cmlxdynp: Dynatrace Performance Metric Probe

CML00123-01

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

March 28, 2024

Contents

1 Introduction

This document describes how to use `cmlxdynp` which is a software utility to fetch Dynatrace performance data using the Dynatrace/RESTFul API via HTTPS. 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 Dynatrace platform and send the metrics to `Serfboard` in the correct form `cmlxdynp` needs to be configured. This is done through its command interface.

Dynatrace itself collects metrics from various machines, appliances and servers, and stores them, after aggregation, for extract.

`cmlxdynp` connects and extracts the performance metrics from the Dynatrace 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

`cmlxdynp` 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 Dynatrace probe V1.0: build 2024-02-29-12.36.24
[./cmlxdynp] $Id: cmlxdynp_help.tex,v 1.1 2024/03/28 18:57:09 hayward Exp $
Copyright (c) 2022 by Code Magus Limited. All rights reserved.
 [Contact: stephen@codemagus.com].
Usage: cmlxdynp [OPTION...]
  -p, --port={60057|<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 `cmlxdynp`, If not specified it will default to 60057.
- '`-c|--command`' Specifies a command to passed to the command interface.
- '`-n|--name`' Specifies a name for this instance of the probe. The Dynatrace probe connects to the Dynatrace server to extract the metrics for a machine. This means there is one probe instance per set of metrics. Often all these instances are run on a single machine and when there is more than one instance, name would usually be specified as the name of the target machine this instance is extracting metrics for.
- '`-v|--verbose`' When specified, `cmlxdynp` operates in a verbose manner.
- '`-t|--trace`' When specified, `cmlxdynp` writes all activity to `stdout`.

3 Invocation

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

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

4 Command Interface

`cmlxdynp` 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: 1.1 $'
```

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

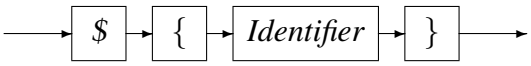
- possibly contains a radix character (decimal point '.').
- is optionally followed by a decimal exponent; consisting of an 'E' or 'e' followed by an optional plus or minus sign followed by a nonempty sequence of decimal digits that indicates multiplication by a power of 10.

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.

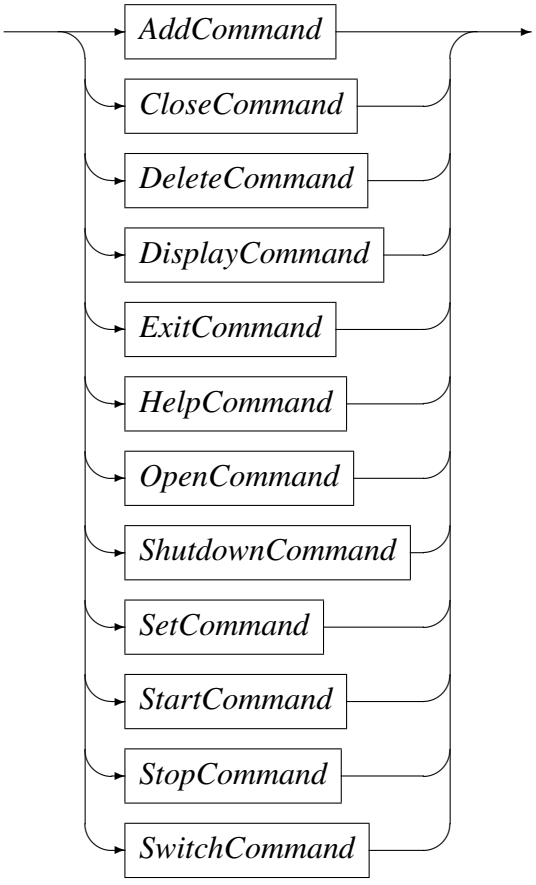
6 Syntax and Semantics

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 21 for an example of a `cmlxdynp` command file.

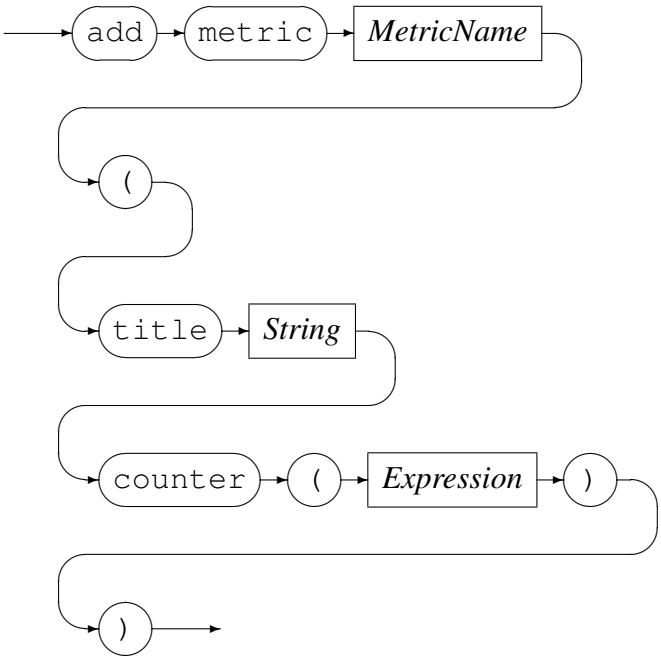
command



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 Dynatrace 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 Dynatrace performance counters.

Example

Add the `Serfboard` metric `cpu.cpu.load_avg1min` derived from the raw Dynatrace performance counters (`cpu.load`:

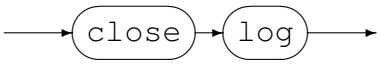
```
Dynatrace example> add metric cpu.cpu.load_avg1min \
                    (title 'CPU Load' counter(cpu.load))

