



cmlxaixp: AIX Performance Metric Probe

CML00045-01

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



August 16, 2016

Contents

1	Introduction	2
1.1	Further Reading	2
2	Synopsis	3
3	Invocation	3
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	5
5.7	Environment Variables	6
6	Syntax and Semantics	6
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
A	Example Command File	17

1 Introduction

This document describes how to use `cmlxaixp` which is a software utility to fetch AIX performance data using the `perfstat` 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 raw counters from the AIX platform and send them to `Serfboard` in the correct form `cmlxaixp` needs to be configured. This is done through its command interface.

The `perfstat` application programming interface (API) is a collection of C programming language subroutines that execute in user space and uses the `perfstat` kernel extension to extract various AIX performance metrics. System component information is also retrieved from the Object Data Manager (ODM) and returned with the performance metrics

`cmlxaixp` reads the values of a list of performance metrics to be fetched from the local AIX 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

`cmlxaixp` 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 AIX PERF probe V1.0: build 2010-12-07-10.57.22
[./cmlxaixp] $Id: cmlxaixp.c,v 1.20 2010/12/03 16:23:16 janvlok Exp $
Copyright (c) 2009 by Code Magus Limited. All rights reserved.
[Contact: stephen@codemagus.com].
```

```
Usage: cmlxaixp [OPTION...]
  -p, --port={60052|<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 `cmlxaixp`, If not specified it will default to 60052.
- `-c|--command` Specifies a command to be passed to the command interface.
- `-v|--verbose` When specified, `cmlxaixp` operates in a verbose manner.
- `-t|--trace` When specified, `cmlxaixp` writes all activity to `stdout`.

3 Invocation

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

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

4 Command Interface

`cmlxaixp` 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:

