



---

cmlxfeed: Metrics Feed Library API Reference  
Version 1

CML00026-01

---

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>API Call Reference</b>	<b>3</b>
2.1	Introduction . . . . .	3
2.2	Functions . . . . .	3
2.2.1	Function <code>cmlxfeed_error()</code> . . . . .	3
2.2.2	Function <code>cmlxfeed_open()</code> . . . . .	3
2.2.3	Function <code>cmlxfeed_send()</code> . . . . .	4
2.2.4	Function <code>cmlxfeed_close()</code> . . . . .	4
2.3	Code Examples . . . . .	4
<b>3</b>	<b>API Header file</b>	<b>6</b>

# 1 Introduction

This manual describes the Code Magus Limited `cmlxfeed` library, which provides an application programming interface (API) for application programs that feed metrics to a Serfboard server, primarily for use in displaying a real time dashboard and/or stored for post processing analysis.

For Serfboard documentation please refer to the following manuals:

- Serfboard User Guide Version 1 [\[4\]](#)
- Serfboard Configuration Guide and Reference Version 1 [\[1\]](#)
- Serfboard Instruments Guide and Reference Version 1 [\[2\]](#)
- Serfboard Installation Guide and Reference Version 1 [\[3\]](#)

## 2 API Call Reference

### 2.1 Introduction

The `cmlxfeed` library supplies an API for sending metrics to a Serfboard server. The API is described by the header file `cmlxfeed.h` (see section 3 on page 6) and implemented by statically linking `libcmlxfeed.a` on Linux and Unix and `cmlxfeed.lib` on Windows into the calling application.

Metrics may be sent using either:

- Transmission Control Protocol - TCP/IP:

TCP/IP provides for a reliable, ordered delivery of messages from the application to the server. This is the default protocol used by the API.

- User Datagram Protocol - UDP:

UDP is a connectionless transport without guarantee of delivery. In other words it provides an unreliable service but avoids the overhead required for the transmission integrity by TCP/IP.

See `cmlxfeed_open()` in section 2.2.2 on page 3 for information on how to select either TCP/IP or UDP.

### 2.2 Functions

#### 2.2.1 Function `cmlxfeed_error()`

```
char *cmlxfeed_error(cmlxfeed_t * instance);
```

Function `cmlxfeed_error()` returns a message relating to the last error returned by one of the other functions of the `cmlxfeed` library for the given `instance`. If the `instance` is supplied as `NULL`, then the last error unrelated to an instance is returned (for example, in the situation that `cmlxfeed_open()` failed).

#### 2.2.2 Function `cmlxfeed_open()`

```
cmlxfeed_t *cmlxfeed_open(char *ip_address, char *port, int is_udp);
```

Function `cmlxfeed_open()` opens a connection to a Serfboard server, using the supplied `host` and `port` parameters. The connection is UDP if the parameter `is_udp` is set to '1', the default is TCP/IP.

If the open succeeds then a `cmlxfeed_t` structure is returned, which is to be used on all other function calls to the `cmlxfeed` library for this instance.

If an error occurs then NULL is returned and a call to `cmlxfeed_error()` will return a message describing the error.

### 2.2.3 Function `cmlxfeed_send()`

```
int cmlxfeed_send(cmlxfeed_t *instance, char *buffer);
```

Function `cmlxfeed_send()` will send the `buffer` to the Serfboard server described by `instance`. The buffer is expected to be a NULL terminated string.

The function returns zero upon success.

If an error occurs then -1 is returned and a call to `cmlxfeed_error()` will return a message describing the error.

### 2.2.4 Function `cmlxfeed_close()`

```
int cmlxfeed_close(cmlxfeed_t *instance);
```

Function `cmlxfeed_close()` closes the connection to a Serfboard server as described by the supplied `instance`.

The function returns zero upon success.

If an error occurs then -1 is returned and a call to `cmlxfeed_error()` will return a message describing the error.

## 2.3 Code Examples

The following code examples are extracted from a Serfboard feeder program:

- Connect to a Serfboard server in order to send metrics via TCP/IP:
  - Open the connection to the Serfboard server:

```
cmlxfeed = cmlxfeed_open(feed_host, feed_port, 0);
if (cmlxfeed)
    serfboard_is_connected = 1;
else
{
    rprintf("Error: %s\n", cmlxfeed_error(NULL));
    serfboard_is_connected = 0;
}
```

- Sending a Serfboard metric:

```
if (cmlxfeed_send(cmlxfeed, buf) < 0)
{
    rprintf("Error %s:%d - %s\n", feed_host, feed_port,
           cmlxfeed_error(cmlxfeed));
    return -1;
}
```

- Connect to a Serfboard server in order to send metrics via UDP:

The only difference from the preceding example is during the open function call where UDP must be requested:

```
cmlxfeed = cmlxfeed_open(feed_host, feed_port, 1);
```

Note, however, that when using UDP, because it provides an unreliable service, there is no error response if the message is not delivered; for example if the server is not active or a transmission error occurs along the path to the server after the metric has been sent.