Dynatrace example> Added metric cpu.cpu.load_avg1min
```

6.2 Close Command

This command is used to either:

- Close an opened recording log:

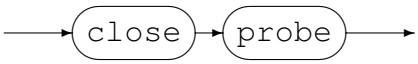


Example:

Close the previous opened log:

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

- Close the active probe:



Example:

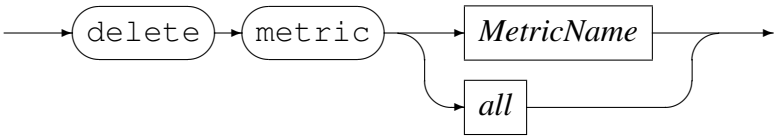
Close an opened probe:

```
Dynatrace example> close probe
Dynatrace example>
```

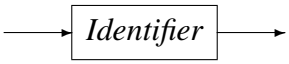
6.3 Delete Command

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:

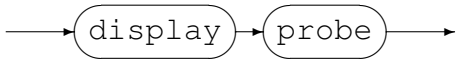
Delete `Serfboard` metric `cpu.cpu.load_avg1min`:

```
Dynatrace example> delete metric cpu.cpu.load_avg1min
Error: Probe is active - metric maintenance suspended!
Dynatrace example> stop probe
Probing stopped
Dynatrace example> delete metric cpu.cpu.load_avg1min
Metric cpu.cpu.load_avg1min deleted
```

6.4 Display Command

This command is used to display various configuration settings:

- Probe status:



Example:

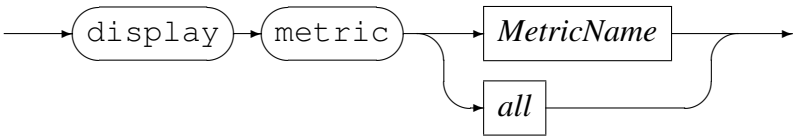
```
Dynatrace example> display probe
   CMLXDYNP Code Magus Limited Dynatrace probe V1.0
   Build    2023-09-28-12.25.26
   CMLXPRB  V2.1
   Probe    level 2
   Required parameters: URL OSType Host
   Optional parameters: APIPath MetricListFunction EntityListFunction MetricQuer
   Status:                Initialised
   Title:                 "Dynatrace example"
   Group:                 cmlxdynp
   Parameters:            ''
   Polling Interval:      30
   Polls per send:        1
   Housekeeping Interval: 600
   Server:                Not specified
   Cache size:            1.000G
   Cache:                 No Cache
  Temp Directory:         /tmp
   Configuration:         cmlxdynp_configuration.cmd
Dynatrace example>
```

- Available raw Dynatrace performance counters for probing:


```
Dynatrace example> display available counters
cpu.idle                        cpu.idle
cpu.iowait                      cpu.iowait
cpu.load                        cpu.load
.
.
.
disk.avail._                    disk.avail./
disk.avail._boot_efi            disk.avail./boot/efi
disk.bytesRead._                disk.bytesRead./
.
.
.
mem.avail.bytes                 mem.avail.bytes
mem.avail.pct                   mem.avail.pct
mem.avail.pfps                  mem.avail.pfps
.
.
.
net.bytesRx                     net.bytesRx
```

