

**CLD**

0.1git

Generated by Doxygen 1.5.9

Tue Aug 11 14:24:16 2009



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	cld_dirent_cur Struct Reference . . . . .	5
3.1.1	Field Documentation . . . . .	5
3.1.1.1	p . . . . .	5
3.1.1.2	tmp_len . . . . .	5
3.2	cld_msg_close Struct Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Field Documentation . . . . .	6
3.2.2.1	fh . . . . .	6
3.2.2.2	hdr . . . . .	6
3.3	cld_msg_data Struct Reference . . . . .	7
3.3.1	Detailed Description . . . . .	7
3.3.2	Field Documentation . . . . .	7
3.3.2.1	hdr . . . . .	7
3.3.2.2	seg . . . . .	7
3.3.2.3	seg_len . . . . .	7
3.3.2.4	strid . . . . .	7
3.4	cld_msg_del Struct Reference . . . . .	8
3.4.1	Detailed Description . . . . .	8
3.4.2	Field Documentation . . . . .	8
3.4.2.1	hdr . . . . .	8
3.4.2.2	name_len . . . . .	8
3.5	cld_msg_event Struct Reference . . . . .	9

3.5.1	Detailed Description	9
3.5.2	Field Documentation	9
3.5.2.1	events	9
3.5.2.2	fh	9
3.5.2.3	hdr	9
3.6	cld_msg_get Struct Reference	10
3.6.1	Detailed Description	10
3.6.2	Field Documentation	10
3.6.2.1	fh	10
3.6.2.2	hdr	10
3.7	cld_msg_get_resp Struct Reference	11
3.7.1	Detailed Description	11
3.7.2	Field Documentation	11
3.7.2.1	flags	11
3.7.2.2	ino_len	11
3.7.2.3	inum	12
3.7.2.4	resp	12
3.7.2.5	size	12
3.7.2.6	strid	12
3.7.2.7	time_create	12
3.7.2.8	time_modify	12
3.7.2.9	version	12
3.8	cld_msg_hdr Struct Reference	13
3.8.1	Detailed Description	13
3.8.2	Field Documentation	13
3.8.2.1	magic	13
3.8.2.2	op	13
3.8.2.3	res1	13
3.8.2.4	xid	13
3.9	cld_msg_lock Struct Reference	14
3.9.1	Detailed Description	14
3.9.2	Field Documentation	14
3.9.2.1	fh	14
3.9.2.2	flags	14
3.9.2.3	hdr	14
3.10	cld_msg_open Struct Reference	15

3.10.1 Detailed Description . . . . .	15
3.10.2 Field Documentation . . . . .	15
3.10.2.1 events . . . . .	15
3.10.2.2 hdr . . . . .	15
3.10.2.3 mode . . . . .	15
3.10.2.4 name_len . . . . .	15
3.11 cld_msg_open_resp Struct Reference . . . . .	16
3.11.1 Detailed Description . . . . .	16
3.11.2 Field Documentation . . . . .	16
3.11.2.1 fh . . . . .	16
3.11.2.2 resp . . . . .	16
3.12 cld_msg_put Struct Reference . . . . .	17
3.12.1 Detailed Description . . . . .	17
3.12.2 Field Documentation . . . . .	17
3.12.2.1 data_size . . . . .	17
3.12.2.2 fh . . . . .	17
3.12.2.3 hdr . . . . .	17
3.12.2.4 strid . . . . .	17
3.13 cld_msg_resp Struct Reference . . . . .	18
3.13.1 Detailed Description . . . . .	18
3.13.2 Field Documentation . . . . .	18
3.13.2.1 code . . . . .	18
3.13.2.2 hdr . . . . .	18
3.13.2.3 rsv . . . . .	18
3.13.2.4 xid_in . . . . .	18
3.14 cld_msg_unlock Struct Reference . . . . .	19
3.14.1 Detailed Description . . . . .	19
3.14.2 Field Documentation . . . . .	19
3.14.2.1 fh . . . . .	19
3.14.2.2 hdr . . . . .	19
3.15 cld_packet Struct Reference . . . . .	20
3.15.1 Detailed Description . . . . .	20
3.15.2 Field Documentation . . . . .	20
3.15.2.1 magic . . . . .	20
3.15.2.2 n_msg . . . . .	20
3.15.2.3 res . . . . .	20

3.15.2.4	seqid	20
3.15.2.5	sid	20
3.15.2.6	user	21
3.16	cldc_call_opts Struct Reference	22
3.16.1	Detailed Description	22
3.16.2	Field Documentation	22
3.16.2.1	buf	22
3.16.2.2	cb	22
3.16.2.3	get	22
3.16.2.4	inode_name	22
3.16.2.5	op	22
3.16.2.6	private	22
3.16.2.7	resp	22
3.16.2.8	size	22
3.16.2.9	u	22
3.17	cldc_fh Struct Reference	23
3.17.1	Detailed Description	23
3.17.2	Field Documentation	23
3.17.2.1	fh_le	23
3.17.2.2	sess	23
3.17.2.3	valid	23
3.18	cldc_host Struct Reference	24
3.18.1	Detailed Description	24
3.18.2	Field Documentation	24
3.18.2.1	host	24
3.18.2.2	known	24
3.18.2.3	port	24
3.18.2.4	prio	24
3.18.2.5	weight	24
3.19	cldc_msg Struct Reference	25
3.19.1	Detailed Description	25
3.19.2	Field Documentation	26
3.19.2.1	cb	26
3.19.2.2	cb_private	26
3.19.2.3	copts	26
3.19.2.4	data	26

3.19.2.5	<a href="#">data_len</a>	26
3.19.2.6	<a href="#">done</a>	26
3.19.2.7	<a href="#">expire_time</a>	26
3.19.2.8	<a href="#">pkt</a>	26
3.19.2.9	<a href="#">retries</a>	26
3.19.2.10	<a href="#">seqid</a>	26
3.19.2.11	<a href="#">sess</a>	26
3.19.2.12	<a href="#">xid</a>	26
3.20	<a href="#">cldc_ops Struct Reference</a>	27
3.20.1	<a href="#">Detailed Description</a>	27
3.20.2	<a href="#">Field Documentation</a>	27
3.20.2.1	<a href="#">event</a>	27
3.20.2.2	<a href="#">pkt_send</a>	27
3.20.2.3	<a href="#">printf</a>	27
3.20.2.4	<a href="#">timer_ctl</a>	27
3.21	<a href="#">cldc_session Struct Reference</a>	28
3.21.1	<a href="#">Detailed Description</a>	28
3.21.2	<a href="#">Field Documentation</a>	29
3.21.2.1	<a href="#">act_log</a>	29
3.21.2.2	<a href="#">addr</a>	29
3.21.2.3	<a href="#">addr_len</a>	29
3.21.2.4	<a href="#">confirmed</a>	29
3.21.2.5	<a href="#">expire_time</a>	29
3.21.2.6	<a href="#">expired</a>	29
3.21.2.7	<a href="#">fh</a>	29
3.21.2.8	<a href="#">msg_scan_time</a>	29
3.21.2.9	<a href="#">next_seqid_in</a>	29
3.21.2.10	<a href="#">next_seqid_in_tr</a>	29
3.21.2.11	<a href="#">next_seqid_out</a>	29
3.21.2.12	<a href="#">ops</a>	29
3.21.2.13	<a href="#">out_msg</a>	29
3.21.2.14	<a href="#">private</a>	29
3.21.2.15	<a href="#">secret_key</a>	29
3.21.2.16	<a href="#">sid</a>	29
3.21.2.17	<a href="#">streams</a>	29
3.21.2.18	<a href="#">user</a>	29

3.21.2.19	verbose	29
3.22	cldc_stream Struct Reference	30
3.22.1	Detailed Description	30
3.22.2	Field Documentation	30
3.22.2.1	buf	30
3.22.2.2	bufp	30
3.22.2.3	copts	30
3.22.2.4	next_seg	31
3.22.2.5	size	31
3.22.2.6	size_left	31
3.22.2.7	strid_le	31
3.23	cldc_udp Struct Reference	32
3.23.1	Detailed Description	32
3.23.2	Field Documentation	32
3.23.2.1	addr	32
3.23.2.2	addr_len	32
3.23.2.3	cb	32
3.23.2.4	cb_private	32
3.23.2.5	fd	32
3.23.2.6	sess	32
3.23.2.7	timer_ev	32
<b>4</b>	<b>File Documentation</b>	<b>33</b>
4.1	include/cld_msg.h File Reference	33
4.1.1	Define Documentation	35
4.1.1.1	CLD_ALIGN8	35
4.1.1.2	CLD_MSG_MAGIC	35
4.1.1.3	CLD_PKT_MAGIC	35
4.1.1.4	SIDARG	35
4.1.1.5	SIDFMT	35
4.1.2	Enumeration Type Documentation	35
4.1.2.1	"@0	35
4.1.2.2	cld_events	36
4.1.2.3	cld_lock_flags	36
4.1.2.4	cld_msg_ops	36
4.1.2.5	cld_open_modes	37
4.1.2.6	cle_err_codes	37