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

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: 1.12 $'
```

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

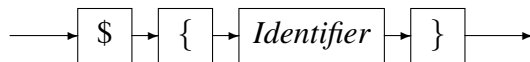
- possibly contains a radix character (decimal point ‘.’).
- is optionally followed by a decimal exponent; consisting of an ‘E’ or ‘e’ followed by an optional plus or minus sign followed by a nonempty sequence of decimal digits that indicates multiplication by a power of 10.

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.

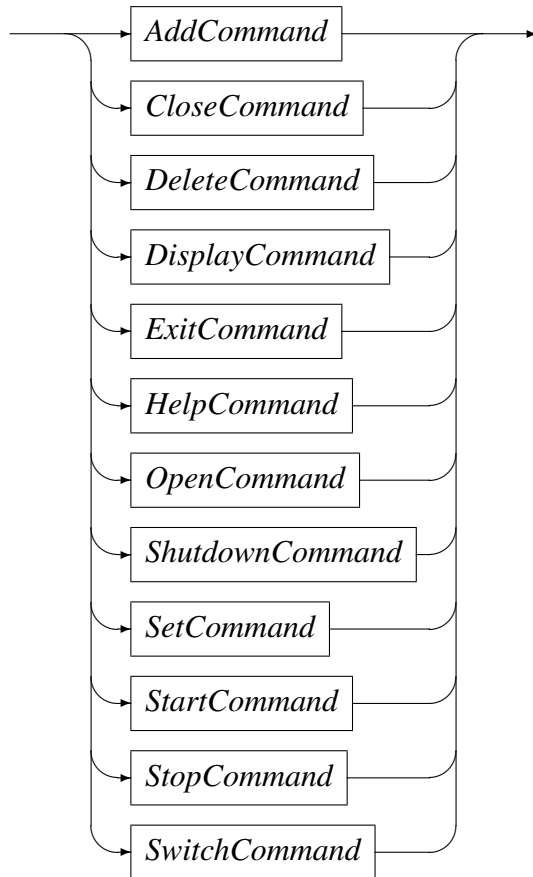
6 Syntax and Semantics

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.

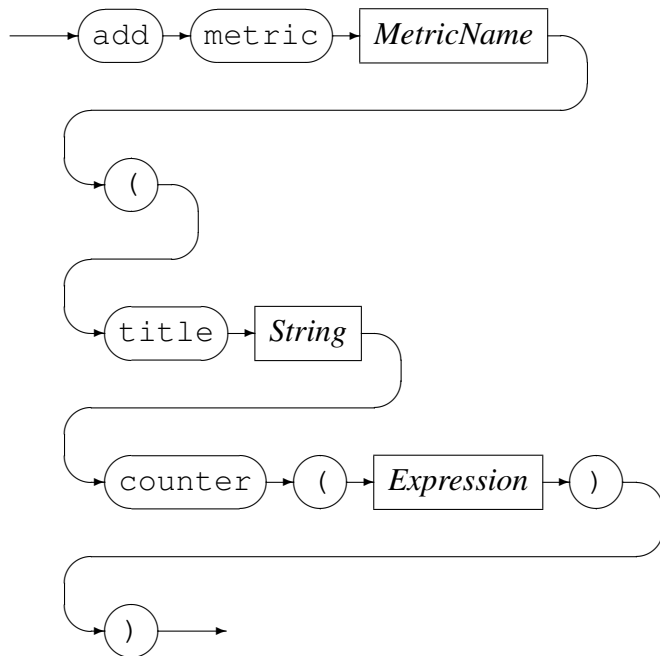
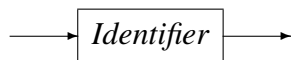
See appendix [A](#) on page 17 for an example of a `cmlxaixp` command file.

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 AIX 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 AIX performance counters.

Example

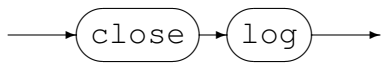
The Serfboard metric `cpu_global_tot_user` is the summation of the two raw AIX performance counters (`cpu.global.user.seconds` and `cpu.global.sys.seconds`):

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

6.2 Close Command

This command is used to either:

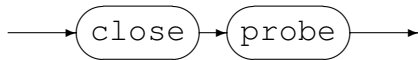
- Close an opened recording log:

**Example:**

Close the previous opened log:

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

- Close the active probe:

**Example:**

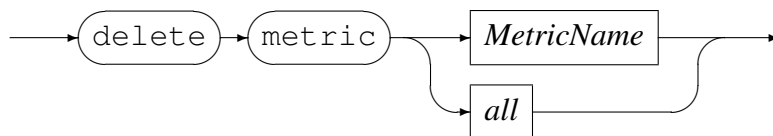
Close an opened probe:

```
AIX example> close probe
AIX example>
```

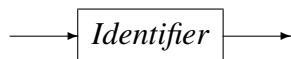
6.3 Delete Command

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:**

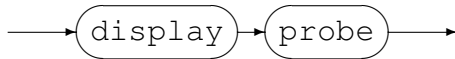
Delete Serfboard metric `cpu_global_tot_user`:

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

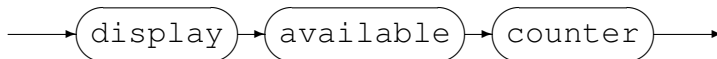


Example:

```

AIX example> display probe
Probe cmlxaixp
  Status:           Active
  Title:            "AIX example"
  Group:            example
  Parameters:       ''
  Polling Interval: 30
  Inactive delay:   18000
  Server:           localhost:41076 Connected
  
```

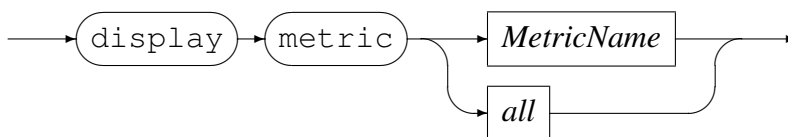
- Available raw AIX performance counters for probing:



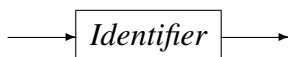
```

AIX 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:

```

AIX example> display metric all
metric cpu_global_tot_user
(
  title "Seconds spent in user + sys mode"
  
```

```

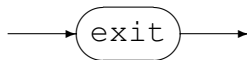
        counter( cpu.global.user.seconds + cpu.global.sys.seconds ))
    )
metric cpu_global_user
.
.
.