```
net.bytesTx                    net.bytesTx
net.nic.bytesRx.eth0           net.nic.bytesRx.eth0
net.nic.bytesTx.eth0           net.nic.bytesTx.eth0
.
.
.
```

- Display metrics:



MetricName

Example:

Display all the metrics that have been defined:

```
Dynatrace example> display metric all
metric cpu.cpu.load_avg_1min
   (
   title "CPU Load"
   counter(cpu.load))
   )
metric cpu.cpu.load_avg_5min
   (
   title "CPU 5M Load"
   counter(cpu.load5m))
   )
.
.
.
metric disk.bytesRead_root
   (
   title "Disk bytes Read /"
   counter(disk.bytesRead._))
   )
.
.
.
metric mem.mem.total
   (
   title "Total Memory"
   counter(mem.total))
   )
.
.
.
metric net.bytesRx
```

```
    (
    title "Network Bytes Received"
    counter(net.bytesRx))
    )
metric net.bytesTx
    (
    title "Network Bytes Sent"
    counter(net.bytesTx))
    )
.
.
.
```

6.5 Exit Command

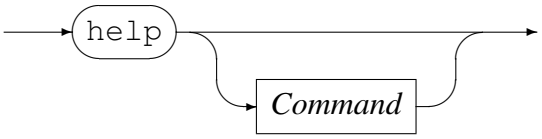
This command terminates an interactive session to the command interface of `cmlxdynp` and disconnects the client from the TCP/IP port.

ExitCommand

6.6 Help Command

Help on `cmlxdynp` commands.

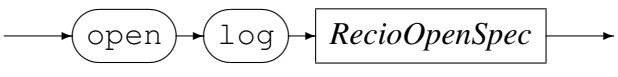
HelpCommand



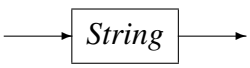
6.7 Open Command

This command is used to either

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



RecioOpenSpec



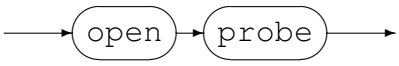
RecioOpenSpec is a `recio` [1] open specification string.

Example:

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

```
Dynatrace 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:

```
Dynatrace example> open probe
CML Dynatrace example initialised
```

6.8 Shutdown Command

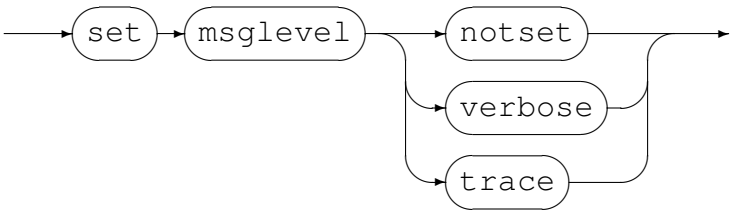
This command terminates `cmlxdynp`.

ShutdownCommand

6.9 Set Command

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



,APIPath=”api/v2”,MetricQueryParms=”metricSelector=For the Dynatrace `cmlxdynp` the following parameters can be set:

- Required Parameters
  - URL
    This sets the URL for the Dynatrace Server instance. It should be of the form `https://dynatrace.host.com`
  - OSType
    This tells the probe that the metrics it will extract pertain a specific type of host. The options for this are LINUX, WINDOWS, AIX or ZOS.
  - Host
    This is the host name as Dynatrace knows it and the host (machine) for which this instance will extract the metrics for.
- Optional Parameters
  There should be no need to change the optional parameters as the defaults are set for normal operation. A good understanding of Dynatrace operation and API is needed in order to use anything other than the defaults.
  - APIPath
    This gets appended to the `URL`. It defaults to

    ```
    e/b1f2a563-4555-4443-963c-43092d459063/api/v2
    ```
  - MetricListFunction
    This will name the the Dynatrace metric list function. It is added to the `URL` and `APIPath` when obtaining a list of Dynatrace metrics. The default is `metrics`.
  - EntityListFunction
    This will name the the Dynatrace entity list function. It is added to the `URL` and `APIPath` when obtaining a list of Dynatrace entities. The default is `entites`.
  - MetricQueryFunction
    This will name the the Dynatrace metric query function. It is added to the `URL` and `APIPath` when querying metric values. The default is `metrics/query`.
  - MetricListParms
    This will name the the Dynatrace metric list parameters. It is added to the `URL`, `APIPath` and `MetricListFunction` when obtaining a list of Dynatrace metrics. The default is