4.1.3	Function Documentation	37
4.1.3.1	__cld_rand64	37
4.1.3.2	cld_sid2llu	37
4.2	include/cldc.h File Reference	38
4.2.1	Function Documentation	40
4.2.1.1	cldc_close	40
4.2.1.2	cldc_del	40
4.2.1.3	cldc_dirent_count	40
4.2.1.4	cldc_dirent_cur_fini	40
4.2.1.5	cldc_dirent_cur_init	40
4.2.1.6	cldc_dirent_first	40
4.2.1.7	cldc_dirent_name	40
4.2.1.8	cldc_dirent_next	40
4.2.1.9	cldc_end_sess	40
4.2.1.10	cldc_get	40
4.2.1.11	cldc_getaddr	40
4.2.1.12	cldc_kill_sess	40
4.2.1.13	cldc_levent_timer	40
4.2.1.14	cldc_lock	40
4.2.1.15	cldc_new_sess	40
4.2.1.16	cldc_nop	40
4.2.1.17	cldc_open	40
4.2.1.18	cldc_put	40
4.2.1.19	cldc_receive_pkt	40
4.2.1.20	cldc_saveaddr	41
4.2.1.21	cldc_udp_free	41
4.2.1.22	cldc_udp_new	41
4.2.1.23	cldc_udp_pkt_send	41
4.2.1.24	cldc_udp_receive_pkt	41
4.2.1.25	cldc_unlock	41



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">cld_dirent_cur</a> . . . . .	5
<a href="#">cld_msg_close</a> (CLOSE message) . . . . .	6
<a href="#">cld_msg_data</a> (DATA message) . . . . .	7
<a href="#">cld_msg_del</a> (DEL message) . . . . .	8
<a href="#">cld_msg_event</a> (Server-to-client EVENT message) . . . . .	9
<a href="#">cld_msg_get</a> (GET message) . . . . .	10
<a href="#">cld_msg_get_resp</a> (GET message response) . . . . .	11
<a href="#">cld_msg_hdr</a> (Header for each message) . . . . .	13
<a href="#">cld_msg_lock</a> (LOCK message) . . . . .	14
<a href="#">cld_msg_open</a> (OPEN message) . . . . .	15
<a href="#">cld_msg_open_resp</a> (OPEN message response) . . . . .	16
<a href="#">cld_msg_put</a> (PUT message) . . . . .	17
<a href="#">cld_msg_resp</a> (Standard response for each message) . . . . .	18
<a href="#">cld_msg_unlock</a> (UNLOCK message) . . . . .	19
<a href="#">cld_packet</a> (Header for each packet) . . . . .	20
<a href="#">cldc_call_opts</a> (Per-operation application options) . . . . .	22
<a href="#">cldc_fh</a> (Open file handle associated with a session) . . . . .	23
<a href="#">cldc_host</a> (Information for a single CLD server host) . . . . .	24
<a href="#">cldc_msg</a> (Outgoing message, from client to server) . . . . .	25
<a href="#">cldc_ops</a> (Application-supplied facilities) . . . . .	27
<a href="#">cldc_session</a> (Single CLD client session) . . . . .	28
<a href="#">cldc_stream</a> (Internal per-data stream information) . . . . .	30
<a href="#">cldc_udp</a> (A UDP implementation of the CLD client protocol) . . . . .	32



# Chapter 2

# File Index

## 2.1 File List

Here is a list of all files with brief descriptions:

include/ <a href="#">cld_msg.h</a>	33
include/ <a href="#">cldc.h</a>	38



## Chapter 3

# Data Structure Documentation

### 3.1 cld\_dirent\_cur Struct Reference

```
#include <cldc.h>
```

#### Data Fields

- const void \* [p](#)
- size\_t [tmp\\_len](#)

#### 3.1.1 Field Documentation

**3.1.1.1** const void\* cld\_dirent\_cur::p

**3.1.1.2** size\_t cld\_dirent\_cur::tmp\_len

The documentation for this struct was generated from the following file:

- include/[cldc.h](#)

## 3.2 cld\_msg\_close Struct Reference

CLOSE message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- `uint64_t` `fh`  
*open file handle*

### 3.2.1 Detailed Description

CLOSE message.

### 3.2.2 Field Documentation

#### 3.2.2.1 `uint64_t cld_msg_close::fh`

open file handle

#### 3.2.2.2 `struct cld_msg_hdr cld_msg_close::hdr` [read]

The documentation for this struct was generated from the following file:

- `include/cld_msg.h`



## 3.3 cld\_msg\_data Struct Reference

DATA message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- uint64\_t `strid`  
*stream id*
- uint32\_t `seg`  
*segment number*
- uint32\_t `seg_len`  
*segment length*

### 3.3.1 Detailed Description

DATA message.

### 3.3.2 Field Documentation

**3.3.2.1** struct `cld_msg_hdr cld_msg_data::hdr` [read]

**3.3.2.2** uint32\_t `cld_msg_data::seg`

segment number

**3.3.2.3** uint32\_t `cld_msg_data::seg_len`

segment length

**3.3.2.4** uint64\_t `cld_msg_data::strid`

stream id

The documentation for this struct was generated from the following file:

- include/[cld\\_msg.h](#)

## 3.4 cld\_msg\_del Struct Reference

DEL message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- uint16\_t [name\\_len](#)  
*length of file name*

### 3.4.1 Detailed Description

DEL message.

### 3.4.2 Field Documentation

**3.4.2.1** struct `cld_msg_hdr cld_msg_del::hdr` `[read]`

**3.4.2.2** uint16\_t `cld_msg_del::name_len`

length of file name

The documentation for this struct was generated from the following file:

- include/[cld\\_msg.h](#)

## 3.5 cld\_msg\_event Struct Reference

Server-to-client EVENT message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- uint64\_t `fh`  
*open file handle*
- uint32\_t `events`  
*CE\_XXX.*

### 3.5.1 Detailed Description

Server-to-client EVENT message.

### 3.5.2 Field Documentation

#### 3.5.2.1 uint32\_t cld\_msg\_event::events

CE\_XXX.

#### 3.5.2.2 uint64\_t cld\_msg\_event::fh

open file handle

#### 3.5.2.3 struct cld\_msg\_hdr cld\_msg\_event::hdr [read]

The documentation for this struct was generated from the following file:

- include/[cld\\_msg.h](#)

## 3.6 cld\_msg\_get Struct Reference

GET message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- [uint64\\_t](#) `fh`  
*open file handle*

### 3.6.1 Detailed Description

GET message.

### 3.6.2 Field Documentation

#### 3.6.2.1 [uint64\\_t](#) `cld_msg_get::fh`

open file handle

#### 3.6.2.2 `struct cld_msg_hdr cld_msg_get::hdr` [read]

The documentation for this struct was generated from the following file:

- [include/cld\\_msg.h](#)

## 3.7 cld\_msg\_get\_resp Struct Reference

GET message response.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_resp](#) `resp`
- uint64\_t `inum`  
*unique inode number*
- uint32\_t `ino_len`  
*inode name len*
- uint32\_t `size`  
*data size*
- uint64\_t `version`  
*inode version*
- uint64\_t `time_create`  
*creation time*
- uint64\_t `time_modify`  
*last modification time*
- uint32\_t `flags`  
*inode flags; CIFL\_XXX*
- uint64\_t `strid`  
*DATA stream id.*

### 3.7.1 Detailed Description

GET message response.

### 3.7.2 Field Documentation

#### 3.7.2.1 uint32\_t cld\_msg\_get\_resp::flags

inode flags; CIFL\_XXX

#### 3.7.2.2 uint32\_t cld\_msg\_get\_resp::ino\_len

inode name len

**3.7.2.3 uint64\_t cld\_msg\_get\_resp::inum**

unique inode number

**3.7.2.4 struct cld\_msg\_resp cld\_msg\_get\_resp::resp** [read]**3.7.2.5 uint32\_t cld\_msg\_get\_resp::size**

data size

**3.7.2.6 uint64\_t cld\_msg\_get\_resp::strid**

DATA stream id.

