
cmlxlinp: Linux Performance Metric Probe

CML00120-01

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com
Copyright © 2014 by Code Magus Limited
All rights reserved

Contents

1	Introduction	2
1.1	Further Reading	2
2	Synopsis	3
3	Invocation	4
4	Command Interface	4
5	Command Elements	4
5.1	Comments	4
5.2	Reserved Words	4
5.3	Identifiers	5
5.4	Strings	5
5.5	Integers	5
5.6	Numbers	6
5.7	Environment Variables	6
6	Syntax and Semantics	7
6.1	Add Command	7
6.2	Close Command	8
6.3	Delete Command	9
6.4	Display Command	10
6.5	Exit Command	11
6.6	Help Command	11
6.7	Open Command	11
6.8	Shutdown Command	12
6.9	Set Command	12
6.10	Start Command	15
6.11	Stop Command	15
6.12	Switch Command	16
7	Installation	17
7.1	Overview	17
7.2	RPM Install Process	17
7.3	Package Install Process	18
A	Example Command File	19

1 Introduction

This document describes how to use `cmlxlinp` which is a software utility to fetch Linux performance data using the `/proc` process information pseudo-file system. The data is fed to a `Serfboard` server for use in displaying a real time dashboard and stored for post processing analysis.

In order to process raw counters from the Linux platform and send them to `Serfboard` in the correct form `cmlxlinp` needs to be configured. This is done through its command interface.

The `proc` file system is a pseudo-file system which is used as an interface to kernel data structures. It is commonly mounted at `/proc`. Most of it is read-only, but some files allow kernel variables to be changed. See `man proc` for more details.

`cmlxlinp` reads the values of a list of performance metrics to be fetched from the local Linux host. Each metric is then formatted as a `Serfboard` metric and sent to a `Serfboard` server.

1.1 Further Reading

For `Serfboard` documentation please refer to the following manuals:

- [Serfboard Configuration Guide and Reference Version 1 \[3\]](#)
- [Serfboard Instruments Guide and Reference Version 1 \[4\]](#)
- [Serfboard Installation Guide and Reference Version 1 \[5\]](#)
- [Serfboard User Guide Version 1 \[6\]](#)

2 Synopsis

cmlxlinp is invoked from the command line and if the ‘--help’ parameter is specified it will display all available parameters and their options if applicable. Below is the help display and following that is a description of each parameter.

```
Code Magus Limited Linux probe V1.0: build 2020-11-16-17.31.46
[../cmlxlinp] $Id: cmlxlinp_help.tex,v 1.1 2020/12/16 17:51:11 hayward Exp $
Copyright (c) 2009 by Code Magus Limited. All rights reserved.
[Contact: stephen@codemagus.com].
```

```
Usage: cmlxlinp [OPTION...]
  -p, --port={60054|<port>}      Command interface port
  -c, --command=<command>        Command to pass to command process
  -n, --name=<name>              Optional instance name
  -s, --cache-size=obsolete      Obsolete
  -v, --verbose                  Verbose output
  -t, --trace                    Trace message output
```

Help options:

```
  -?, --help                    Show this help message
      --usage                    Display brief usage message
```

Where:

- ‘-p|--port’ Specifies the command interface port for cmlxlinp, If not specified it will default to 60054.
- ‘-c|--command’ Specifies a command to passed to the command interface.
- ‘-v|--verbose’ When specified, cmlxlinp operates in a verbose manner.
- ‘-t|--trace’ When specified, cmlxlinp writes all activity to stdout.

3 Invocation

When `cmlxlinp` is invoked it starts a command interface through which the processing of the probe is configured and listens on a TCP/IP port (see invocation parameters in sub-section 2 on page 3) for connections. The command supplied as a parameter is presented directly to the command interface, followed by commands supplied through connections to the TCP/IP port, either interactively with a client like `telnet` or through `cmlcmd` [2].

Once configured the metrics and definitions may also be viewed or the `cmlxlinp` environment dynamically reconfigured via further commands.

This following sub-sections describe the `cmlxlinp` command interface and commands.

4 Command Interface

`cmlxlinp` is configured from commands presented to its command interface. Input is either a single command or the name of a command file. Commands are explained in detail in the following sections. A command file is a text based file that consists of one or more commands, where each command is on a separate line. Typically a probe requires multiple commands to be effectively configured so commands are often written as a logical group in a command file. A command file name is validated using the library `cmdname` [7].