    ```
    pageSize=600&metricSelector=builtin:host.*&fields=displayName,
    description,unit,tags,entityType,metricSelector
    ```
  - EntityListParms
    This will name the the Dynatrace entity list parameters. It is added to the `URL`, `APIPath` and `EntityListFunction` when obtaining a list of Dynatrace entities. The default is

```
pageSize=4000&entitySelector=type("HOST"),
entityName.equals(%s)&fields=+properties.osType
```

The %s is replaced with the HostId of the machine the instance is extracting the metrics for.

* MetricQueryParms
  This will name the the Dynatrace metric query parameters. It is added to the `URL`, `APIPath` and `MetricQueryFunction` when querying metric values. The default is

  ```
  metricSelector=%s&resolution=1m&from=now-4m/m&entitySelector=entityId("%
  ```

  The first %s will be used to substitute in a comma separated list of metricIDs (up to 10 max) for which data points are required and the second %s will be used as the hostId of the entity (machine) that the data points are required for.

  For example, to extract the cpu type metrics the metricSelector would be set to this:

  ```
  builtin:host.cpu.(entc,entConfig,gcpu.usage,idle,iowait,load,load15m,loa
  msu.avg,msu.capacity,other,physc,steal,system,usage,user,ziip.eligible)
  ```

  and similarly for storage and network metrics.

* AuthorisationToken
  This sets the `Authorization HTTP` header for correct access to the Dynatrace server. There is a default, but as it is the access token it is not shared here. It would only need to be overridden on advice by a Dynatrace administrator.

Example:

Reset the level to not produce diagnostic messages:

```
Dynatrace example> set msglevel notset
Dynatrace example>
```

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

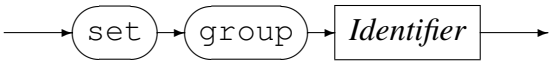
Example:

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

```
Dynatrace Probe> set title "Dynatrace example"
display probe
   CMLXDYNP Code Magus Limited Dynatrace probe V1.0
   Build    2024-03-28-15.17.50
   CMLXPRB  V2.1
   Probe    level 2
```

```
   Required parameters: URL OSType Host
   Optional parameters: APIPath MetricListFunction EntityListFunction MetricQuer
   Status:                Initialised
   Title:                 "Dynatrace example"
   Group:                 cmlxdynp
   Parameters:            ''
   Polling Interval:      30
   Polls per send:        1
   Housekeeping Interval: 600
   Server:                Not specified
   Cache size:            1.000G
   Cache:                 No Cache
  Temp Directory:         /tmp
   Configuration:         cmlxdynp_configuration.cmd
Dynatrace example>
```

- Set the `Serfboard` group name:

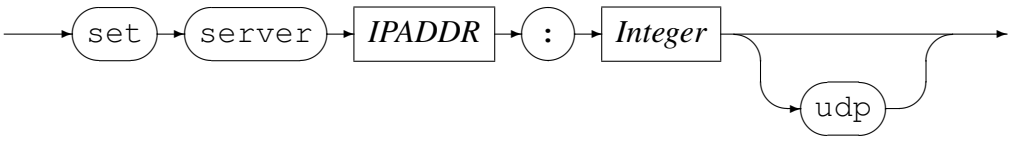


Example:

Set the `Serfboard` group name to `example`:

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

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



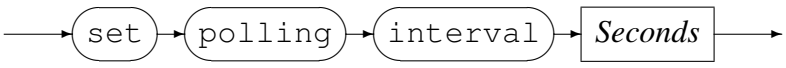
IPAddress can be specified as a host name or by using the Internet notation of dots and numbers. The default connection is TCP/IP, but if *udp* is specified, UDP will be used, a connectionless transport without guarantee of delivery.

Example:

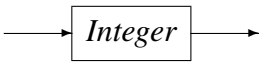
Set the host address of the `Serfboard` server to `www.codemagus.com`, listening on port `41000` and use UDP :

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

- Set the polling interval:



Seconds



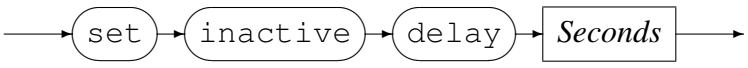
The polling interval is specified in seconds and is the frequency at which the Dynatrace counters will be polled in order to extract metric data. The frequency must be greater than zero and less than 100.