**3.7.2.7 uint64\_t cld\_msg\_get\_resp::time\_create**

creation time

**3.7.2.8 uint64\_t cld\_msg\_get\_resp::time\_modify**

last modification time

**3.7.2.9 uint64\_t cld\_msg\_get\_resp::version**

inode version

The documentation for this struct was generated from the following file:

- [include/cld\\_msg.h](#)

## 3.8 cld\_msg\_hdr Struct Reference

header for each message

```
#include <cld_msg.h>
```

### Data Fields

- uint8\_t [magic](#) [CLD\_MAGIC\_SZ]  
*magic number; constant*
- uint64\_t [xid](#)  
*opaque message id*
- uint8\_t [op](#)  
*operation code*
- uint8\_t [res1](#) [7]

### 3.8.1 Detailed Description

header for each message

### 3.8.2 Field Documentation

#### 3.8.2.1 uint8\_t cld\_msg\_hdr::magic[CLD\_MAGIC\_SZ]

magic number; constant

#### 3.8.2.2 uint8\_t cld\_msg\_hdr::op

operation code

#### 3.8.2.3 uint8\_t cld\_msg\_hdr::res1[7]

#### 3.8.2.4 uint64\_t cld\_msg\_hdr::xid

opaque message id

The documentation for this struct was generated from the following file:

- include/[cld\\_msg.h](#)

## 3.9 cld\_msg\_lock Struct Reference

LOCK message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- [uint64\\_t](#) `fh`  
*open file handle*
- [uint32\\_t](#) `flags`  
*CLF\_XXX.*

### 3.9.1 Detailed Description

LOCK message.

### 3.9.2 Field Documentation

#### 3.9.2.1 [uint64\\_t](#) `cld_msg_lock::fh`

open file handle

#### 3.9.2.2 [uint32\\_t](#) `cld_msg_lock::flags`

CLF\_XXX.

#### 3.9.2.3 [struct cld\\_msg\\_hdr](#) `cld_msg_lock::hdr` [read]

The documentation for this struct was generated from the following file:

- [include/cld\\_msg.h](#)



## 3.10 cld\_msg\_open Struct Reference

OPEN message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) **hdr**
- uint32\_t **mode**  
*open mode, COM\_xxx*
- uint32\_t **events**  
*events mask, CE\_xxx*
- uint16\_t **name\_len**  
*length of file name*

### 3.10.1 Detailed Description

OPEN message.

### 3.10.2 Field Documentation

#### 3.10.2.1 uint32\_t cld\_msg\_open::events

events mask, CE\_xxx

#### 3.10.2.2 struct cld\_msg\_hdr cld\_msg\_open::hdr [read]

#### 3.10.2.3 uint32\_t cld\_msg\_open::mode

open mode, COM\_xxx

#### 3.10.2.4 uint16\_t cld\_msg\_open::name\_len

length of file name

The documentation for this struct was generated from the following file:

- include/[cld\\_msg.h](#)

## 3.11 cld\_msg\_open\_resp Struct Reference

OPEN message response.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_resp](#) `resp`
- [uint64\\_t](#) `fh`  
*handle opened*

### 3.11.1 Detailed Description

OPEN message response.

### 3.11.2 Field Documentation

#### 3.11.2.1 [uint64\\_t](#) `cld_msg_open_resp::fh`

*handle opened*

#### 3.11.2.2 `struct cld_msg_resp cld_msg_open_resp::resp` [read]

The documentation for this struct was generated from the following file:

- [include/cld\\_msg.h](#)

## 3.12 cld\_msg\_put Struct Reference

PUT message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- [uint64\\_t](#) `fh`  
*open file handle*
- [uint64\\_t](#) `strid`  
*DATA stream id.*
- [uint32\\_t](#) `data_size`  
*total size of data*

### 3.12.1 Detailed Description

PUT message.

### 3.12.2 Field Documentation

#### 3.12.2.1 [uint32\\_t](#) `cld_msg_put::data_size`

total size of data

#### 3.12.2.2 [uint64\\_t](#) `cld_msg_put::fh`

open file handle

#### 3.12.2.3 [struct cld\\_msg\\_hdr](#) `cld_msg_put::hdr` [read]

#### 3.12.2.4 [uint64\\_t](#) `cld_msg_put::strid`

DATA stream id.

The documentation for this struct was generated from the following file:

- include/[cld\\_msg.h](#)

## 3.13 cld\_msg\_resp Struct Reference

standard response for each message

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) [hdr](#)
- uint32\_t [code](#)  
*error code, CLE\_XXX*
- uint32\_t [rsv](#)  
*reserved*
- uint64\_t [xid\\_in](#)  
*C->S xid.*

### 3.13.1 Detailed Description

standard response for each message

### 3.13.2 Field Documentation

#### 3.13.2.1 uint32\_t cld\_msg\_resp::code

error code, CLE\_XXX

#### 3.13.2.2 struct cld\_msg\_hdr cld\_msg\_resp::hdr [read]

#### 3.13.2.3 uint32\_t cld\_msg\_resp::rsv

reserved

#### 3.13.2.4 uint64\_t cld\_msg\_resp::xid\_in

C->S xid.

The documentation for this struct was generated from the following file:

- include/[cld\\_msg.h](#)

## 3.14 cld\_msg\_unlock Struct Reference

UNLOCK message.

```
#include <cld_msg.h>
```

### Data Fields

- struct [cld\\_msg\\_hdr](#) `hdr`
- `uint64_t` `fh`  
*open file handle*

### 3.14.1 Detailed Description

UNLOCK message.

### 3.14.2 Field Documentation

#### 3.14.2.1 `uint64_t` `cld_msg_unlock::fh`

open file handle

#### 3.14.2.2 `struct cld_msg_hdr` `cld_msg_unlock::hdr` [read]

The documentation for this struct was generated from the following file:

- `include/cld_msg.h`

## 3.15 cld\_packet Struct Reference

header for each packet

```
#include <cld_msg.h>
```

### Data Fields

- uint8\_t [magic](#) [CLD\_MAGIC\_SZ]  
*magic number; constant*
- uint64\_t [seqid](#)  
*sequence id*
- uint8\_t [sid](#) [CLD\_SID\_SZ]  
*client id*
- uint8\_t [n\\_msg](#)  
*num msgs in packet*
- uint8\_t [res](#) [7]
- char [user](#) [CLD\_MAX\_USERNAME]  
*authenticated user*

### 3.15.1 Detailed Description

header for each packet

### 3.15.2 Field Documentation

#### 3.15.2.1 uint8\_t cld\_packet::magic[CLD\_MAGIC\_SZ]

magic number; constant

#### 3.15.2.2 uint8\_t cld\_packet::n\_msg

num msgs in packet

#### 3.15.2.3 uint8\_t cld\_packet::res[7]

#### 3.15.2.4 uint64\_t cld\_packet::seqid

sequence id

#### 3.15.2.5 uint8\_t cld\_packet::sid[CLD\_SID\_SZ]

client id

#### 3.15.2.6 char cld\_packet::user[CLD\_MAX\_USERNAME]

authenticated user

The documentation for this struct was generated from the following file:

- [include/cld\\_msg.h](#)

## 3.16 cldc\_call\_opts Struct Reference

per-operation application options

```
#include <cldc.h>
```

### Data Fields

- `int(* cb)(struct cldc_call_opts *, enum cle_err_codes)`
- `void * private`
- `enum cld_msg_ops op`
- `union {`
  - `struct {`
    - `struct cld_msg_get_resp resp`
    - `char * buf`
    - `unsigned int size`
    - `char inode_name [CLD_INODE_NAME_MAX]`
  - `} get`
- `} u`

### 3.16.1 Detailed Description

per-operation application options

### 3.16.2 Field Documentation

**3.16.2.1** `char* cldc_call_opts::buf`

**3.16.2.2** `int(* cldc_call_opts::cb)(struct cldc_call_opts *, enum cle_err_codes)`

**3.16.2.3** `struct { ... } cldc_call_opts::get`

**3.16.2.4** `char cldc_call_opts::inode_name[CLD_INODE_NAME_MAX]`

**3.16.2.5** `enum cld_msg_ops cldc_call_opts::op`

**3.16.2.6** `void* cldc_call_opts::private`

**3.16.2.7** `struct cld_msg_get_resp cldc_call_opts::resp` [read]

**3.16.2.8** `unsigned int cldc_call_opts::size`

**3.16.2.9** `union { ... } cldc_call_opts::u`

The documentation for this struct was generated from the following file:

- `include/cldc.h`



## 3.17 cldc\_fh Struct Reference

an open file handle associated with a session

```
#include <cldc.h>
```

### Data Fields

- uint64\_t [fh\\_le](#)
- struct [cldc\\_session](#) \* [sess](#)
- bool [valid](#)

### 3.17.1 Detailed Description

an open file handle associated with a session

### 3.17.2 Field Documentation

**3.17.2.1**    `uint64_t cldc_fh::fh_le`

**3.17.2.2**    `struct cldc_session* cldc_fh::sess`    [`read`]

**3.17.2.3**    `bool cldc_fh::valid`

The documentation for this struct was generated from the following file:

- `include/cldc.h`

## 3.18 cldc\_host Struct Reference

Information for a single CLD server host.

```
#include <cldc.h>
```

### Data Fields

- int [known](#)
- unsigned int [prio](#)
- unsigned int [weight](#)
- char \* [host](#)
- unsigned short [port](#)

### 3.18.1 Detailed Description

Information for a single CLD server host.

