 2008-01-09	SLCNET-E-T	V159
LCLS		QUAD:LI28:501:STRG changed to , MAJOR	00:00:21 2008-01-09		lcls-dat
SLC	EROR	TW04 TR10:IOC:ALRM:HRTBT changed from 1 to 0	00:00:21 2008-01-09	CAU-E-EPIC	OPI000
SLC	EROR	EP02 PROF 813 PR-813 FOCUS is	00:00:23 2008-01-09	STAT-E-O\$D	V016
SLC	EROR	TA00 0x18bf2a8 (pp1): 14 of 15 assigned channels have connected	00:00:24 2008-01-09	CAU-E-EPIC	RFS00
SLC	INFO	SLCCAS_GETDIAG - Standalone Startup (SLCCAS_GETDIAG)	00:00:28 2008-01-09	OPS-I-STAR	V206
SLC	INFO	SLCCAS_GETDIAG - Image: \$255\$DUA2:[PRODUCTION.][IMAGE_ALPHA]SLCC,	00:00:28 2008-01-09	OPS-I-STAR	V206
SLC	INFO	SLCCAS_GETDIAG - Standalone Exit (SLCCAS_GETDIAG)	00:00:28 2008-01-09	OPS-I-STOP	V206
SLC	INFO	SLCCAS_GETDIAG - Standalone Startup (SLCCAS_GETDIAG)	00:00:28 2008-01-09	OPS-I-STAR	V206
SLC	INFO	SLCCAS_GETDIAG - Image: \$255\$DUA2:[PRODUCTION.][IMAGE_ALPHA]SLCC,	00:00:29 2008-01-09	OPS-I-STAR	V206
SLC	INFO	SLCCAS_GETDIAG - Standalone Exit (SLCCAS_GETDIAG)	00:00:29 2008-01-09	OPS-I-STOP	V206
SLC	EROR	EP02 PROF 813 PR-813 FOCUS is	00:00:29 2008-01-09	STAT-E-O\$D	V016

Connected to opi00gtw00-lavc.slac.stanford.edu:8093 44054 messages.

Writing Messages from Matlab to cmlogviewer

You can use the [Err package](#) to write from Matlab to cmlog.

Initialize Err with `aidainit`. Then just once, you must get an Err object using its `getInstance` method. Give the name of your program to `getInstance`, like `getInstance('Myprogram')` then all of your subsequent messages will be labeled in cmlog as coming from the facility 'Myprogram'. After that, you can issue messages to cmlog using the `log` method Err, or issue messages to cmlog and print them to the local screen using the `logl`.

Example:

```
>> aidainit;
Aida client initialization completed
>> e = Err.getInstance(Myprogram');
>> e.log('Logs this message in cmlog only');
>> e.logl('Logs this message in cmlog and to the screen');
Thu May 20 16:37:48 PDT 2010: Logs this message in cmlog and to the screen
```

Programmers' Notes

To build on production

Use `iclipse` - to set yourself up with Eclipse in your "own" production subdir (This procedure is the same as for SEAL in general, where for SEAL the product is `feature.seal.application` and double-click on `seal.product`. When building SEAL you also need the CSS stuff in `physics.css`).

- mount `lcls` cvs repo.
- Expand `physics/seal`. Select All in `physics/seal`, then check them out (not checkout `physics/seal` itself, because of funny subdir issue in Eclipse)
- Select `feature.cmlogviewer`; double-click on `cmlogviewer.product`.
- Go to Exporting -> Eclipse Product Export Wizard
- Override the Product Configuration "Root directory" to new version number, eg "2.2"
- Check that Destination is `/usr/local/lcls/physics/apps/cmlogviewer/product/`
- Build
- Check that the output was built in the destination

- Test it (without using the script therefore)
 - Optionally, set CMLOG_HOST and CMLOG_PORT
 - `$LCLS_ROOT/physics/apps/cmlogviewer/product/<version>/cmlogviewer -data $HOME/.cmlogviewer`
 - Ignore Gtk-WARNING
- Repoint the "current" link to the new production: Eg from `/usr/local/lcls/physics/apps/cmlogviewer/product/` execute:
`rm current && ln -s 2.2 current`
- check executes from `lclshome`

Known Issues

In Jira, Cmlog Viewer is a component in the SEAL Project.

Summary of known Jira Issues: <https://jira.slac.stanford.edu/secure/IssueNavigator.jspx?reset=true&mode=hide&sorter/order=ASC&sorter/field=priority&pid=10447&component=10850>