```

6.5 Exit Command

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

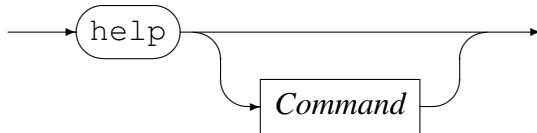
ExitCommand



6.6 Help Command

Help on `cmlxaixp` commands.

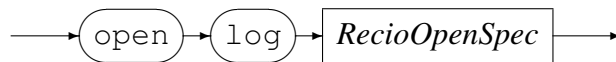
HelpCommand



6.7 Open Command

This command is used to either

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



RecioOpenSpec



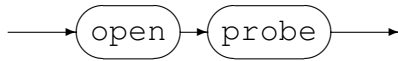
RecioOpenSpec is a `recio` [1] open specification string.

Example:

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

```
AIX 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:

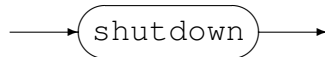
Initialise the probe by opening it:

```
AIX example> open probe
CML AIX probe initialised
```

6.8 Shutdown Command

This command terminates `cmlxaixp`.

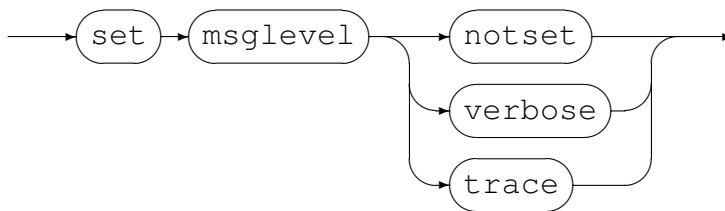
ShutdownCommand



6.9 Set Command

This command is use to set the internal variables and parameters of `cmlxaixp`. 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 `cmlxaixp`:



Example:

Reset the level to not produce diagnostic messages:

```
AIX example> set msglevel notset
AIX example>
```

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

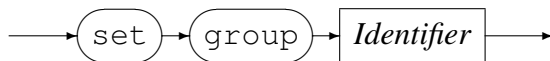
**Example:**

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

```

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

- Set the Serfboard group name:

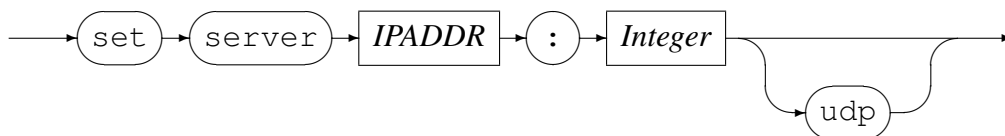
**Example:**

Set the Serfboard group name to example:

```

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

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



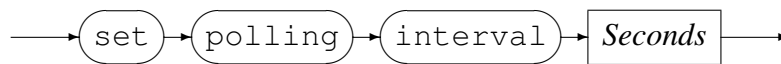
IPAddress can be specified as a host name or by using the Internet notation of dots and numbers. The default connection is TCP/IP, but if *udp* is specified, UDP will be used, a connectionless transport without guarantee of delivery.

Example:

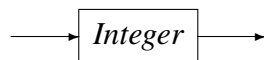
Set the host address of the Serfboard server to `codemagus.it.nednet.co.za`, listening on port 41000 and use UDP :

```
AIX example> set server codemagus.it.nednet.co.za:41000 UDP
Probe cmlxaixp
  Status:          Initialised
  Title:           "AIX 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



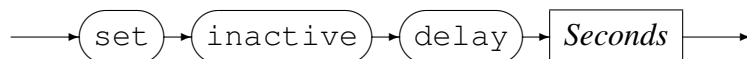
The polling interval is specified in seconds and is the frequency at which the AIX counters will be polled in order to extract metric data. The frequency must be greater than zero and less than 100.