### 3.18.2 Field Documentation

**3.18.2.1** char\* cldc\_host::host

**3.18.2.2** int cldc\_host::known

**3.18.2.3** unsigned short cldc\_host::port

**3.18.2.4** unsigned int cldc\_host::prio

**3.18.2.5** unsigned int cldc\_host::weight

The documentation for this struct was generated from the following file:

- include/[cldc.h](#)

## 3.19 cldc\_msg Struct Reference

an outgoing message, from client to server

```
#include <cldc.h>
```

### Data Fields

- uint64\_t seqid
- uint64\_t xid
- struct cldc\_session \* sess
- ssize\_t(\* cb )(struct cldc\_msg \*, const void \*, size\_t, bool)
- void \* cb\_private
- struct cldc\_call\_opts copts
- bool done
- time\_t expire\_time
- int retries
- int data\_len
- struct cld\_packet pkt
- uint8\_t data [0]

### 3.19.1 Detailed Description

an outgoing message, from client to server

### 3.19.2 Field Documentation

3.19.2.1 `ssize_t(* cldc_msg::cb)(struct cldc_msg *, const void *, size_t, bool)`

3.19.2.2 `void* cldc_msg::cb_private`

3.19.2.3 `struct cldc_call_opts cldc_msg::copts` [read]

3.19.2.4 `uint8_t cldc_msg::data[0]`

3.19.2.5 `int cldc_msg::data_len`

3.19.2.6 `bool cldc_msg::done`

3.19.2.7 `time_t cldc_msg::expire_time`

3.19.2.8 `struct cld_packet cldc_msg::pkt` [read]

3.19.2.9 `int cldc_msg::retries`

3.19.2.10 `uint64_t cldc_msg::seqid`

3.19.2.11 `struct cldc_session* cldc_msg::sess` [read]

3.19.2.12 `uint64_t cldc_msg::xid`

The documentation for this struct was generated from the following file:

- [include/cldc.h](#)

## 3.20 cldc\_ops Struct Reference

application-supplied facilities

```
#include <cldc.h>
```

### Data Fields

- `bool(* timer_ctl )(void *private, bool add, int(*cb)(struct cldc\_session *, void *), void *cb_private, time_t secs)`
- `int(* pkt_send )(void *private, const void *addr, size_t addrlen, const void *buf, size_t buflen)`
- `void(* event )(void *private, struct cldc\_session *, struct cldc\_fh *, uint32_t)`
- `void(* printf )(const char *fmt,...)`

### 3.20.1 Detailed Description

application-supplied facilities

### 3.20.2 Field Documentation

**3.20.2.1** `void(* cldc_ops::event)(void *private, struct cldc\_session *, struct cldc\_fh *, uint32_t)`

**3.20.2.2** `int(* cldc_ops::pkt_send)(void *private, const void *addr, size_t addrlen, const void *buf, size_t buflen)`

**3.20.2.3** `void(* cldc_ops::printf)(const char *fmt,...)`

**3.20.2.4** `bool(* cldc_ops::timer_ctl)(void *private, bool add, int(*cb)(struct cldc\_session *, void *), void *cb_private, time_t secs)`

The documentation for this struct was generated from the following file:

- `include/cldc.h`

## 3.21 cldc\_session Struct Reference

a single CLD client session

```
#include <cldc.h>
```

### Data Fields

- uint8\_t [sid](#) [CLD\_SID\_SZ]
- bool [verbose](#)
- struct [cldc\\_ops](#) \* [ops](#)
- void(\* [act\\_log](#) )(const char \*fmt,...)
- void \* [private](#)
- uint8\_t [addr](#) [64]
- size\_t [addr\\_len](#)
- GArray \* [fh](#)
- GList \* [out\\_msg](#)
- time\_t [msg\\_scan\\_time](#)
- GList \* [streams](#)
- time\_t [expire\\_time](#)
- bool [expired](#)
- uint64\_t [next\\_seqid\\_in](#)
- uint64\_t [next\\_seqid\\_in\\_tr](#)
- uint64\_t [next\\_seqid\\_out](#)
- char [user](#) [CLD\_MAX\_USERNAME]
- char [secret\\_key](#) [CLD\_MAX\_SECRET\_KEY]
- bool [confirmed](#)

### 3.21.1 Detailed Description

a single CLD client session

### 3.21.2 Field Documentation

- 3.21.2.1 `void(* cldc_session::act_log)(const char *fmt,...)`
- 3.21.2.2 `uint8_t cldc_session::addr[64]`
- 3.21.2.3 `size_t cldc_session::addr_len`
- 3.21.2.4 `bool cldc_session::confirmed`
- 3.21.2.5 `time_t cldc_session::expire_time`
- 3.21.2.6 `bool cldc_session::expired`
- 3.21.2.7 `GArray* cldc_session::fh`
- 3.21.2.8 `time_t cldc_session::msg_scan_time`
- 3.21.2.9 `uint64_t cldc_session::next_seqid_in`
- 3.21.2.10 `uint64_t cldc_session::next_seqid_in_tr`
- 3.21.2.11 `uint64_t cldc_session::next_seqid_out`
- 3.21.2.12 `struct cldc_ops* cldc_session::ops` [read]
- 3.21.2.13 `GList* cldc_session::out_msg`
- 3.21.2.14 `void* cldc_session::private`
- 3.21.2.15 `char cldc_session::secret_key[CLD_MAX_SECRET_KEY]`
- 3.21.2.16 `uint8_t cldc_session::sid[CLD_SID_SZ]`
- 3.21.2.17 `GList* cldc_session::streams`
- 3.21.2.18 `char cldc_session::user[CLD_MAX_USERNAME]`
- 3.21.2.19 `bool cldc_session::verbose`

The documentation for this struct was generated from the following file:

- [include/cldc.h](#)

## 3.22 cldc\_stream Struct Reference

internal per-data stream information

```
#include <cldc.h>
```

### Data Fields

- uint64\_t [strid\\_le](#)  
*stream id, LE*
- uint32\_t [size](#)  
*total bytes in stream*
- uint32\_t [next\\_seg](#)  
*next segment number expected*
- void \* [bufp](#)  
*pointer to next input loc*
- uint32\_t [size\\_left](#)  
*bytes remaining*
- struct [cldc\\_call\\_opts](#) [copts](#)  
*call options*
- char [buf](#) [0]  
*the raw data stream bytes*

### 3.22.1 Detailed Description

internal per-data stream information

### 3.22.2 Field Documentation

#### 3.22.2.1 char cldc\_stream::buf[0]

the raw data stream bytes

#### 3.22.2.2 void\* cldc\_stream::bufp

pointer to next input loc

#### 3.22.2.3 struct cldc\_call\_opts cldc\_stream::copts [read]

call options



**3.22.2.4 uint32\_t cldc\_stream::next\_seg**

next segment number expected

**3.22.2.5 uint32\_t cldc\_stream::size**

total bytes in stream

**3.22.2.6 uint32\_t cldc\_stream::size\_left**

bytes remaining

**3.22.2.7 uint64\_t cldc\_stream::strid\_le**

stream id, LE

The documentation for this struct was generated from the following file:

- [include/cldc.h](#)

## 3.23 cldc\_udp Struct Reference

A UDP implementation of the CLD client protocol.

```
#include <cldc.h>
```

### Data Fields

- uint8\_t [addr](#) [64]
- size\_t [addr\\_len](#)
- int [fd](#)
- struct event [timer\\_ev](#)
- struct [cldc\\_session](#) \* [sess](#)
- int(\* [cb](#) )(struct [cldc\\_session](#) \*, void \*)
- void \* [cb\\_private](#)

### 3.23.1 Detailed Description

A UDP implementation of the CLD client protocol.