Example:

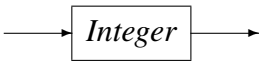
Set the polling interval to one minute:

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

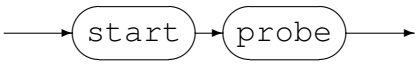
Set the inactive delay to one hour:

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

6.10 Start Command

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

StartCommand



Examples:

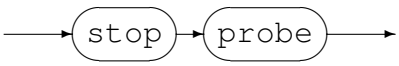
Start probing:

```
Dynatrace 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`; `cmlxdynp` reverts back to the idle state.

StopCommand



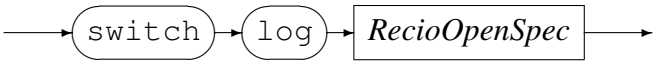
Examples:

Stop probing:

```
Dynatrace 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.

```
Dynatrace 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

The installation of cmlxdynp is quite straight forward and performed by extracting a TARBALL install image on a Linux machine. The Dynatrace probe is usually installed under any user.

To install the cmlxdynp 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_CMLXDYNP___Dccyymmdd_Thhmmss.tgz
   ```

   Where `ccmmyydd` 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:

   ```
   nohup ./bin/cmlxdynp.sh &
   ```

   This will start the probe in the background.

A Example Command File

```
-- Generated by : create_cmlxdynp.sh
--   Mon  9 Oct 12:46:13 SAST 2023
--
--
close probe
set title "SERVER1"
set group SERVER1
set refclock server 10.58.35.140:60001
set server 10.58.35.140:41212 UDP
set polling interval 60
set parameter(Host='d_server1',OSType="LINUX",\
URL="https://metrics.dynatrace.com",\
AuthorisationToken="Authorization: Api-Token xxxxxxxxx"\
)
-- The set parameter command automatically calls open probe.
-- So no need to redo it as it always just returns and error
-- open probe
--
delete metric all
--
add metric cpu.cpu.load_avg_1min (title 'CPU Load' counter(cpu.load))
add metric cpu.cpu.load_avg_5min (title 'CPU 5M Load' counter(cpu.load5m))
add metric cpu.cpu.load_avg_15min (title 'CPU 15M Load' counter(cpu.load15m))
add metric cpu.cpu.stolen (title 'CPU Stolen' counter(cpu.steal))
add metric cpu.cpu.system (title 'CPU System' counter(cpu.system))
add metric cpu.cpu.user (title 'CPU User' counter(cpu.user))
--
add metric disk.bytesRead_root (title 'Disk bytes Read /' counter(disk.bytesRead._))
add metric disk.bytesWritten_root (title 'Disk bytes Written /' counter(disk.bytesWri
add metric disk.bytesRead_var (title 'Disk bytes Read /var' counter(disk.bytesRead._v
add metric disk.bytesWritten_var (title 'Disk bytes Written /var' counter(disk.bytesW
add metric disk.bytesRead_home (title 'Disk bytes Read /home' counter(disk.bytesRead.
add metric disk.bytesWritten_home (title 'Disk bytes Written /home' counter(disk.byte
add metric disk.bytesRead_usr (title 'Disk bytes Read /usr' counter(disk.bytesRead._u
add metric disk.bytesWritten_usr (title 'Disk bytes Written /usr' counter(disk.bytesW
add metric disk.bytesRead_opt_app (title 'Disk bytes Read /opt/app' counter(disk.byte
add metric disk.bytesWritten_opt_app (title 'Disk bytes Written /opt/app' counter(dis
--
add metric mem.mem.total (title "Total Memory" counter(mem.total))
add metric mem.mem.free(title "Free memory" counter(mem.avail.bytes))
add metric mem.used(title 'Memory Used' counter(mem.used))
add metric mem.swap.total(title 'Memory Swap Total' counter(mem.swap.total))
add metric mem.swap.used(title 'Memory Swap Used' counter(mem.swap.used))
add metric mem.swap.free(title 'Memory Swap Free' counter(mem.swap.avail))
--
add metric net.bytesRx(title 'Network Bytes Received' counter(net.bytesRx))
add metric net.bytesTx(title 'Network Bytes Sent' counter(net.bytesTx))
--
```

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.