5 Command Elements

5.1 Comments

Comments are introduced by using a double minus (“--”) and continue up to the end of the current input line.

Examples:

```
-- This is an example of a command comment.  
-- and is useful in documenting command files.
```

5.2 Reserved Words

Reserved words have a special meaning in terms of directing the parsing of commands. The reserved words are:

add	group	parameter	udp
all	help	parameters	trace
available	inactive	probe	verbose
close	interval	server	
counter	log	set	
counters	metric	shutdown	
delete	msglevel	start	
delay	notset	stop	
display	open	switch	
exit	polling	title	

5.3 Identifiers

Identifiers are case sensitive and start with a letter which can be followed by any number of letters, digits, decimal point '.' or the under-score character.

Examples:

```
cpu.cpu.system
```

5.4 Strings

Strings are:

- any sequence of characters (except double quotes and the newline character) enclosed by double quotes.
- any sequence of characters (except single quotes and the newline character) enclosed by single quotes.

Examples:

```
"Seconds spent in user mode"
"ABC Company's Metric File"
'$Revision: 1.2 $'
```

5.5 Integers

An integer consists of a nonempty sequence of decimal digits.

Examples:

```
1234
0
```

5.6 Numbers

A number consists of a nonempty sequence of decimal digits that

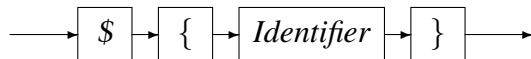
- possibly contains a radix character (decimal point ‘.’).
- is optionally followed by a decimal exponent; consisting of an ‘E’ or ‘e’ followed by an optional plus or minus sign followed by a nonempty sequence of decimal digits that indicates multiplication by a power of 10.

Examples:

1234
0.001
1.2
123.45E-12

5.7 Environment Variables

EnvironmentVariable



Environment variables are substituted by their value when encountered in command input text.

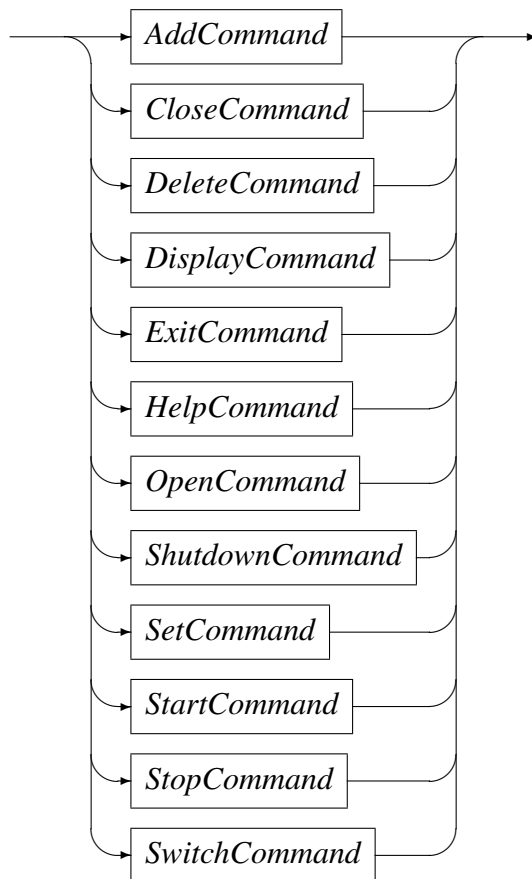
6 Syntax and Semantics

Input to the command processor is either:

- A *Comment*. The whole line is ignored by the command processor, see subsection 5.1 on page 4.
- A *Command*.
- A *Command File Name*. If the input is not a command, the command processor interprets the input as a command file name and, after validating it with `cmdname` [7], will attempt to open it and process each command within it.