### 3.23.2 Field Documentation

**3.23.2.1** uint8\_t [cldc\\_udp::addr](#)[64]

**3.23.2.2** size\_t [cldc\\_udp::addr\\_len](#)

**3.23.2.3** int(\* [cldc\\_udp::cb](#))(struct [cldc\\_session](#) \*, void \*)

**3.23.2.4** void\* [cldc\\_udp::cb\\_private](#)

**3.23.2.5** int [cldc\\_udp::fd](#)

**3.23.2.6** struct [cldc\\_session](#)\* [cldc\\_udp::sess](#) [read]

**3.23.2.7** struct event [cldc\\_udp::timer\\_ev](#) [read]

The documentation for this struct was generated from the following file:

- include/[cldc.h](#)

# Chapter 4

## File Documentation

### 4.1 include/cld\_msg.h File Reference

```
#include <stdint.h>
```

#### Data Structures

- struct [cld\\_packet](#)  
*header for each packet*
- struct [cld\\_msg\\_hdr](#)  
*header for each message*
- struct [cld\\_msg\\_resp](#)  
*standard response for each message*
- struct [cld\\_msg\\_open](#)  
*OPEN message.*
- struct [cld\\_msg\\_open\\_resp](#)  
*OPEN message response.*
- struct [cld\\_msg\\_get](#)  
*GET message.*
- struct [cld\\_msg\\_get\\_resp](#)  
*GET message response.*
- struct [cld\\_msg\\_data](#)  
*DATA message.*
- struct [cld\\_msg\\_put](#)  
*PUT message.*
- struct [cld\\_msg\\_close](#)

*CLOSE message.*

- struct `cld_msg_del`  
*DEL message.*
- struct `cld_msg_unlock`  
*UNLOCK message.*
- struct `cld_msg_lock`  
*LOCK message.*
- struct `cld_msg_event`  
*Server-to-client EVENT message.*

## Defines

- #define `CLD_PKT_MAGIC` "CLDc1pkt"
- #define `CLD_MSG_MAGIC` "CLDc1msg"
- #define `CLD_ALIGN8(n)`  $((8 - ((n) \& 7)) \& 7)$
- #define `SIDFMT` "%016lX"
- #define `SIDARG(sid)` `cld_sid2llu(sid)`

## Enumerations

- enum {  
`CLD_MAGIC_SZ` = 8, `CLD_SID_SZ` = 8, `CLD_INODE_NAME_MAX` = 256, `CLD_MAX_USERNAME` = 32,  
`CLD_MAX_SECRET_KEY` = 128, `CLD_MAX_DATA_MSGS` = 1024 }
- enum `cld_msg_ops` {  
`cmo_nop` = 0, `cmo_new_sess` = 1, `cmo_open` = 2, `cmo_get_meta` = 3,  
`cmo_get` = 4, `cmo_data_s` = 5, `cmo_put` = 6, `cmo_close` = 7,  
`cmo_del` = 8, `cmo_lock` = 9, `cmo_unlock` = 10, `cmo_trylock` = 11,  
`cmo_ack` = 12, `cmo_end_sess` = 13, `cmo_ping` = 30, `cmo_not_master` = 31,  
`cmo_event` = 32, `cmo_data_c` = 33 }  
*available RPC operations*
- enum `cle_err_codes` {  
`CLE_OK` = 0, `CLE_SESS_EXISTS` = 1, `CLE_SESS_INVAL` = 2, `CLE_DB_ERR` = 3,  
`CLE_BAD_PKT` = 4, `CLE_INODE_INVAL` = 5, `CLE_NAME_INVAL` = 6, `CLE_OOM` = 7,  
`CLE_FH_INVAL` = 8, `CLE_DATA_INVAL` = 9, `CLE_LOCK_INVAL` = 10, `CLE_LOCK_CONFLICT` = 11,  
`CLE_LOCK_PENDING` = 12, `CLE_MODE_INVAL` = 13, `CLE_INODE_EXISTS` = 14, `CLE_DIR_NOTEMPTY` = 15,  
`CLE_INTERNAL_ERR` = 16, `CLE_TIMEOUT` = 17, `CLE_SIG_INVAL` = 18 }  
*CLD error codes.*

- enum `cld_open_modes` {  
`COM_READ` = (1 << 0), `COM_WRITE` = (1 << 1), `COM_LOCK` = (1 << 2), `COM_ACL` = (1 << 3),  
`COM_CREATE` = (1 << 4), `COM_EXCL` = (1 << 5), `COM_DIRECTORY` = (1 << 6) }  
*availble OPEN mode flags*
- enum `cld_events` {  
`CE_UPDATED` = (1 << 0), `CE_DELETED` = (1 << 1), `CE_LOCKED` = (1 << 2), `CE_MASTER_FAILOVER` = (1 << 3),  
`CE_SESS_FAILED` = (1 << 4) }  
*potential events client may receive*
- enum `cld_lock_flags` { `CLF_SHARED` = (1 << 0) }  
*LOCK flags.*

## Functions

- unsigned long long `cld_sid2llu` (const uint8\_t \*sid)
- void `__cld_rand64` (void \*p)

### 4.1.1 Define Documentation

**4.1.1.1** `#define CLD_ALIGN8(n) ((8 - ((n) & 7)) & 7)`

**4.1.1.2** `#define CLD_MSG_MAGIC "CLDc1msg"`

**4.1.1.3** `#define CLD_PKT_MAGIC "CLDc1pkt"`

**4.1.1.4** `#define SIDARG(sid) cld_sid2llu(sid)`

**4.1.1.5** `#define SIDFMT "%016llx"`

### 4.1.2 Enumeration Type Documentation

#### 4.1.2.1 anonymous enum

Enumerator:

`CLD_MAGIC_SZ` length of magic number  
`CLD_SID_SZ` length of session id  
`CLD_INODE_NAME_MAX` max total pathname len  
`CLD_MAX_USERNAME` includes req.  
nul  
`CLD_MAX_SECRET_KEY` includes req.  
nul  
`CLD_MAX_DATA_MSGS` max data msgs in a stream

#### 4.1.2.2 enum cld\_events

potential events client may receive

**Enumerator:**

*CE\_UPDATED* contents updated  
*CE\_DELETED* inode deleted  
*CE\_LOCKED* lock acquired  
*CE\_MASTER\_FAILOVER* master failover  
*CE\_SESS\_FAILED*

#### 4.1.2.3 enum cld\_lock\_flags

LOCK flags.

**Enumerator:**

*CLF\_SHARED* a shared (read) lock

#### 4.1.2.4 enum cld\_msg\_ops

available RPC operations

**Enumerator:**

*cmo\_nop* no op  
*cmo\_new\_sess* new session  
*cmo\_open* open file  
*cmo\_get\_meta* get metadata  
*cmo\_get* get metadata + data  
*cmo\_data\_s* data message to server  
*cmo\_put* put data  
*cmo\_close* close file  
*cmo\_del* delete file  
*cmo\_lock* lock  
*cmo\_unlock* unlock  
*cmo\_trylock* trylock  
*cmo\_ack* ack of seqid rx'd  
*cmo\_end\_sess* end session  
*cmo\_ping* server to client ping  
*cmo\_not\_master* I am not the master!  
*cmo\_event* server->cli async event  
*cmo\_data\_c* data message to client

#### 4.1.2.5 enum cld\_open\_modes

available OPEN mode flags

**Enumerator:**

*COM\_READ* read  
*COM\_WRITE* write  
*COM\_LOCK* lock  
*COM\_ACL* ACL update.  
*COM\_CREATE* create file, if not exist  
*COM\_EXCL* fail create if file exists  
*COM\_DIRECTORY* operate on a directory

#### 4.1.2.6 enum cle\_err\_codes

CLD error codes.

**Enumerator:**

*CLE\_OK* success / no error  
*CLE\_SESS\_EXISTS* session exists  
*CLE\_SESS\_INVALID* session doesn't exist  
*CLE\_DB\_ERR* db error  
*CLE\_BAD\_PKT* invalid/corrupted packet  
*CLE\_INODE\_INVALID* inode doesn't exist  
*CLE\_NAME\_INVALID* inode name invalid  
*CLE\_OOM* server out of memory  
*CLE\_FH\_INVALID* file handle invalid  
*CLE\_DATA\_INVALID* invalid data pkt  
*CLE\_LOCK\_INVALID* invalid lock  
*CLE\_LOCK\_CONFLICT* conflicting lock held  
*CLE\_LOCK\_PENDING* lock waiting to be acq.  
*CLE\_MODE\_INVALID* op incompat.  
w/ file mode  
*CLE\_INODE\_EXISTS* inode exists  
*CLE\_DIR\_NOTEMPTY* dir not empty  
*CLE\_INTERNAL\_ERR* nonspecific internal err  
*CLE\_TIMEOUT* session timed out  
*CLE\_SIG\_INVALID* HMAC sig bad / auth failed.

