



---

## cmlxsolp: Solaris Performance Metric Probe

CML00065-01

---

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



August 16, 2016

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Synopsis</b>	<b>3</b>
<b>3</b>	<b>Invocation</b>	<b>4</b>
<b>4</b>	<b>Command Interface</b>	<b>4</b>
<b>5</b>	<b>Command Elements</b>	<b>4</b>
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
<b>6</b>	<b>Syntax and Semantics</b>	<b>6</b>
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
<b>A</b>	<b>Example Command File</b>	<b>17</b>

# 1 Introduction

This document describes how to use `cmlxsolp` which is a software utility to fetch Solaris performance data using the `kstat` application programming interface. 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 metrics from the raw Solaris platform and send them to `Serfboard` in the correct form `cmlxsolp` needs to be configured. This is done through its command interface.

The `kstat` application programming interface (API) is a collection of C programming language subroutines that execute in user space and uses the `kstat` kernel extension to extract various Solaris performance metrics.

This application reads the values of a list of performance counters to be fetched from the local Solaris host. Each metric is then formatted as a `Serfboard` metric and sent to a `Serfboard` server.

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

`cmlxsolp` 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 Solaris kstat probe V1.0: build 2010-12-03-13.31.19
[cmlxsolp] $Id: cmlxsolp.c,v 1.2 2010/11/29 08:31:37 janvlok Exp $
Copyright (c) 2010 by Code Magus Limited. All rights reserved.
```

```
[Contact: stephen@codemagus.com].
```

```
Usage: cmlxsolp [OPTION...]
```

```
-p, --port={60053|<port>}    Command interface port
-c, --command=<command>      Command to pass to command process
-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 `cmlxsolp`, If not specified it will default to 60053.
- `-c|--command` Specifies a command to be passed to the command interface.
- `-v|--verbose` When specified, `cmlxsolp` operates in a verbose manner.
- `-t|--trace` When specified, `cmlxsolp` writes all activity to `stdout`.

### 3 Invocation

When `cmlxsolp` 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 `cmlxsolp` environment dynamically reconfigured via further commands.

The following sub-sections describe the `cmlxsolp` command interface and commands.

### 4 Command Interface

`cmlxsolp` 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.global.user.seconds  cpu_global_user
```

### 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: 1.7 $'
```

### 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

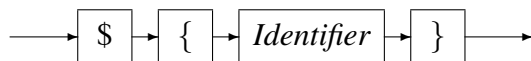
- possibly contains a radix character (decimal point ‘.’).
- is optionally followed by a decimal exponent; consisting of an ‘E’ or ‘e’ followed by an optional plus or minus sign followed by a nonempty sequence of decimal digits that indicates multiplication by a power of 10.

### 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.

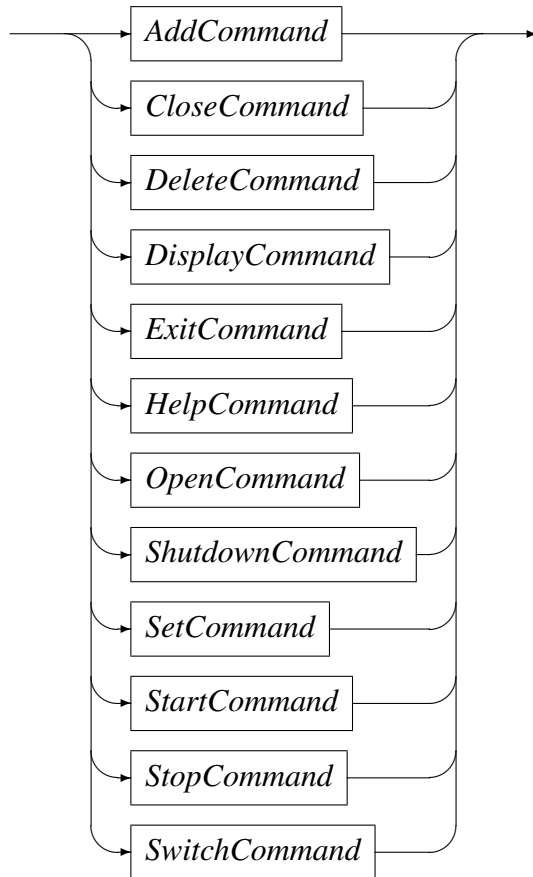
## 6 Syntax and Semantics

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 17 for an example of a `cmlxsolp` command file.

