cmlxrmfp: z/OS RMF Performance Metric Probe

CML00122-01

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com

August 5, 2022

Contents

1 Introduction

This document describes how to use the cmlxrmfp which is a software utility to fetch RMF performance data using the ERBSMFI 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 z/OS platform and send them to Serfboard in the correct form cmlxrmfp needs to be configured. This is done through its command interface.

The ERBSMFI is a z/OS callable application programming interface (API) that executes in user space. It extracts various SMF performance metrics.

cmlxrmfp reads the values of a list of performance metrics to be fetched from the local z/OS 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

`cmlxrmfp` can be invoked from the command line (through BPXBATCH, telnet or ssh, but not omvs) 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 2017-09-12-04.39.51
[./bin/cmlxrmfp] $Id: cmlxrmfp_help.tex,v 1.1 2022/08/05 14:13:14 hayward Exp $
Copyright (c) 2009-2022 by Code Magus Limited. All rights reserved.
 [Contact: stephen@codemagus.com].
refclock(): Code Magus Limited - Reference Clock Synchronisation:
Environmental variable 'CODEMAGUS_REFCLOCK_SERVER' is not set!
Local clock of client RMF0 continuing with unsynchronised clocks

Usage: cmlxrmfp [OPTION...]
  -p, --port={60055|<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 `cmlxrmfp`, If not specified it will default to 60055.
- '`-c|--command`' Specifies a command to passed to the command interface.
- '`-n|--name`' Specifies an optional instance name
- '`-v|--verbose`' When specified, `cmlxrmfp` operates in a verbose manner.
- '`-t|--trace`' When specified, `cmlxrmfp` writes all activity to `stdout`.

3 Invocation

When cmlxrmfp 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 cmlxrmfp environment dynamically reconfigured via further commands.

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

4 Command Interface

cmlxrmfp 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:

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:

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

```
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: 1.1 $’
```

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

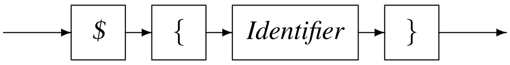
- possibly contains a radix character (decimal point ‘.’).
- is optionally followed by a decimal exponent; consisting of an ‘E’ or ‘e’ followed by an optional plus or minus sign followed by a nonempty sequence of decimal digits that indicates multiplication by a power of 10.

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.

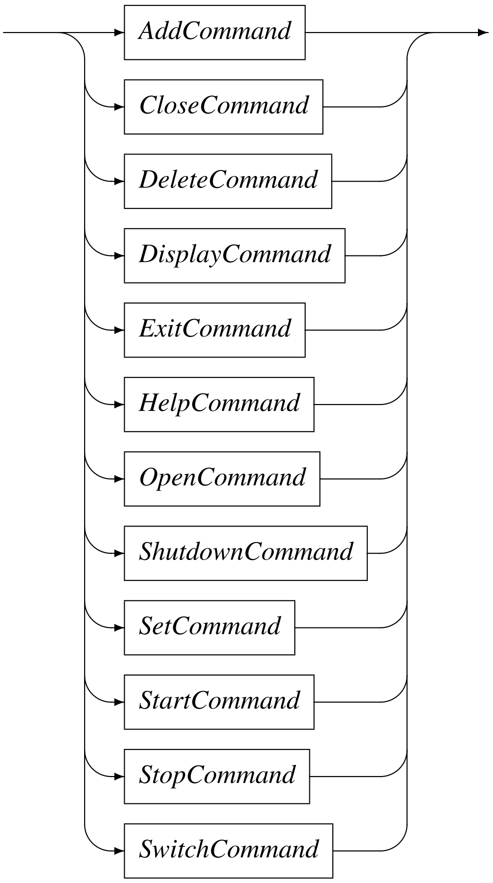
6 Syntax and Semantics

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 `cmlxrmfp` command file.