### 4.1.3 Function Documentation

#### 4.1.3.1 void \_\_cld\_rand64 (void \* p)

#### 4.1.3.2 unsigned long long cld\_sid2llu (const uint8\_t \* sid)

## 4.2 include/cldc.h File Reference

```
#include <sys/types.h>
#include <stdbool.h>
#include <event.h>
#include <glib.h>
#include <cld_msg.h>
```

### Data Structures

- struct [cldc\\_call\\_opts](#)  
*per-operation application options*
- struct [cldc\\_stream](#)  
*internal per-data stream information*
- struct [cldc\\_msg](#)  
*an outgoing message, from client to server*
- struct [cldc\\_fh](#)  
*an open file handle associated with a session*
- struct [cldc\\_ops](#)  
*application-supplied facilities*
- struct [cldc\\_session](#)  
*a single CLD client session*
- struct [cldc\\_host](#)  
*Information for a single CLD server host.*
- struct [cldc\\_udp](#)  
*A UDP implementation of the CLD client protocol.*
- struct [cld\\_dirent\\_cur](#)

### Functions

- int [cldc\\_receive\\_pkt](#) (struct [cldc\\_session](#) \*sess, const void \*net\_addr, size\_t net\_addrlen, const void \*buf, size\_t buflen)  
*Packet received from remote host.*
- int [cldc\\_new\\_sess](#) (const struct [cldc\\_ops](#) \*ops, const struct [cldc\\_call\\_opts](#) \*copts, const void \*addr, size\_t addr\_len, const char \*user, const char \*secret\_key, void \*private, struct [cldc\\_session](#) \*\*sess\_out)
- void [cldc\\_kill\\_sess](#) (struct [cldc\\_session](#) \*sess)
- int [cldc\\_end\\_sess](#) (struct [cldc\\_session](#) \*sess, const struct [cldc\\_call\\_opts](#) \*copts)
- int [cldc\\_nop](#) (struct [cldc\\_session](#) \*sess, const struct [cldc\\_call\\_opts](#) \*copts)



- int `cldc_del` (struct `cldc_session` \*sess, const struct `cldc_call_opts` \*copts, const char \*pathname)
- int `cldc_open` (struct `cldc_session` \*sess, const struct `cldc_call_opts` \*copts, const char \*pathname, uint32\_t open\_mode, uint32\_t events, struct `cldc_fh` \*\*fh\_out)
- int `cldc_close` (struct `cldc_fh` \*fh, const struct `cldc_call_opts` \*copts)
- int `cldc_unlock` (struct `cldc_fh` \*fh, const struct `cldc_call_opts` \*copts)
- int `cldc_lock` (struct `cldc_fh` \*fh, const struct `cldc_call_opts` \*copts, uint32\_t lock\_flags, bool wait\_for\_lock)
- int `cldc_put` (struct `cldc_fh` \*fh, const struct `cldc_call_opts` \*copts, const void \*data, size\_t data\_len)
- int `cldc_get` (struct `cldc_fh` \*fh, const struct `cldc_call_opts` \*copts, bool metadata\_only)
- int `cldc_dirent_count` (const void \*data, size\_t data\_len)
- int `cldc_dirent_first` (struct `cld_dirent_cur` \*dc)
- int `cldc_dirent_next` (struct `cld_dirent_cur` \*dc)
- void `cldc_dirent_cur_init` (struct `cld_dirent_cur` \*dc, const void \*buf, size\_t buflen)
- void `cldc_dirent_cur_fini` (struct `cld_dirent_cur` \*dc)
- char \* `cldc_dirent_name` (struct `cld_dirent_cur` \*dc)
- void `cldc_udp_free` (struct `cldc_udp` \*udp)
- int `cldc_udp_new` (const char \*hostname, int port, struct `cldc_udp` \*\*udp\_out)
- int `cldc_udp_receive_pkt` (struct `cldc_udp` \*udp)
- int `cldc_udp_pkt_send` (void \*private, const void \*addr, size\_t addrlen, const void \*buf, size\_t buflen)
- bool `cldc_levent_timer` (void \*private, bool add, int(\*cb)(struct `cldc_session` \*, void \*), void \*cb\_private, time\_t secs)
- int `cldc_getaddr` (GList \*\*host\_list, const char \*thishost, bool verbose, void(\*act\_log)(const char \*fmt,...))
- int `cldc_saveaddr` (struct `cldc_host` \*hp, unsigned int priority, unsigned int weight, unsigned int port, unsigned int nlen, const char \*name, bool verbose, void(\*act\_log)(const char \*fmt,...))

## 4.2.1 Function Documentation

- 4.2.1.1 `int cldc_close (struct cldc_fh *fh, const struct cldc_call_opts *copts)`
- 4.2.1.2 `int cldc_del (struct cldc_session *sess, const struct cldc_call_opts *copts, const char *pathname)`
- 4.2.1.3 `int cldc_dirent_count (const void *data, size_t data_len)`
- 4.2.1.4 `void cldc_dirent_cur_fini (struct cld_dirent_cur *dc)`
- 4.2.1.5 `void cldc_dirent_cur_init (struct cld_dirent_cur *dc, const void *buf, size_t buflen)`
- 4.2.1.6 `int cldc_dirent_first (struct cld_dirent_cur *dc)`
- 4.2.1.7 `char* cldc_dirent_name (struct cld_dirent_cur *dc)`
- 4.2.1.8 `int cldc_dirent_next (struct cld_dirent_cur *dc)`
- 4.2.1.9 `int cldc_end_sess (struct cldc_session *sess, const struct cldc_call_opts *copts)`
- 4.2.1.10 `int cldc_get (struct cldc_fh *fh, const struct cldc_call_opts *copts, bool metadata_only)`
- 4.2.1.11 `int cldc_getaddr (GList **host_list, const char *thishost, bool verbose, void (*)(const char *fmt,...) act_log)`
- 4.2.1.12 `void cldc_kill_sess (struct cldc_session *sess)`
- 4.2.1.13 `bool cldc_levent_timer (void *private, bool add, int (*)(struct cldc_session *, void *) cb, void *cb_private, time_t secs)`
- 4.2.1.14 `int cldc_lock (struct cldc_fh *fh, const struct cldc_call_opts *copts, uint32_t lock_flags, bool wait_for_lock)`
- 4.2.1.15 `int cldc_new_sess (const struct cldc_ops *ops, const struct cldc_call_opts *copts, const void *addr, size_t addr_len, const char *user, const char *secret_key, void *private, struct cldc_session **sess_out)`
- 4.2.1.16 `int cldc_nop (struct cldc_session *sess, const struct cldc_call_opts *copts)`
- 4.2.1.17 `int cldc_open (struct cldc_session *sess, const struct cldc_call_opts *copts, const char *pathname, uint32_t open_mode, uint32_t events, struct cldc_fh **fh_out)`
- 4.2.1.18 `int cldc_put (struct cldc_fh *fh, const struct cldc_call_opts *copts, const void *data, size_t data_len)`
- 4.2.1.19 `int cldc_receive_pkt (struct cldc_session *sess, const void *net_addr, size_t net_addrlen, const void *buf, size_t buflen)`

Packet received from remote host.

Called by app when a packet is received from a remote host over the network.

**Parameters:**

*sess* Session associated with received packet  
*net\_addr* Opaque network address  
*net\_addrlen* Size of opaque network address  
*buf* Pointer to data buffer containing packet  
*buflen* Length of received packet

**Returns:**

Zero for success, non-zero on error

**4.2.1.20** `int cldc_saveaddr (struct cldc_host * hp, unsigned int priority, unsigned int weight, unsigned int port, unsigned int nlen, const char * name, bool verbose, void(*) (const char * fmt,...) act_log)`

**4.2.1.21** `void cldc_udp_free (struct cldc_udp * udp)`

**4.2.1.22** `int cldc_udp_new (const char * hostname, int port, struct cldc_udp ** udp_out)`

**4.2.1.23** `int cldc_udp_pkt_send (void * private, const void * addr, size_t addrlen, const void * buf, size_t buflen)`