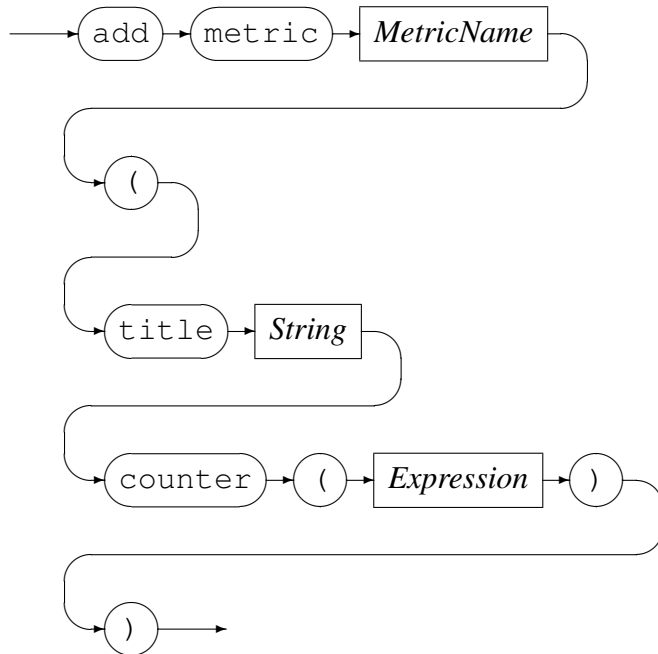
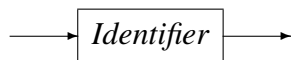
*command*



## 6.1 Add Command

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

**Example**

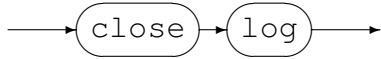
The Serfboard metric `disk_tot_nread` is the summation of the three Solaris performance counters (`sd.0.disk.sd0.nread`, `sd.0.disk.sd1.nread` and `ssd.0.disk.ssd30.nread`):

```
Solaris example> add metric disk_tot_nread ( \
    title "disk_tot_nread" \
    counter \
        ( sd.0.disk.sd0.nread \
          + sd.0.disk.sd1.nread \
          + ssd.0.disk.ssd30.nread \
        ) \
    )
Added metric disk_tot_nread
```

**6.2 Close Command**

This command is used to either:

- Close an opened recording log:

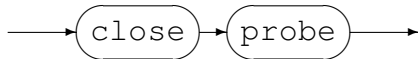


**Example:**

Close the previous opened log:

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

- Close the active probe:



**Example:**

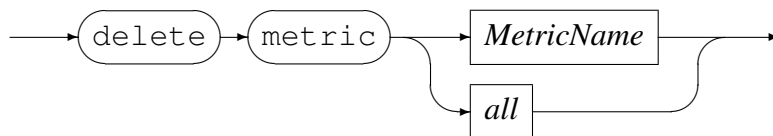
Close an opened probe:

```
Solaris example> close probe
Solaris 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:**

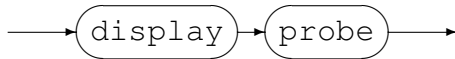
Delete Serfboard metric disk\_tot\_nread:

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

## 6.4 Display Command

This command is used to display various configuration settings:

- Probe status:

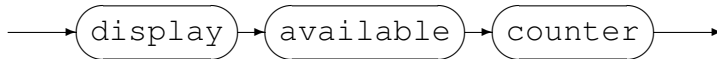


### Example:

```

Solaris example> display probe
Probe cmlxsolp
  Status:           Active
  Title:            "Solaris example"
  Group:            example
  Parameters:       ''
  Polling Interval: 30
  Inactive delay:   36000
  Server:           codemagus.it.nednet.co.za:41068 UDP Connected
  
```

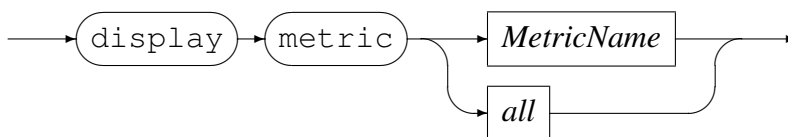
- Available Solaris performance counters for probing:



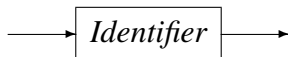
```