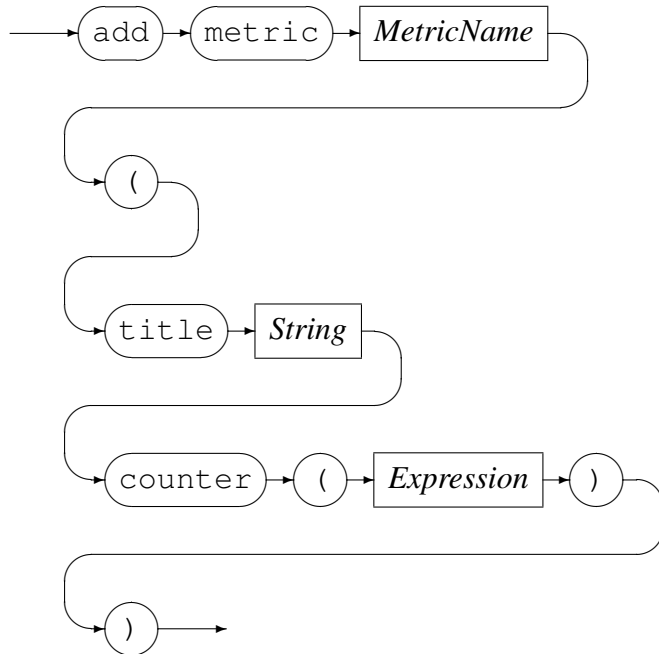
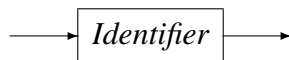
See appendix A on page 19 for an example of a `cmlxlinp` command file.

command



6.1 Add Command

This command is used to add a definition of a metric to be extracted. It describes the binding of the raw Linux performance counters to a `Seifboard` metric.

AddCommand*MetricName*

MetricName is the name of the configured metric in the Serfboard server.

Expression is a regular arithmetic expression where the variables are raw Linux performance counters.

Example

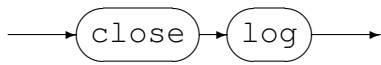
The Serfboard metric `cpu_global_tot_user` is the summation of the two raw Linux performance counters (`cpu.global.user.seconds` and `cpu.global.sys.seconds`):

```
Linux example> add metric cpu_global_tot_user (\
    title "Seconds spent in user + sys mode" \
    counter( cpu.cpu.system \
            + cpu.cpu.user \
            ) \
    )
Added metric cpu_global_tot_user
```

6.2 Close Command

This command is used to either:

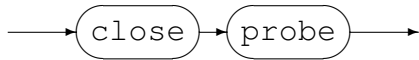
- Close an opened recording log:

**Example:**

Close the previous opened log:

```
Linux example> close log
Log "text(example_probe.txt,mode=w)" closed, record count = 11
```

- Close the active probe:

**Example:**

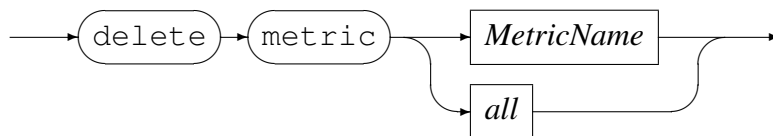
Close an opened probe:

```
Linux example> close probe
Linux example>
```

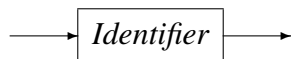
6.3 Delete Command

This command will delete the definition of a previously defined metric and can not be performed when a probe has been started. The metric data will no longer be extracted and sent to a Serfboard server when the probe is restarted.

DeleteCommand



MetricName

**Example:**

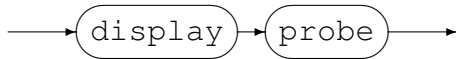
Delete Serfboard metric `cpu_global_tot_user`:

```
Linux example> delete metric cpu_global_tot_user
Error: Probe is active - metric maintenance suspended!
Linux example> stop probe
Probing stopped
Linux example> delete metric cpu_global_tot_user
Metric cpu_global_tot_user deleted
```

6.4 Display Command

This command is used to display various configuration settings:

- Probe status:

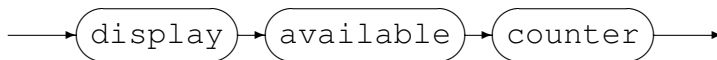


Example:

```

Linux example> display probe
CMLXLINP Code Magus Limited Linux probe V1.0
Build      2020-11-16-17.31.46
CMLXPRB    V2.1
Probe      level 2
Status:                    Idle
Title:                     "Linux proc counters"
Group:                     cmlxlinp
Parameters:                 ''
Polling Interval:          30
Polls per send:            1
Housekeeping Interval:    600
Server:                    Not specified
Cache size:                1.000G
Cache:                     No Cache
Configuration:             cmlxlinp_configuration.cmd
  
```

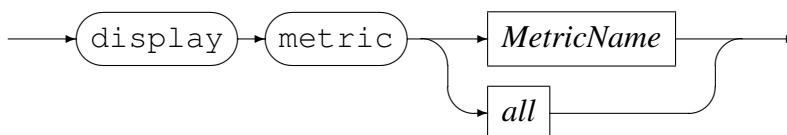
- Available raw Linux performance counters for probing:



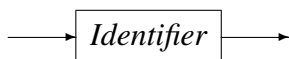
```

Linux example> display available counters
cpu.global.user      : raw total number of clock ticks spent in user mode
cpu.global.sys       : raw total number of clock ticks spent in system mode
cpu.global.idle      : raw total number of clock ticks spent idle
.
.
.
  