**4.2.1.24** `int cldc_udp_receive_pkt (struct cldc_udp * udp)`

**4.2.1.25** `int cldc_unlock (struct cldc_fh * fh, const struct cldc_call_opts * copts)`

# Index

- [\\_\\_cld\\_rand64](#)
  - [cld\\_msg.h, 37](#)
- [act\\_log](#)
  - [cldc\\_session, 29](#)
- [addr](#)
  - [cldc\\_session, 29](#)
  - [cldc\\_udp, 32](#)
- [addr\\_len](#)
  - [cldc\\_session, 29](#)
  - [cldc\\_udp, 32](#)
- [buf](#)
  - [cldc\\_call\\_opts, 22](#)
  - [cldc\\_stream, 30](#)
- [bufp](#)
  - [cldc\\_stream, 30](#)
- [cb](#)
  - [cldc\\_call\\_opts, 22](#)
  - [cldc\\_msg, 26](#)
  - [cldc\\_udp, 32](#)
- [cb\\_private](#)
  - [cldc\\_msg, 26](#)
  - [cldc\\_udp, 32](#)
- [CE\\_DELETED](#)
  - [cld\\_msg.h, 36](#)
- [CE\\_LOCKED](#)
  - [cld\\_msg.h, 36](#)
- [CE\\_MASTER\\_FAILOVER](#)
  - [cld\\_msg.h, 36](#)
- [CE\\_SESS\\_FAILED](#)
  - [cld\\_msg.h, 36](#)
- [CE\\_UPDATED](#)
  - [cld\\_msg.h, 36](#)
- [CLD\\_INODE\\_NAME\\_MAX](#)
  - [cld\\_msg.h, 35](#)
- [CLD\\_MAGIC\\_SZ](#)
  - [cld\\_msg.h, 35](#)
- [CLD\\_MAX\\_DATA\\_MSGS](#)
  - [cld\\_msg.h, 35](#)
- [CLD\\_MAX\\_SECRET\\_KEY](#)
  - [cld\\_msg.h, 35](#)
- [CLD\\_MAX\\_USERNAME](#)
  - [cld\\_msg.h, 35](#)
- [cld\\_msg.h](#)
  - [CE\\_DELETED, 36](#)
  - [CE\\_LOCKED, 36](#)
  - [CE\\_MASTER\\_FAILOVER, 36](#)
  - [CE\\_SESS\\_FAILED, 36](#)
  - [CE\\_UPDATED, 36](#)
  - [CLD\\_INODE\\_NAME\\_MAX, 35](#)
  - [CLD\\_MAGIC\\_SZ, 35](#)
  - [CLD\\_MAX\\_DATA\\_MSGS, 35](#)
  - [CLD\\_MAX\\_SECRET\\_KEY, 35](#)
  - [CLD\\_MAX\\_USERNAME, 35](#)
  - [CLD\\_SID\\_SZ, 35](#)
  - [CLE\\_BAD\\_PKT, 37](#)
  - [CLE\\_DATA\\_INVAL, 37](#)
  - [CLE\\_DB\\_ERR, 37](#)
  - [CLE\\_DIR\\_NOTEMPTY, 37](#)
  - [CLE\\_FH\\_INVAL, 37](#)
  - [CLE\\_INODE\\_EXISTS, 37](#)
  - [CLE\\_INODE\\_INVAL, 37](#)
  - [CLE\\_INTERNAL\\_ERR, 37](#)
  - [CLE\\_LOCK\\_CONFLICT, 37](#)
  - [CLE\\_LOCK\\_INVAL, 37](#)
  - [CLE\\_LOCK\\_PENDING, 37](#)
  - [CLE\\_MODE\\_INVAL, 37](#)
  - [CLE\\_NAME\\_INVAL, 37](#)
  - [CLE\\_OK, 37](#)
  - [CLE\\_OOM, 37](#)
  - [CLE\\_SESS\\_EXISTS, 37](#)
  - [CLE\\_SESS\\_INVAL, 37](#)
  - [CLE\\_SIG\\_INVAL, 37](#)
  - [CLE\\_TIMEOUT, 37](#)
  - [CLF\\_SHARED, 36](#)
  - [cmo\\_ack, 36](#)
  - [cmo\\_close, 36](#)
  - [cmo\\_data\\_c, 36](#)
  - [cmo\\_data\\_s, 36](#)
  - [cmo\\_del, 36](#)
  - [cmo\\_end\\_sess, 36](#)
  - [cmo\\_event, 36](#)
  - [cmo\\_get, 36](#)
  - [cmo\\_get\\_meta, 36](#)
  - [cmo\\_lock, 36](#)
  - [cmo\\_new\\_sess, 36](#)
  - [cmo\\_nop, 36](#)
  - [cmo\\_not\\_master, 36](#)

- cmo\_open, 36
- cmo\_ping, 36
- cmo\_put, 36
- cmo\_trylock, 36
- cmo\_unlock, 36
- COM\_ACL, 37
- COM\_CREATE, 37
- COM\_DIRECTORY, 37
- COM\_EXCL, 37
- COM\_LOCK, 37
- COM\_READ, 37
- COM\_WRITE, 37
- CLD\_SID\_SZ
  - cld\_msg.h, 35
- CLD\_ALIGN8
  - cld\_msg.h, 35
- cld\_dirent\_cur, 5
  - p, 5
  - tmp\_len, 5
- cld\_events
  - cld\_msg.h, 35
- cld\_lock\_flags
  - cld\_msg.h, 36
- cld\_msg.h
  - \_\_cld\_rand64, 37
  - CLD\_ALIGN8, 35
  - cld\_events, 35
  - cld\_lock\_flags, 36
  - CLD\_MSG\_MAGIC, 35
  - cld\_msg\_ops, 36
  - cld\_open\_modes, 36
  - CLD\_PKT\_MAGIC, 35
  - cld\_sid2llu, 37
  - cle\_err\_codes, 37
  - SIDARG, 35
  - SIDFMT, 35
- cld\_msg\_close, 6
  - fh, 6
  - hdr, 6
- cld\_msg\_data, 7
  - hdr, 7
  - seg, 7
  - seg\_len, 7
  - strid, 7
- cld\_msg\_del, 8
  - hdr, 8
  - name\_len, 8
- cld\_msg\_event, 9
  - events, 9
  - fh, 9
  - hdr, 9
- cld\_msg\_get, 10
  - fh, 10
  - hdr, 10
- cld\_msg\_get\_resp, 11
  - flags, 11
  - ino\_len, 11
  - inum, 11
  - resp, 12
  - size, 12
  - strid, 12
  - time\_create, 12
  - time\_modify, 12
  - version, 12
- cld\_msg\_hdr, 13
  - magic, 13
  - op, 13
  - res1, 13
  - xid, 13
- cld\_msg\_lock, 14
  - fh, 14
  - flags, 14
  - hdr, 14
- CLD\_MSG\_MAGIC
  - cld\_msg.h, 35
- cld\_msg\_open, 15
  - events, 15
  - hdr, 15
  - mode, 15
  - name\_len, 15
- cld\_msg\_open\_resp, 16
  - fh, 16
  - resp, 16
- cld\_msg\_ops
  - cld\_msg.h, 36
- cld\_msg\_put, 17
  - data\_size, 17
  - fh, 17
  - hdr, 17
  - strid, 17
- cld\_msg\_resp, 18
  - code, 18
  - hdr, 18
  - rsv, 18
  - xid\_in, 18
- cld\_msg\_unlock, 19
  - fh, 19
  - hdr, 19
- cld\_open\_modes
  - cld\_msg.h, 36
- cld\_packet, 20
  - magic, 20
  - n\_msg, 20
  - res, 20
  - seqid, 20
  - sid, 20
  - user, 20
- CLD\_PKT\_MAGIC

- cld\_msg.h, 35
- cld\_sid2llu
  - cld\_msg.h, 37
- cldc.h
  - cldc\_close, 40
  - cldc\_del, 40
  - cldc\_dirent\_count, 40
  - cldc\_dirent\_cur\_fini, 40
  - cldc\_dirent\_cur\_init, 40
  - cldc\_dirent\_first, 40
  - cldc\_dirent\_name, 40
  - cldc\_dirent\_next, 40
  - cldc\_end\_sess, 40
  - cldc\_get, 40
  - cldc\_getaddr, 40
  - cldc\_kill\_sess, 40
  - cldc\_levent\_timer, 40
  - cldc\_lock, 40
  - cldc\_new\_sess, 40
  - cldc\_nop, 40
  - cldc\_open, 40
  - cldc\_put, 40
  - cldc\_receive\_pkt, 40
  - cldc\_saveaddr, 41
  - cldc\_udp\_free, 41
  - cldc\_udp\_new, 41
  - cldc\_udp\_pkt\_send, 41
  - cldc\_udp\_receive\_pkt, 41
  - cldc\_unlock, 41
- cldc\_call\_opts, 22
  - buf, 22
  - cb, 22
  - get, 22
  - inode\_name, 22
  - op, 22
  - private, 22
  - resp, 22
  - size, 22
  - u, 22
- cldc\_close
  - cldc.h, 40
- cldc\_del
  - cldc.h, 40
- cldc\_dirent\_count
  - cldc.h, 40
- cldc\_dirent\_cur\_fini
  - cldc.h, 40
- cldc\_dirent\_cur\_init
  - cldc.h, 40
- cldc\_dirent\_first
  - cldc.h, 40
- cldc\_dirent\_name
  - cldc.h, 40
- cldc\_dirent\_next
  - cldc.h, 40
- cldc\_end\_sess
  - cldc.h, 40
- cldc\_fh, 23
  - fh\_le, 23
  - sess, 23
  - valid, 23
- cldc\_get
  - cldc.h, 40
- cldc\_getaddr
  - cldc.h, 40
- cldc\_host, 24
  - host, 24
  - known, 24
  - port, 24
  - prio, 24
  - weight, 24
- cldc\_kill\_sess
  - cldc.h, 40
- cldc\_levent\_timer
  - cldc.h, 40
- cldc\_lock
  - cldc.h, 40
- cldc\_msg, 25
  - cb, 26
  - cb\_private, 26
  - copts, 26
  - data, 26
  - data\_len, 26
  - done, 26
  - expire\_time, 26
  - pkt, 26
  - retries, 26
  - seqid, 26
  - sess, 26
  - xid, 26
- cldc\_new\_sess
  - cldc.h, 40
- cldc\_nop
  - cldc.h, 40
- cldc\_open
  - cldc.h, 40
- cldc\_ops, 27
  - event, 27
  - pkt\_send, 27
  - printf, 27
  - timer\_ctl, 27
- cldc\_put
  - cldc.h, 40
- cldc\_receive\_pkt
  - cldc.h, 40
- cldc\_saveaddr
  - cldc.h, 41
- cldc\_session, 28