Solaris example> display available counters
unix.0.kstat.kstat_types.raw KSTAT_TYPE_NAMED KSTAT_DATA_UINT32 [0]
unix.0.kstat.kstat_types.name=value KSTAT_TYPE_NAMED KSTAT_DATA_UINT32 [1]
unix.0.kstat.kstat_types.interrupt KSTAT_TYPE_NAMED KSTAT_DATA_UINT32 [2]
.
.
.
  
```

- Display metrics:



*MetricName*



### Example:

Display all the metrics that have been defined:

```

Solaris example> display metric all
metric memory_freemem
(
  title "unix.0.pages.system_pages.freemem"
)
  
```

```

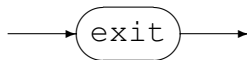
        counter (unix.0.pages.system_pages.freemem)
    )
metric memory_availrmem
.
.
.

```

## 6.5 Exit Command

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

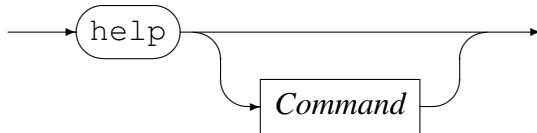
*ExitCommand*



## 6.6 Help Command

Help on `cmlxsolp` commands.

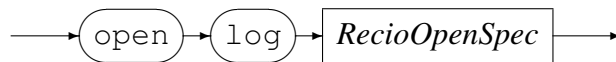
*HelpCommand*



## 6.7 Open Command

This command is used to either

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



*RecioOpenSpec*



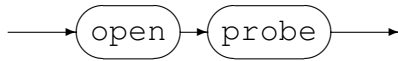
*RecioOpenSpec* is a `recio [1]` open specification string.

### Example:

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

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

- Initialise the probe:



### Example:

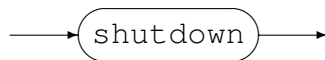
Initialise the probe by opening it:

```
Solaris example> open probe
CML Solaris probe initialised
```

## 6.8 Shutdown Command

This command terminates `cmlxsolp`.

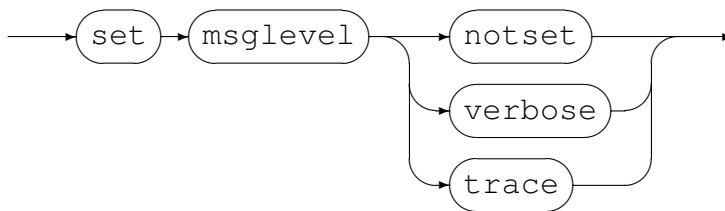
### *ShutdownCommand*



## 6.9 Set Command

This command is used to set the internal variables and parameters of `cmlxsolp`. 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 `cmlxsolp`:



### Example:

Reset the level to not produce diagnostic messages:

```
Solaris example> set msglevel notset
Solaris example>
```

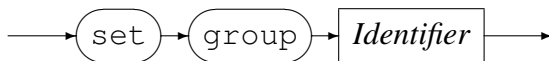
- Set the title for `cmlxsolp`, this is used for the prompt of the command interface:

**Example:**

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

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

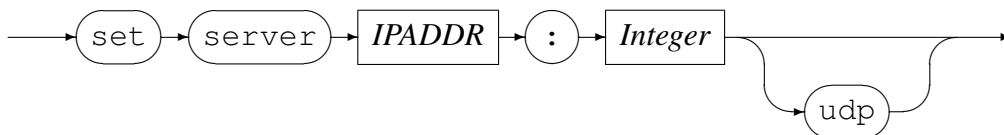
- Set the Serfboard group name:

**Example:**

Set the Serfboard group name to example:

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

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



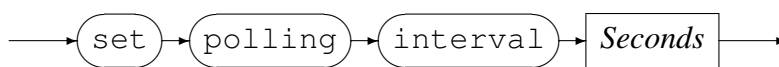
*IPAddress* can be specified as a host name or by using the Internet notation of dots and numbers. The default connection is TCP/IP, but if *udp* is specified, UDP will be used, a connectionless transport without guarantee of delivery.

**Example:**

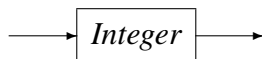
Set the host address of the Serfboard server to `codemagus.it.nednet.co.za`, listening on port 41000 and use UDP :

