

CLD

0.1git

Generated by Doxygen 1.5.9

Tue Aug 11 09:24:54 2009

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
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	data_len	26
3.19.2.6	done	26
3.19.2.7	expire_time	26
3.19.2.8	pkt	26
3.19.2.9	retries	26
3.19.2.10	seqid	26
3.19.2.11	sess	26
3.19.2.12	xid	26
3.20	cldc_ops Struct Reference	27
3.20.1	Detailed Description	27
3.20.2	Field Documentation	27
3.20.2.1	event	27
3.20.2.2	pkt_send	27
3.20.2.3	printf	27
3.20.2.4	timer_ctl	27
3.21	cldc_session Struct Reference	28
3.21.1	Detailed Description	28
3.21.2	Field Documentation	29
3.21.2.1	act_log	29
3.21.2.2	addr	29
3.21.2.3	addr_len	29
3.21.2.4	confirmed	29
3.21.2.5	expire_time	29
3.21.2.6	expired	29
3.21.2.7	fh	29
3.21.2.8	msg_scan_time	29
3.21.2.9	next_seqid_in	29
3.21.2.10	next_seqid_in_tr	29
3.21.2.11	next_seqid_out	29
3.21.2.12	ops	29
3.21.2.13	out_msg	29
3.21.2.14	private	29
3.21.2.15	secret_key	29
3.21.2.16	sid	29
3.21.2.17	streams	29
3.21.2.18	user	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
4	File Documentation	33
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:

cld_dirent_cur	5
cld_msg_close (CLOSE message)	6
cld_msg_data (DATA message)	7
cld_msg_del (DEL message)	8
cld_msg_event (Server-to-client EVENT message)	9
cld_msg_get (GET message)	10
cld_msg_get_resp (GET message response)	11
cld_msg_hdr (Header for each message)	13
cld_msg_lock (LOCK message)	14
cld_msg_open (OPEN message)	15
cld_msg_open_resp (OPEN message response)	16
cld_msg_put (PUT message)	17
cld_msg_resp (Standard response for each message)	18
cld_msg_unlock (UNLOCK message)	19
cld_packet (Header for each packet)	20
cldc_call_opts (Per-operation application options)	22
cldc_fh (Open file handle associated with a session)	23
cldc_host (Information for a single CLD server host)	24
cldc_msg (Outgoing message, from client to server)	25
cldc_ops (Application-supplied facilities)	27
cldc_session (Single CLD client session)	28
cldc_stream (Internal per-data stream information)	30
cldc_udp (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/ cld_msg.h	33
include/ cldc.h	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`
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 clde_stream::next_seg

next segment number expected

3.22.2.5 uint32_t clde_stream::size

total bytes in stream

3.22.2.6 uint32_t clde_stream::size_left

bytes remaining

3.22.2.7 uint64_t clde_stream::strid_le

stream id, LE

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

- [include/clde.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](#)