```

- Display metrics:



MetricName



Example:

Display all the metrics that have been defined:

```
Linux example> display metric all
metric cpu_global_tot_user
(
  title "Seconds spent in user + sys mode"
  counter( cpu.global.user.seconds + cpu.global.sys.seconds )
)
metric cpu_global_user
.
.
.
```

6.5 Exit Command

This command terminates an interactive session to the command interface of `cmlxlinp` and disconnects the client from the TCP/IP port.

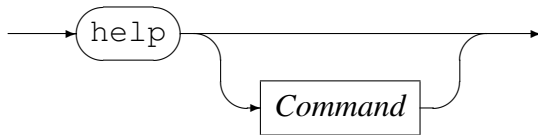
ExitCommand



6.6 Help Command

Help on `cmlxlinp` commands.

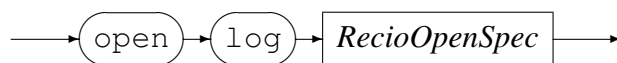
HelpCommand



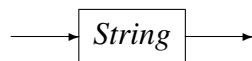
6.7 Open Command

This command is used to either

- Open a log file for recording the metrics sent to Serfboard.



RecioOpenSpec



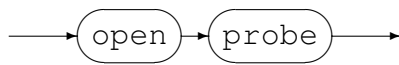
RecioOpenSpec is a `recio` [1] open specification string.

Example:

Open a log file for recording the metrics. The two environment variables will be expanded to the current date and time respectively.

```
Linux example> open log \
    "text(example_probe_D${DATE_YYMMDD}_T${TIME_HHMMSS}).txt,mode=w)"
Log "text(example_probe_D101210_T103857.txt,mode=w)" opened
```

- Initialise the probe:



Example:

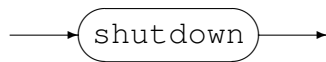
Initialise the probe by opening it:

```
Linux example> open probe
CML Linux probe initialised
```

6.8 Shutdown Command

This command terminates `cmlxlinp`.

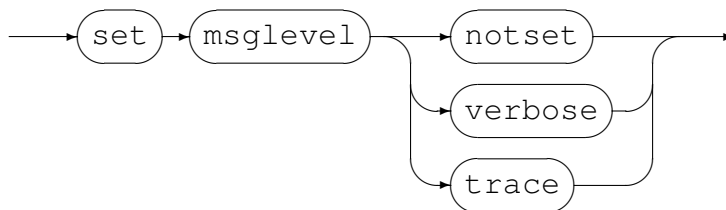
ShutdownCommand



6.9 Set Command

This command is used to set the internal variables and parameters of `cmlxlinp`. If the `set` command results in changing an internal variable or a parameter the response to the `set` command is to display the probe status. The following can be set:

- Set the level of diagnostics produced by `cmlxlinp`:



Example:

Reset the level to not produce diagnostic messages:

```
Linux example> set msglevel notset
Linux example>
```

- Set the title for `cmlxlinp`, this is used for the prompt of the command interface:

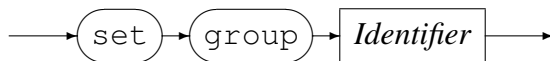


Example:

Set the title of the probe to "Linux example":

```
Linux performance counters> set title "Linux example"
Probe cmlxlinp
  Status:          Initialised
  Title:           "Linux example"
  Group:           cmlxlinp
  Parameters:      ''
  Polling Interval: 30
  Inactive delay:  18000
  Server:          Not specified
Linux example>
```

- Set the Serfboard group name:

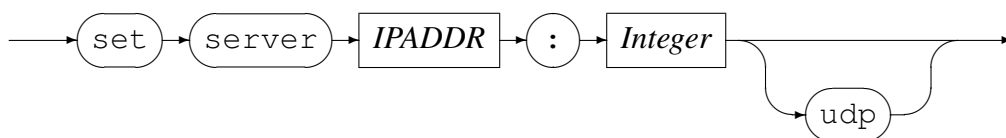


Example:

Set the Serfboard group name to example:

```
Linux example> set group example
Probe cmlxlinp
  Status:          Initialised
  Title:           "Linux example"
  Group:           example
  Parameters:      ''
  Polling Interval: 30
  Inactive delay:  18000
  Server:          Not specified
Linux example>
```

- Set the host address of the Serfboard server to which the metrics are sent:



IPAddress can be specified as a host name or by using the Internet notation of dots

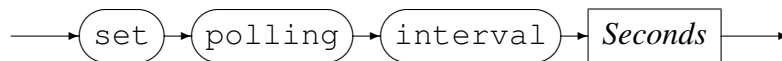
and numbers. The default connection is TCP/IP, but if *udp* is specified, UDP will be used, a connectionless transport without guarantee of delivery.

Example:

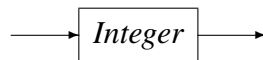
Set the host address of the Serfboard server to `www.codemagus.com`, listening on port 41000 and use UDP :

```
Linux example> set server www.codemagus.com:41000 UDP
Probe cmlxlinp
  Status:          Initialised
  Title:           "Linux example"
  Group:           example
  Parameters:      ''
  Polling Interval: 30
  Inactive delay:  18000
  Server:          www.codemagus.com:41000 UDP Not Connected
```

- Set the polling interval:



Seconds



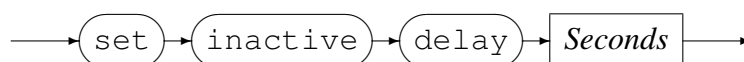
The polling interval is specified in seconds and is the frequency at which the Linux counters will be polled in order to extract metric data. The frequency must be greater than zero and less than 100.

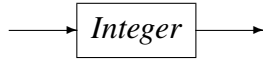
Example:

Set the polling interval to one minute:

```
Linux example> set polling interval 60
Probe cmlxlinp
  Status:          Initialised
  Title:           "Linux example"
  Group:           example
  Parameters:      ''
  Polling Interval: 60
  Inactive delay:  18000
  Server:          www.codemagus.com:41000 UDP Not Connected
Linux example>
```

- Set inactive delay:



Seconds

The inactive delay is specified in seconds and the default value is equivalent to five hours. The `DELAY` defines the amount of time from the last command processed by the command interface before the probe automatically stops extracting metric data by performing a ‘stop probe’ command internally. This feature is always active and prevents the probe from flooding the network with metric data when no longer required.

Example:

Set the inactive delay to one hour:

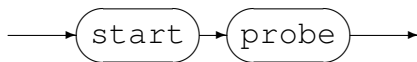
```

Linux example> set inactive delay 36000
Probe cmlxlinp
  Status:          Initialised
  Title:           "Linux example"
  Group:           example
  Parameters:      ''
  Polling Interval: 60
  Inactive delay:  36000
  Server:          www.codemagus.com:41000 UDP Not Connected
Linux example>

```

6.10 Start Command

This command causes the probe to start extracting metric data and sending it on to Serfboard.

StartCommand**Examples:**

Start probing:

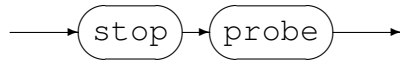
```

Linux example> start probe
Probing started

```

6.11 Stop Command

This command causes the probe to stop extracting metric data and sending it on to Serfboard; `cmlxlinp` reverts back to the idle state.

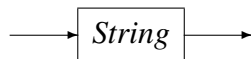
StopCommand**Examples:**

Stop probing:

```
Linux example> stop probe
Probing stopped
```

6.12 Switch Command

Close a previous recording log file and open a new one.

*RecioOpenSpec*

RecioOpenSpec is a `recio [1]` open specification string.

Example:

Close the current log file and open a new log file for recording the metrics. The two environment variables will be expanded to the current date and time respectively.

```
Linux example> switch log \
                "text(example_probe_D${DATE_YYMMDD}_T${TIME_HHMMSS}).txt.txt,mode=w) "
Log "text(example_probe_D101210_T103857.txt.txt,mode=w) " closed,
    record count = 0
Log "text(example_probe_D101210_T105920.txt.txt,mode=w) " opened
```

7 Installation

7.1 Overview

On Linux, cmlxlinp is now delivered as an RPM package. See section 7.2 on page 17 for details.

See the section 7.3 on page 18 for details of the (now deprecated) old TAR package method of install. If you require a non root (and non systemctl) installation then this is still the best option.

7.2 RPM Install Process

This install is very easy, but must be performed under root authority on the target machine.

