

CLD

0.1git

Generated by Doxygen 1.6.1

Thu Aug 27 03:22:30 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_ack_frag Struct Reference	6
3.2.1	Detailed Description	6
3.2.2	Field Documentation	6
3.2.2.1	hdr	6
3.2.2.2	seqid	6
3.3	cld_msg_close Struct Reference	7
3.3.1	Detailed Description	7
3.3.2	Field Documentation	7
3.3.2.1	fh	7
3.3.2.2	hdr	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
3.4.2.3	res	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
3.5.2.4	res	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	11
3.7.2.4	res	12
3.7.2.5	resp	12
3.7.2.6	size	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
3.9.2.4	res	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
3.10.2.5 res	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 res	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 flags	20

3.15.2.2	magic	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	port	24
3.18.2.3	prio	24
3.18.2.4	weight	24
3.19	cldc_msg Struct Reference	25
3.19.1	Detailed Description	25
3.19.2	Field Documentation	25
3.19.2.1	cb	25
3.19.2.2	cb_private	25
3.19.2.3	copts	25

3.19.2.4	data	25
3.19.2.5	data_len	25
3.19.2.6	done	25
3.19.2.7	expire_time	25
3.19.2.8	n_pkts	25
3.19.2.9	pkt_info	25
3.19.2.10	sess	25
3.19.2.11	xid	25
3.20	cldc_ops Struct Reference	26
3.20.1	Detailed Description	26
3.20.2	Field Documentation	26
3.20.2.1	errlog	26
3.20.2.2	event	26
3.20.2.3	pkt_send	26
3.20.2.4	timer_ctl	26
3.21	cldc_pkt_info Struct Reference	27
3.21.1	Field Documentation	27
3.21.1.1	data	27
3.21.1.2	pkt	27
3.21.1.3	pkt_len	27
3.21.1.4	retries	27
3.22	cldc_session Struct Reference	28
3.22.1	Detailed Description	28
3.22.2	Field Documentation	29
3.22.2.1	act_log	29
3.22.2.2	addr	29
3.22.2.3	addr_len	29
3.22.2.4	confirmed	29
3.22.2.5	expire_time	29
3.22.2.6	expired	29
3.22.2.7	fh	29
3.22.2.8	msg_buf	29
3.22.2.9	msg_buf_len	29
3.22.2.10	msg_scan_time	29
3.22.2.11	next_seqid_in	29
3.22.2.12	next_seqid_in_tr	29

3.22.2.13	next_seqid_out	29
3.22.2.14	ops	29
3.22.2.15	out_msg	29
3.22.2.16	private	29
3.22.2.17	secret_key	29
3.22.2.18	sid	29
3.22.2.19	user	29
3.22.2.20	verbose	29
3.23	cldc_udp Struct Reference	30
3.23.1	Detailed Description	30
3.23.2	Field Documentation	30
3.23.2.1	addr	30
3.23.2.2	addr_len	30
3.23.2.3	cb	30
3.23.2.4	cb_private	30
3.23.2.5	fd	30
3.23.2.6	sess	30
3.23.2.7	timer_ev	30
4	File Documentation	31
4.1	include/cld-private.h File Reference	31
4.2	include/cld_msg.h File Reference	32
4.2.1	Define Documentation	34
4.2.1.1	CLD_ALIGN8	34
4.2.1.2	CLD_MSG_MAGIC	34
4.2.1.3	CLD_PKT_MAGIC	34
4.2.1.4	SIDARG	34
4.2.1.5	SIDFMT	34
4.2.2	Enumeration Type Documentation	34
4.2.2.1	"@0	34
4.2.2.2	cld_events	34
4.2.2.3	cld_lock_flags	35
4.2.2.4	cld_msg_ops	35
4.2.2.5	cld_open_modes	35
4.2.2.6	cld_packet_flags	36
4.2.2.7	cle_err_codes	36
4.2.3	Function Documentation	36