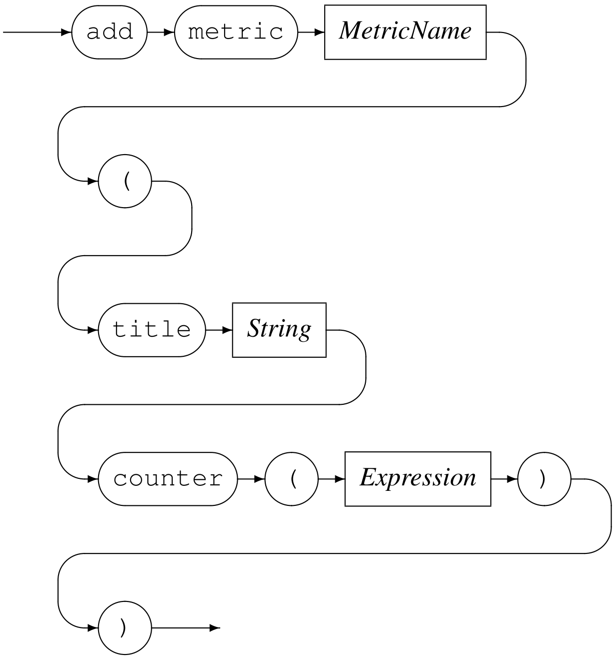
command



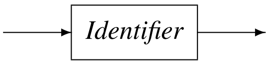
6.1 Add Command

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

Example

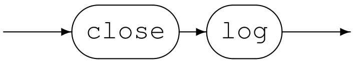
The `Serfboard` metric `cpu_global_tot_user` is the summation of the two raw RMF performance counters (`cpu.global.user.seconds` and `cpu.global.sys.seconds`):

```
RMF example> add metric db2_cpu_total (\
   title "Total CPU for all DB2" \
   counter( job.RMF0.DSN1.R791TCPU + job.RMF0.DSN2.R791TCPU ) \
   )
Added metric db2_cpu_total
```

6.2 Close Command

This command is used to either:

- Close an opened recording log:

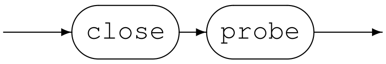


Example:

Close the previous opened log:

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

- Close the active probe:



Example:

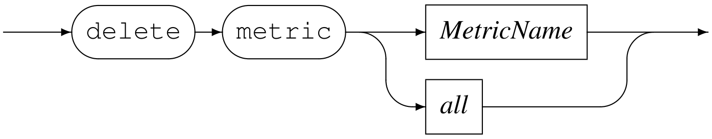
Close an opened probe:

```
RMF example> close probe
RMF example>
```

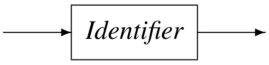
6.3 Delete Command

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:

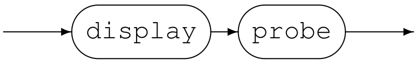
Delete `Serfboard` metric `db2_cpu_total`:

```
RMF example> delete metric db2_cpu_total
Error: Probe is active - metric maintenance suspended!
RMF example> stop probe
Probing stopped
RMF example> delete metric db2_cpu_total
Metric db2_cpu_total deleted
```

6.4 Display Command

This command is used to display various configuration settings:

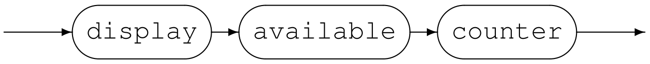
- Probe status:



Example:

```
RMF example> display probe
   CMLXRMFP Code Magus Limited Linux probe V1.0
   Build    2017-09-12-04.39.51
   CMLXPRB  V2.0
   Probe    level 2
   Status:                 Active
   Title:                  "RMF Probe RMF0"
   Group:                  RMF0
   Parameters:             ''
   Polling Interval:       10
   Polls per send:         1
   Housekeeping Interval: 600
   Server:                 81.137.201.165:41212 UDP
   Cache size:             10.000M
   Cache:                  No Cache
   Configuration:          cmlxrmfp_RMF0_configuration.cmd
```

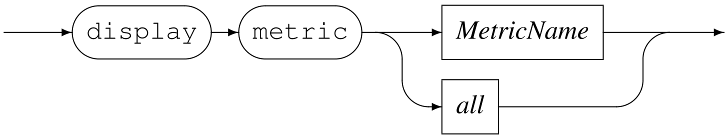
- Available raw RMF performance counters for probing:



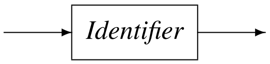
```
RMF Probe RMF0> display available counters
cpu.RMF0.prsm_cp_util_percent  RMF0: PR/SM CP Uilisation percentage
spag.RMF0.page_ins_per_second  RMF0: Demand page-ins per second
cpu.RMF0.mvs_srm_cpu_util_percent RMF0: MVS or SRM CPU util percent
cpu.RMF0.zaap_util_percent     RMF0: zAAP CPU Util percent
cpu.RMF0.ziip_util_percent     RMF0: zIIP CPU Util percent
Current running processes:
job.RMF0.CSQ2CHIN.R791ASID     Address space identifier.
...
job.RMF0.CSQ2CHIN.R795WMS      Total service units for transaction (zeros when A
job.RMF0.CSQ2MSTR.R791ASID     Address space identifier.
...
job.RMF0.CSQ2MSTR.R795WMS      Total service units for transaction (zeros when A
job.RMF0.DSN1DIST.R791ASID     Address space identifier.
...
job.RMF0.DSN1DIST.R795WMS      Total service units for transaction (zeros when A
job.RMF0.DSN1MSTR.R791ASID     Address space identifier.
...
job.RMF0.DSN1MSTR.R795WMS      Total service units for transaction (zeros when A
job.RMF0.FTRQQAA0.R791ASID     Address space identifier.
...
job.RMF0.FTRQQAA0.R795WMS      Total service units for transaction (zeros when A
job.RMF0.TCPIP.R791ASID        Address space identifier.
...
job.RMF0.TCPIP.R795WMS         Total service units for transaction (zeros when A
Current devices:
device.RMF0.PPGQ24.R799SSC     Start subchannel (SSCH) count.
device.RMF0.PPGQ24.R799CNN     Device connect time.
...
device.RMF0.PPGQ05.R799SSC     Start subchannel (SSCH) count.
device.RMF0.PPGQ05.R799CNN     Device connect time.
```

```
...
device.RMF0.PPGQ05.R799PSM     Number of successful PAV samples.
```

- Display metrics:



MetricName



Example:

Display all the metrics that have been defined:

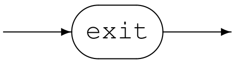
```
display metric all

metric cpu.RMF0.prsm_cp_util_percent
   (
   title "prsm_cp_util_percent"
   counter('cpu.RMF0.prsm_cp_util_percent'))
   )
metric cpu.RMF0.mvs_srm_cpu_util_percent
   (
   title "mvs_srm_cpu_util_percent"
   counter('cpu.RMF0.mvs_srm_cpu_util_percent'))
   )
metric cpu.RMF0.zaap_util_percent
   (
   title "zaap_util_percent"
   counter('cpu.RMF0.zaap_util_percent'))
   )
metric cpu.RMF0.ziip_util_percent
   (
   title "ziip_util_percent"
   counter('cpu.RMF0.ziip_util_percent'))
   )
metric spag.RMF0.page_ins_per_second
   (
   title "page_ins_per_second"
   counter('spag.RMF0.page_ins_per_second'))
   )
metric job.RMF0.DSN1MQ00.R791ASID
   (
   title "Address space identifier."
   counter(job.RMF0.DSN1MQ00.R791ASID)
   )
metric job.RMF0.DSN1MQ00.R791TTOD
   (
   title "Real time into transaction (milliseconds)."
   counter(job.RMF0.DSN1MQ00.R791TTOD)
   )
```

```
metric job.RMF0.DSN1MQ00.R791TAS
   (
   title "Type of user Contents."
   counter(job.RMF0.DSN1MQ00.R791TAS)
   )
```

6.5 Exit Command

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

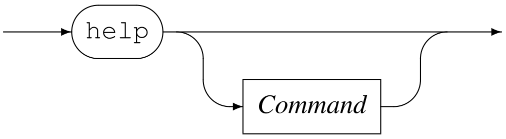
ExitCommand



6.6 Help Command

Help on `cmlxrmfp` commands.