Example:

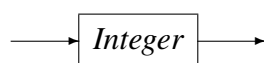
Set the polling interval to one minute:

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

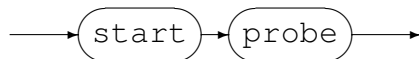
Set the inactive delay to one hour:

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

6.10 Start Command

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

StartCommand



Examples:

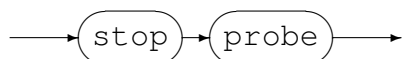
Start probing:

```
AIX 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; cmlxaixp reverts back to the idle state.

StopCommand



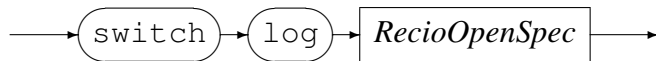
Examples:

Stop probing:

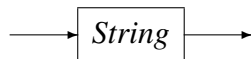

```
AIX example> stop probe
Probing stopped
```

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.

```
AIX 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
--
-- AIX PERF probe example.
--
-- Copyright (c) 2010 Code Magus Limited. All rights reserved.
--
-- $Author: janvlok $
-- $Date: 2010/12/22 11:29:07 $
-- $Id: example.cmd,v 1.1 2010/12/22 11:29:07 janvlok Exp $
-- $Source: /home/cvs/cvsroot/cmlxaixp/documents/example.cmd,v $
-- $Revision: 1.1 $
-- $State: Exp $
--
-- $Log: example.cmd,v $
-- Revision 1.1 2010/12/22 11:29:07 janvlok
-- Take on
--
-- first time we need the select, later we need the close
stop probe
close probe
--
--
set title "AIX example"
set group example
set server codemagus.it.nednet.co.za:41076 UDP
set server localhost:41076
set polling interval 30
delete metric all
add metric cpu_global_tot_user (\
    title "Seconds spent in user + sys mode" \
    counter( cpu.global.user.seconds \
        + cpu.global.sys.seconds \
        ) \
    )
add metric cpu_global_user (\
    title "Seconds spent in user mode" \
    counter(cpu.global.user.seconds) \
    )
add metric cpu_global_sys (\
    title "Seconds spent in system mode" \
    counter(cpu.global.sys.seconds) \
    )
add metric cpu_global_idle (\
    title "Seconds spent in idle mode" \
    counter(cpu.global.idle.seconds) \
    )
add metric cpu_global_wait (\
    title "Seconds spent in wait for io" \
    counter(cpu.global.wait.seconds) \
    )
```

```
-----
add metric disk_global_wblks (\
    title "512 bytes blocks written to all disks" \
    counter(disk.global.wblks) \
)
add metric disk_global_rblks (\
    title "512 bytes blocks read from all disks" \
    counter(disk.global.rblks) \
)
-----
add metric memory_global_pgins (\
    title "number of pages paged out" \
    counter(memory.global.pgins) \
)
add metric memory_global_pgouts (\
    title "number of bad pages" \
    counter(memory.global.pgouts) \
)
add metric memory_global_pgspins (\
    title "number of page ins from paging space" \
    counter(memory.global.pgspins) \
)
add metric memory_global_pgspouts (\
    title "number of page outs from paging space" \
    counter(memory.global.pgspouts) \
)
-----
add metric netinterface_en4_ipackets (\
    title "Number of packets received" \
    counter(netinterface.en4.ipackets) \
)
add metric netinterface_en4_opackets (\
    title "Number of packets sent" \
    counter(netinterface.en4.opackets) \
)
add metric netinterface_en4_ibytes (\
    title "Number of bytes received" \
    counter(netinterface.en4.ibytes) \
)
add metric netinterface_en4_obytes (\
    title "Number of bytes sent" \
    counter(netinterface.en4.obytes) \
)
-----
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](#).