```
Solaris example> set server codemagus.it.nednet.co.za:41000 UDP
Probe cmlxsolp
  Status:          Initialised
  Title:           "Solaris example"
  Group:           example
  Parameters:      ''
  Polling Interval: 30
  Inactive delay:  18000
  Server:         codemagus.it.nednet.co.za:41000 UDP Not Connected
```

- Set the polling interval:



*Seconds*



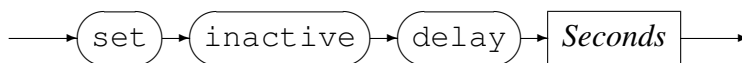
The polling interval is specified in seconds and is the frequency at which the Solaris performance counters will be polled in order to extract metric data. The frequency must be greater than zero and less than 100.

#### **Example:**

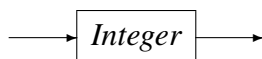
Set the polling interval to one minute:

```
Solaris example> set polling interval 60
Probe cmlxsolp
  Status:          Initialised
  Title:           "Solaris example"
  Group:           example
  Parameters:      ''
  Polling Interval: 60
  Inactive delay:  18000
  Server:         codemagus.it.nednet.co.za:41000 UDP Not Connected
```

- 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.

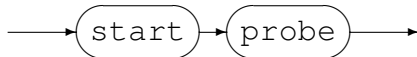
**Example:**

Set the inactive delay to one hour:

```
Solaris example> set inactive delay 36000
Probe cmlxsolp
  Status:          Initialised
  Title:           "Solaris example"
  Group:           example
  Parameters:      ''
  Polling Interval: 60
  Inactive delay:  36000
  Server:         codemagus.it.nednet.co.za:41000 UDP Not Connected
```

**6.10 Start Command**

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

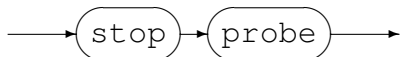
*StartCommand***Examples:**

Start probing:

```
Solaris 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; `cmlxsolp()` reverts back to the idle state.

*StopCommand***Examples:**

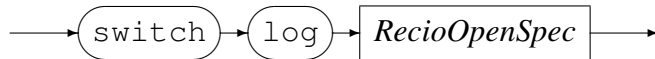
Stop probing:

```
Solaris example> stop probe
Probing stopped
```

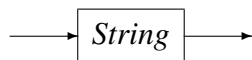


## 6.12 Switch Command

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.

```

Solaris 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
  
```

## A Example Command File

```

-- File example.cmd
--
-- Solaris probe example.
--
-- Copyright (c) 2010 Code Magus Limited. All rights reserved.
--
-- $Author: janvlok $
-- $Date: 2010/12/23 09:26:50 $
-- $Id: example.cmd,v 1.1 2010/12/23 09:26:50 janvlok Exp $
-- $Source: /home/cvs/cvsroot/cmlxsolp/documents/example.cmd,v $
-- $Revision: 1.1 $
-- $State: Exp $
--
-- $Log: example.cmd,v $
-- Revision 1.1 2010/12/23 09:26:50 janvlok
-- Take on
--
--
-- first time we need the select, later we need the close
close probe
set title "Solaris example"
set group example
set server codemagus.it.nednet.co.za:41068 UDP
set polling interval 30
delete metric all
-----
add metric memory_freemem ( \
    title "unix.0.pages.system_pages.freemem" \
    counter (unix.0.pages.system_pages.freemem) \
)
add metric memory_availrmem ( \
    title "unix.0.pages.system_pages.availrmem" \
    counter (unix.0.pages.system_pages.availrmem) \
)
-----
add metric cpu_tot ( \
    title "cpu_tot" \
    counter \
        ( cpu_stat.0.misc.cpu_stat0.user \
        + cpu_stat.0.misc.cpu_stat1.user \
        + cpu_stat.0.misc.cpu_stat2.user \
        + cpu_stat.0.misc.cpu_stat3.user \
        + cpu_stat.0.misc.cpu_stat8.user \
        + cpu_stat.0.misc.cpu_stat9.user \
        + cpu_stat.0.misc.cpu_stat0.kernel \
        + cpu_stat.0.misc.cpu_stat1.kernel \
        + cpu_stat.0.misc.cpu_stat2.kernel \
        + cpu_stat.0.misc.cpu_stat3.kernel \
        + cpu_stat.0.misc.cpu_stat0.wait \
        + cpu_stat.0.misc.cpu_stat1.wait \