### 3 API Header file

```
#ifndef CMLXFEED_H
#define CMLXFEED_H

/* File: cmlxfeed.h
 *
 * This header file describes the Code Magus Limited cmlxfeed, which provides
 * an interface for application programs cmlxfeeding metrics to a serfboard
 * server.for use in displaying a real time dashboard and stored for post
 * processing analysis.
 *
 * Author: JV.
 *
 * Copyright (c) 2008 Code Magus Limited. All rights reserved.
 */

/*
 * $Author: janvlok $
 * $Date: 2010/12/07 10:18:26 $
 * $Id: cmlxfeed.h,v 1.3 2010/12/07 10:18:26 janvlok Exp $
 * $Revision: 1.3 $
 * $State: Exp $
 *
 * $Log: cmlxfeed.h,v $
 * Revision 1.3 2010/12/07 10:18:26 janvlok
 * Rewritten
 */

static char cvs_cmlxfeed_h[] =
    "$Id: cmlxfeed.h,v 1.3 2010/12/07 10:18:26 janvlok Exp $";

/*
 * Constants and options:
 */

#define ERR_BUFF_SIZE 256

/*
 * Exposed types and structures:
 */

/* cmlxfeed_t describes a TCP/IP connection instance to a serfboard
 * .server. It is created by the cmlxfeed_open() function and must be passed
 * when using the cmlxfeed_send() function.
 */

typedef struct cmlxfeed cmlxfeed_t;
struct cmlxfeed
{
```

### 3 API HEADER FILE

---

```
char host_name[64];          /* host name or IP address in dotted quad */
int port;                   /* port number */
char errmsg[ERR_BUFF_SIZE]; /* error message */
int socktype;               /* SOCK_STREAM | SOCK_DGRAM */
int sockfd;                 /* file descriptor used by the socket */
struct sockaddr *saddr;     /* socket address */
int saddrlen;               /* socket address length */
};

/* Function cmlxfeed_error() returns a message relating to the last error
 * returned by one of the other functions of the cmlxfeed library. If the
 * instance is supplied as NULL, then the last error returned unrelated to
 * a instance is returned (for example, in the situation that an cmlxfeed_open
 * failed , then the error message is placed in a global
 * structure. Otherwise for instance related errors the error message is
 * taken from the instance structure.
 */

char *cmlxfeed_error(cmlxfeed_t *instance);

/* Function cmlxfeed_open() opens a TCP/IP connection to a serfboard.
 * server, using the supplied host and port parameters. If the open succeeds
 * then a cmlxfeed_t structure is. returned which is to be used on
 * the cmlxfeed_send() function. If an error occurs then NULL is returned
 * and cmlxfeed_error() describes the error.
 */

cmlxfeed_t *cmlxfeed_open(char *ip_address,int port,int is_udp);

/* Function cmlxfeed_send() send the buffer to the supplied instance.
 * The buffer is expected to be a string, NULL terminated.
 * Returns zero upon success, else -1 is returned. and cmlxfeed_error()
 * describes the error.
 */

int cmlxfeed_send(cmlxfeed_t *instance,char *buffer);

/* Function cmlxfeed_close() closes the TCP/IP connection to a serfboard.
 * server as described by the supplied instance.
 * Returns zero upon success, else -1 is returned. and cmlxfeed_error()
 * describes the error.
 */

int cmlxfeed_close(cmlxfeed_t *instance);

#endif /* CMLXFEED_H */
```



## References

- [1] Serfboard Configuration Guide and Reference Version 1. CML Document CML00023-01, Code Magus Limited, July 2008. [PDF](#).
- [2] Serfboard Instruments Guide and Reference Version 1. CML Document CML00024-01, Code Magus Limited, July 2008. [PDF](#).
- [3] Serfboard Installation Guide and Reference Version 1. CML Document CML00025-01, Code Magus Limited, July 2008. [PDF](#).
- [4] Serfboard User Guide Version 1. CML Document CML00027-01, Code Magus Limited, July 2008. [PDF](#).