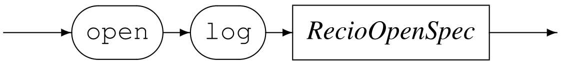
HelpCommand



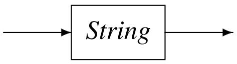
6.7 Open Command

This command is used to either

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



RecioOpenSpec



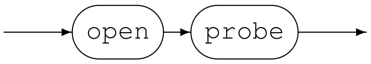
RecioOpenSpec is a `recio` [1] open specification string.

Example:

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

```
RMF 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:

```
RMF example> open probe
CML RMF probe initialised
```

6.8 Shutdown Command

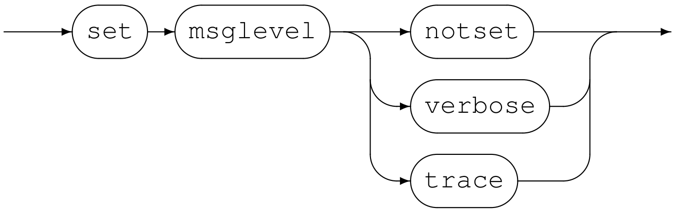
This command terminates `cmlxrmfp`.

ShutdownCommand

6.9 Set Command

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

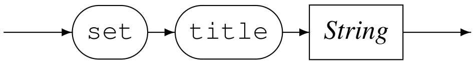


Example:

Reset the level to not produce diagnostic messages:

```
RMF example> set msglevel notset
RMF example>
```

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

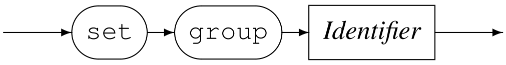


Example:

Set the title of the probe to ”RMF example”:

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

- Set the Serfboard group name:

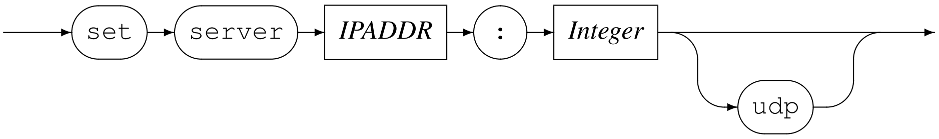


Example:

Set the Serfboard group name to example:

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

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



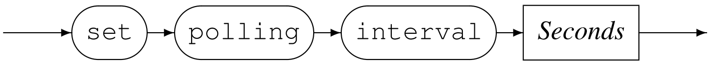
IPAddress can be specified as a host name or by using the Internet notation of dots and numbers. The default connection is TCP/IP, but if *udp* is specified, UDP will be used, a connectionless transport without guarantee of delivery.

Example:

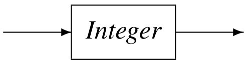
Set the host address of the Serfboard server to www.codemagus.com, listening on port 41000 and use UDP :

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

- Set the polling interval:



Seconds



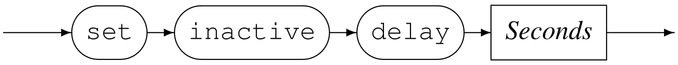
The polling interval is specified in seconds and is the frequency at which the RMF counters will be polled in order to extract metric data. The frequency must be greater than zero and less than 100.

Example:

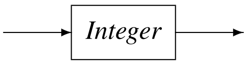
Set the polling interval to one minute:

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

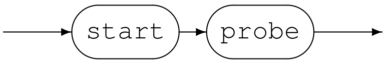
Set the inactive delay to one hour:

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

6.10 Start Command

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

StartCommand



Examples:

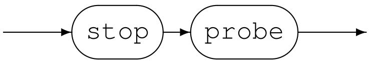
Start probing:

```
RMF 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`; `cmlxrmfp` reverts back to the idle state.

StopCommand



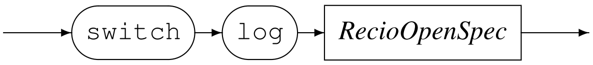
Examples:

Stop probing:

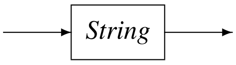
```
RMF example> stop probe
Probing stopped
```

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.