To install the cmlxlinp probe perform the following steps:

1. Obtain the cmlxlinp RPM package. It usually has a name similar to this:

```
cmlxlinp-2-1.el7.x86_64.rpm
```

2. Copy it to the target system.
3. As root run the install with:

```
yum install cmlxlinp-2-1.el7.x86_64.rpm
```

Once installed the probe will be running. It will listen on port 60054 for incoming configuration requests. Until then it does nothing else.

4. To stop the probe log on as root and issue the command:

```
systemctl stop cmlxlinp.service
```

5. To start the probe again, log on as root and issue the command:

```
systemctl start cmlxlinp.service
```

6. To check the status of the probe, log on as root and issue the command:

```
systemctl status cmlxlinp.service
```

7.3 Package Install Process

The installation of cmlxlinp is quite straight forward and performed by extracting a TARBALL install image. The Linux probe is usually installed under user cmlmnt, but the scripts and code can be run under any user.

To install the cmlxlinp probe perform the following steps:

1. Obtain the install tarball from Code Magus.

Always use the latest TAR install image; They are named according to the following name mask:

```
CODEMAGUS_CENTOS7____X86_64__PRODUCT_BIN_CMLXLINP____Dccyymmdd_Thhmmss.tgz
```

Where cmmmyydd and hhmmdd, are the date and time.

2. On the probe machine, create a user (and group) to run the probe in. For example user cmlmnt and group mntcml. Copy the tarball to the target machine with scp

```
scp <tarball> cmlmnt@targetmachine:
```

3. Log on to the target machine with

```
ssh cmlmnt@targetmachine
```

4. Create the required directories

```
mkdir CodeMagus
mkdir logs
```

5. Change the current directory to the CodeMagus directory:

```
cd CodeMagus
```

6. Decompress the tarball with

```
tar -xvof <tarball>
```

This will create the bin directory and place the executable and shell scripts in it.

7. To start the probe run the shell script:

```
~/CodeMagus/bin/cmlxlinp.sh
```

This will start the keep alive script bin/cmlxlinp.keep in the background.

8. To check that the probe is running use the ps command

```
ps -ef | grep [c]mlxlinp
cmlmnt 10486036 1 0 Jul 23 - 0:00 sh -- /home/cmlmnt/CodeMagus/bin/cmlxlinp.keep
cmlmnt 20644298 10486036 0 Jul 23 - 1:27 cmlxlinp -v
```

9. Once started the keep script will continuously start the program should it be stopped or killed for any reason.

A Example Command File

```
-- File: example.cmd
--
-- Linux probe example.
--
-- Copyright (c) 2021 Code Magus Limited. All rights reserved.
--
-- $Author: hayward $
-- $Date: 2020/12/16 17:51:11 $
-- $Id: example.cmd,v 1.1 2020/12/16 17:51:11 hayward Exp $
-- $Source: /home/cvs/cvsroot/cmlxlinp/documents/example.cmd,v $
-- $Revision: 1.1 $
-- $State: Exp $
--
-- $Log: example.cmd,v $
-- Revision 1.1 2020/12/16 17:51:11 hayward
-- Add documentation for the probe
--
--
--
--
stop probe
close probe
--
set title "Linux example"
set group example
set server www.codemagus.com:41076 UDP
set polling interval 30
--
delete metric all
add class cpu
add class disk
add class memory
add class netinterface
--
set refclock server www.codemagus.com:60001
--
display metric all
open probe
start probe
```

References

- [1] recio: Record Stream I/O Library Version 1. CML Document CML00001-01, Code Magus Limited, July 2008. [PDF](#).
- [2] cmlcmd: Command Utility Version 1. CML Document CML00007-01, Code Magus Limited, July 2008. [PDF](#).
- [3] Serfboard Configuration Guide and Reference Version 1. CML Document CML00023-01, Code Magus Limited, July 2008. [PDF](#).
- [4] Serfboard Instruments Guide and Reference Version 1. CML Document CML00024-01, Code Magus Limited, July 2008. [PDF](#).
- [5] Serfboard Installation Guide and Reference Version 1. CML Document CML00025-01, Code Magus Limited, July 2008. [PDF](#).
- [6] Serfboard User Guide Version 1. CML Document CML00027-01, Code Magus Limited, July 2008. [PDF](#).
- [7] cmdname: Command Name Resolver Library Version 1. CML Document CML00076-01, Code Magus Limited, December 2010. [PDF](#).