```

```
        + cpu_stat.0.misc.cpu_stat2.wait \
        + cpu_stat.0.misc.cpu_stat3.wait \
    ) \
)
add metric cpu_tot_idle ( \
    title "cpu_tot_idle" \
    counter \
        ( cpu_stat.0.misc.cpu_stat0.idle \
        + cpu_stat.0.misc.cpu_stat1.idle \
        + cpu_stat.0.misc.cpu_stat2.idle \
        + cpu_stat.0.misc.cpu_stat3.idle \
        ) \
    )
add metric cpu_tot_kernel ( \
    title "cpu_tot_kernel" \
    counter \
        ( cpu_stat.0.misc.cpu_stat0.kernel \
        + cpu_stat.0.misc.cpu_stat1.kernel \
        + cpu_stat.0.misc.cpu_stat2.kernel \
        + cpu_stat.0.misc.cpu_stat3.kernel \
        ) \
    )
add metric cpu_tot_user ( \
    title "cpu_tot_user" \
    counter \
        ( cpu_stat.0.misc.cpu_stat0.user \
        + cpu_stat.0.misc.cpu_stat1.user \
        + cpu_stat.0.misc.cpu_stat2.user \
        + cpu_stat.0.misc.cpu_stat3.user \
        ) \
    )
add metric cpu_tot_wait ( \
    title "cpu_tot_wait" \
    counter \
        ( cpu_stat.0.misc.cpu_stat0.wait \
        + cpu_stat.0.misc.cpu_stat1.wait \
        + cpu_stat.0.misc.cpu_stat2.wait \
        + cpu_stat.0.misc.cpu_stat3.wait \
        ) \
    )
add metric disk_tot_nread ( \
    title "disk_tot_nread" \
    counter \
        ( sd.0.disk.sd0.nread \
        + sd.0.disk.sd1.nread \
        + ssd.0.disk.ssd30.nread \
        ) \
    )
add metric disk_tot_nwritten ( \
    title "disk_tot_nwritten" \
    counter \
        ( sd.0.disk.sd0.nwritten \
```

```
        + sd.0.disk.sd1.nwritten \  
        + ssd.0.disk.ssd30.nwritten \  
    ) \  
)  
add metric disk_tot_reads ( \  
    title "disk_tot_reads" \  
    counter \  
        ( sd.0.disk.sd0.reads \  
        + sd.0.disk.sd1.reads \  
        + ssd.0.disk.ssd30.reads \  
        ) \  
    ) \  
)  
add metric disk_tot_writes ( \  
    title "disk_tot_writes" \  
    counter \  
        ( sd.0.disk.sd0.writes \  
        + sd.0.disk.sd1.writes \  
        + ssd.0.disk.ssd30.writes \  
        ) \  
    ) \  
)  
add metric net_tot_rbytes ( \  
    title "net_tot_rbytes" \  
    counter \  
        ( bge.0.net.mac.rbytes \  
        + bge.0.net.bge0.rbytes \  
        + bge.2.net.mac.rbytes \  
        + bge.2.net.bge2.rbytes \  
        + nxge.0.net.mac.rbytes \  
        + nxge.0.net.nxge0.rbytes \  
        ) \  
    ) \  
)  
add metric net_tot_obytes ( \  
    title "net_tot_obytes" \  
    counter \  
        ( bge.0.net.mac.obytes \  
        + bge.0.net.bge0.obytes \  
        + bge.2.net.mac.obytes \  
        + bge.2.net.bge2.obytes \  
        + nxge.0.net.mac.obytes \  
        + nxge.0.net.nxge0.obytes \  
        ) \  
    ) \  
)  
add metric net_tot_ipackets ( \  
    title "net_tot_ipackets" \  
    counter \  
        ( bge.0.net.mac.ipackets \  
        + bge.0.net.bge0.ipackets \  
        + bge.2.net.mac.ipackets \  
        + bge.2.net.bge2.ipackets \  
        + nxge.0.net.mac.ipackets \  
        + nxge.0.net.nxge0.ipackets \  
        ) \  
    ) \  
)
```

```
)
add metric net_tot_opackets ( \
  title "net_tot_opackets" \
  counter \
    ( bge.0.net.mac.opackets \
      + bge.0.net.bge0.opackets \
      + bge.2.net.mac.opackets \
      + bge.2.net.bge2.opackets \
      + nxge.0.net.mac.opackets \
      + nxge.0.net.nxge0.opackets \
    ) \
  )
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](#).