- act\_log, 29
- addr, 29
- addr\_len, 29
- confirmed, 29
- expire\_time, 29
- expired, 29
- fh, 29
- msg\_scan\_time, 29
- next\_seqid\_in, 29
- next\_seqid\_in\_tr, 29
- next\_seqid\_out, 29
- ops, 29
- out\_msg, 29
- private, 29
- secret\_key, 29
- sid, 29
- streams, 29
- user, 29
- verbose, 29
- cldc\_stream, 30
  - buf, 30
  - bufp, 30
  - copts, 30
  - next\_seg, 30
  - size, 31
  - size\_left, 31
  - strid\_le, 31
- cldc\_udp, 32
  - addr, 32
  - addr\_len, 32
  - cb, 32
  - cb\_private, 32
  - fd, 32
  - sess, 32
  - timer\_ev, 32
- cldc\_udp\_free
  - cldc.h, 41
- cldc\_udp\_new
  - cldc.h, 41
- cldc\_udp\_pkt\_send
  - cldc.h, 41
- cldc\_udp\_receive\_pkt
  - cldc.h, 41
- cldc\_unlock
  - cldc.h, 41
- CLE\_BAD\_PKT
  - cld\_msg.h, 37
- CLE\_DATA\_INVAL
  - cld\_msg.h, 37
- CLE\_DB\_ERR
  - cld\_msg.h, 37
- CLE\_DIR\_NOTEMPTY
  - cld\_msg.h, 37
- CLE\_FH\_INVAL
  - cld\_msg.h, 37
- CLE\_INODE\_EXISTS
  - cld\_msg.h, 37
- CLE\_INODE\_INVAL
  - cld\_msg.h, 37
- CLE\_INTERNAL\_ERR
  - cld\_msg.h, 37
- CLE\_LOCK\_CONFLICT
  - cld\_msg.h, 37
- CLE\_LOCK\_INVAL
  - cld\_msg.h, 37
- CLE\_LOCK\_PENDING
  - cld\_msg.h, 37
- CLE\_MODE\_INVAL
  - cld\_msg.h, 37
- CLE\_NAME\_INVAL
  - cld\_msg.h, 37
- CLE\_OK
  - cld\_msg.h, 37
- CLE\_OOM
  - cld\_msg.h, 37
- CLE\_SESS\_EXISTS
  - cld\_msg.h, 37
- CLE\_SESS\_INVAL
  - cld\_msg.h, 37
- CLE\_SIG\_INVAL
  - cld\_msg.h, 37
- CLE\_TIMEOUT
  - cld\_msg.h, 37
- cle\_err\_codes
  - cld\_msg.h, 37
- CLF\_SHARED
  - cld\_msg.h, 36
- cmo\_ack
  - cld\_msg.h, 36
- cmo\_close
  - cld\_msg.h, 36
- cmo\_data\_c
  - cld\_msg.h, 36
- cmo\_data\_s
  - cld\_msg.h, 36
- cmo\_del
  - cld\_msg.h, 36
- cmo\_end\_sess
  - cld\_msg.h, 36
- cmo\_event
  - cld\_msg.h, 36
- cmo\_get
  - cld\_msg.h, 36
- cmo\_get\_meta
  - cld\_msg.h, 36
- cmo\_lock
  - cld\_msg.h, 36
- cmo\_new\_sess

- cld\_msg.h, 36
- cmo\_nop
  - cld\_msg.h, 36
- cmo\_not\_master
  - cld\_msg.h, 36
- cmo\_open
  - cld\_msg.h, 36
- cmo\_ping
  - cld\_msg.h, 36
- cmo\_put
  - cld\_msg.h, 36
- cmo\_trylock
  - cld\_msg.h, 36
- cmo\_unlock
  - cld\_msg.h, 36
- code
  - cld\_msg\_resp, 18
- COM\_ACL
  - cld\_msg.h, 37
- COM\_CREATE
  - cld\_msg.h, 37
- COM\_DIRECTORY
  - cld\_msg.h, 37
- COM\_EXCL
  - cld\_msg.h, 37
- COM\_LOCK
  - cld\_msg.h, 37
- COM\_READ
  - cld\_msg.h, 37
- COM\_WRITE
  - cld\_msg.h, 37
- confirmed
  - cldc\_session, 29
- copts
  - cldc\_msg, 26
  - cldc\_stream, 30
- data
  - cldc\_msg, 26
- data\_len
  - cldc\_msg, 26
- data\_size
  - cld\_msg\_put, 17
- done
  - cldc\_msg, 26
- event
  - cldc\_ops, 27
- events
  - cld\_msg\_event, 9
  - cld\_msg\_open, 15
- expire\_time
  - cldc\_msg, 26
  - cldc\_session, 29
- expired
  - cldc\_session, 29
- fd
  - cldc\_udp, 32
- fh
  - cld\_msg\_close, 6
  - cld\_msg\_event, 9
  - cld\_msg\_get, 10
  - cld\_msg\_lock, 14
  - cld\_msg\_open\_resp, 16
  - cld\_msg\_put, 17
  - cld\_msg\_unlock, 19
  - cldc\_session, 29
- fh\_le
  - cldc\_fh, 23
- flags
  - cld\_msg\_get\_resp, 11
  - cld\_msg\_lock, 14
- get
  - cldc\_call\_opts, 22
- hdr
  - cld\_msg\_close, 6
  - cld\_msg\_data, 7
  - cld\_msg\_del, 8
  - cld\_msg\_event, 9
  - cld\_msg\_get, 10
  - cld\_msg\_lock, 14
  - cld\_msg\_open, 15
  - cld\_msg\_put, 17
  - cld\_msg\_resp, 18
  - cld\_msg\_unlock, 19
- host
  - cldc\_host, 24
- include/cld\_msg.h, 33
- include/cldc.h, 38
- ino\_len
  - cld\_msg\_get\_resp, 11
- inode\_name
  - cldc\_call\_opts, 22
- inum
  - cld\_msg\_get\_resp, 11
- known
  - cldc\_host, 24
- magic
  - cld\_msg\_hdr, 13
  - cld\_packet, 20
- mode
  - cld\_msg\_open, 15
- msg\_scan\_time



- cldc\_session, 29
- n\_msg
  - cld\_packet, 20
- name\_len
  - cld\_msg\_del, 8
  - cld\_msg\_open, 15
- next\_seg
  - cldc\_stream, 30
- next\_seqid\_in
  - cldc\_session, 29
- next\_seqid\_in\_tr
  - cldc\_session, 29
- next\_seqid\_out
  - cldc\_session, 29
- op
  - cld\_msg\_hdr, 13
  - cldc\_call\_opts, 22
- ops
  - cldc\_session, 29
- out\_msg
  - cldc\_session, 29
- p
  - cld\_dirent\_cur, 5
- pkt
  - cldc\_msg, 26
- pkt\_send
  - cldc\_ops, 27
- port
  - cldc\_host, 24
- printf
  - cldc\_ops, 27
- prio
  - cldc\_host, 24
- private
  - cldc\_call\_opts, 22
  - cldc\_session, 29
- res
  - cld\_packet, 20
- res1
  - cld\_msg\_hdr, 13
- resp
  - cld\_msg\_get\_resp, 12
  - cld\_msg\_open\_resp, 16
  - cldc\_call\_opts, 22
- retries
  - cldc\_msg, 26
- rsv
  - cld\_msg\_resp, 18
- secret\_key
  - cldc\_session, 29
- seg
  - cld\_msg\_data, 7
- seg\_len
  - cld\_msg\_data, 7
- seqid
  - cld\_packet, 20
  - cldc\_msg, 26
- sess
  - cldc\_fh, 23
  - cldc\_msg, 26
  - cldc\_udp, 32
- sid
  - cld\_packet, 20
  - cldc\_session, 29
- SIDARG
  - cld\_msg.h, 35
- SIDFMT
  - cld\_msg.h, 35
- size
  - cld\_msg\_get\_resp, 12
  - cldc\_call\_opts, 22
  - cldc\_stream, 31
- size\_left
  - cldc\_stream, 31
- streams
  - cldc\_session, 29
- strid
  - cld\_msg\_data, 7
  - cld\_msg\_get\_resp, 12
  - cld\_msg\_put, 17
- strid\_le
  - cldc\_stream, 31
- time\_create
  - cld\_msg\_get\_resp, 12
- time\_modify
  - cld\_msg\_get\_resp, 12
- timer\_ctl
  - cldc\_ops, 27
- timer\_ev
  - cldc\_udp, 32
- tmp\_len
  - cld\_dirent\_cur, 5
- u
  - cldc\_call\_opts, 22
- user
  - cld\_packet, 20
  - cldc\_session, 29
- valid
  - cldc\_fh, 23
- verbose
  - cldc\_session, 29

version  
    cld\_msg\_get\_resp, [12](#)

weight  
    cldc\_host, [24](#)

xid  
    cld\_msg\_hdr, [13](#)  
    cldc\_msg, [26](#)

xid\_in  
    cld\_msg\_resp, [18](#)