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

7 Installation

7.1 Overview

The installation of cmlxrmfp is quite straight forward and performed by extracting a TAR install package.

7.2 Process

To install the cmlxrmfp perform the following steps:

1. Obtain the install package. This is usually on request to Code Magus.

   Only the latest TAR install image should be used and is named similarly to this mask

   ```
   CODEMAGUS_ZOSn.nn_____Z390____PRODUCT_BIN_CMLXRMFP___Dccyymmdd_Thhmmss.tgz
   ```

   Where `ccmmyydd` and `hhmmdd`, are the date and time, and `n.nn` is the z/OS version of the OS when the package was generated.

2. On the probe machine, under Unix System Services, create a user (and group) to run the probe in. Usually user cmlmnt. Copy the tarball to the target machine with scp or sftp.

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

3. Log on to the target machine with

   ```
   ssh cmlmnt@newserver.co.za
   ```

4. Create the required directories. This can be called anything, but as per the Linux convention `/usr/local` has been chosen. Of course the user under which the install is done requires suitable file system access.

   ```
   cd /usr/local
   mkdir CodeMagus
   mkdir logs
   ```

5. Change the current directory to the CodeMagus directory:

   ```
   cd /usr/local/CodeMagus
   ```

6. Decompress the tarball with

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

   This will create the bin directory and place the executable it.

7. The probe runs in a shell and to start it, submit JCL similar to this:

   ```
   //CMLXRMF0 JOB  (0100),'CML',CLASS=R,MSGCLASS=X,NOTIFY=&SYSUID,
   //    TYPRUN=HOLD,TIME=1440,REGION=0M
   //*
   // SET DIR='/usr/local/CodeMagus'
   //*
   //CMLXRMFP EXEC PGM=BPXBATCH,                                          X
   //         PARM='pgm &DIR/bin/cmlxrmfp -n RMF0 -v'
   //STEPLIB  DD   DSN=QNCT00.SRDA1.LOADLIB,DISP=SHR
   ```

```
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//STDIN    DD   SYSOUT=*
//STDOUT   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STDENV   DD   *
_BPX_SHAREAS=YES
CODEMAGUS_TEMP_DIRECTORY=/usr/local/CodeMagus/etc
/*
//*
```

8. To check that the probe is running either use SDSF or in a shell the `ps` command

```
ps -ef | grep [c]mlxrmfp
```

```
U000001 16842983 33620063 - Aug 03 ? 17:45 /qaplex/usr/local/CodeMagus/bin/cmlxrmfp -n RMF0 -v
```

A Example Command File

```
set title "RMF Probe RMF0"
set group RMF0
set polling interval 10
set polls per send 1
set housekeeping interval 600
set server 81.137.201.165:41212 UDP
set cache size 10485760
add generic metric DB2(title ’All DB2 Metrics’     probematch counter(’DSN*’))
add generic metric CICS(title ’All CICS Metrics’   probematch counter(’DFH*’))
add generic metric MQ(title ’All MQ Metrics’       probematch counter(’CSQ*’))
add generic metric TCPIP(title ’All TCPIP Metrics’ probematch counter(’TCPIP*’))
add generic metric VOLS(title ’All VOLX* Volumes’  probematch counter(’device:VOLX*’)
add metric cpu.RMF0.prsm_cp_util_percent(title ’prsm_cp_util_percent’ counter(’cpu.RM
add metric cpu.RMF0.mvs_srm_cpu_util_percent(title ’mvs_srm_cpu_util_percent’ counter
add metric cpu.RMF0.zaap_util_percent(title ’zaap_util_percent’ counter(’cpu.RMF0.zaa
add metric cpu.RMF0.ziip_util_percent(title ’ziip_util_percent’ counter(’cpu.RMF0.zii
add metric spag.RMF0.page_ins_per_second(title ’page_ins_per_second’ counter(’spag.RM
set refclock server 81.137.201.165:60001
```

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.