4.2.3.1	<code>__cld_rand64</code>	36
4.2.3.2	<code>cld_sid2llu</code>	36
4.3	<code>include/cldc.h</code> File Reference	37
4.3.1	Function Documentation	39
4.3.1.1	<code>cldc_close</code>	39
4.3.1.2	<code>cldc_del</code>	39
4.3.1.3	<code>cldc_dirent_count</code>	39
4.3.1.4	<code>cldc_dirent_cur_fini</code>	39
4.3.1.5	<code>cldc_dirent_cur_init</code>	39
4.3.1.6	<code>cldc_dirent_first</code>	39
4.3.1.7	<code>cldc_dirent_name</code>	39
4.3.1.8	<code>cldc_dirent_next</code>	39
4.3.1.9	<code>cldc_end_sess</code>	39
4.3.1.10	<code>cldc_get</code>	39
4.3.1.11	<code>cldc_getaddr</code>	39
4.3.1.12	<code>cldc_init</code>	39
4.3.1.13	<code>cldc_kill_sess</code>	39
4.3.1.14	<code>cldc_levent_timer</code>	39
4.3.1.15	<code>cldc_lock</code>	39
4.3.1.16	<code>cldc_new_sess</code>	39
4.3.1.17	<code>cldc_nop</code>	39
4.3.1.18	<code>cldc_open</code>	39
4.3.1.19	<code>cldc_put</code>	39
4.3.1.20	<code>cldc_receive_pkt</code>	39
4.3.1.21	<code>cldc_saveaddr</code>	40
4.3.1.22	<code>cldc_udp_free</code>	40
4.3.1.23	<code>cldc_udp_new</code>	40
4.3.1.24	<code>cldc_udp_pkt_send</code>	40
4.3.1.25	<code>cldc_udp_receive_pkt</code>	40
4.3.1.26	<code>cldc_unlock</code>	40

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

cld_dirent_cur	5
cld_msg_ack_frag (ACK-FRAG message)	6
cld_msg_close (CLOSE 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)	26
cldc_pkt_info	27
cldc_session (Single CLD client session)	28
cldc_udp (A UDP implementation of the CLD client protocol)	30

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

include/ cld-private.h	31
include/ cld_msg.h	32
include/ cldc.h	37

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_ack_frag Struct Reference

ACK-FRAG message.

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

Data Fields

- struct [cld_msg_hdr](#) `hdr`
- uint64_t `seqid`
sequence id to ack

3.2.1 Detailed Description

ACK-FRAG message.

3.2.2 Field Documentation

3.2.2.1 struct `cld_msg_hdr cld_msg_ack_frag::hdr` [`read`]

3.2.2.2 uint64_t `cld_msg_ack_frag::seqid`

sequence id to ack

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

- `include/cld_msg.h`

3.3 cld_msg_close Struct Reference

CLOSE message.

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

Data Fields

- struct [cld_msg_hdr](#) `hdr`
- `uint64_t` `fh`
open file handle

3.3.1 Detailed Description

CLOSE message.

3.3.2 Field Documentation

3.3.2.1 `uint64_t cld_msg_close::fh`

open file handle

3.3.2.2 `struct cld_msg_hdr cld_msg_close::hdr` [read]

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

- `include/cld_msg.h`

3.4 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
- uint8_t [res](#) [6]

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

3.4.2.3 uint8_t [cld_msg_del::res](#)[6]

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.
- uint8_t `res` [4]

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]

3.5.2.4 uint8_t cld_msg_event::res[4]

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
- [uint8_t](#) `res` [4]

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 `uint8_t cld_msg_get_resp::res[4]`

3.7.2.5 `struct cld_msg_resp cld_msg_get_resp::resp` `[read]`

3.7.2.6 `uint32_t cld_msg_get_resp::size`

data size

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.
- [uint8_t](#) `res` [4]

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]

3.9.2.4 [uint8_t](#) `cld_msg_lock::res`[4]

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
- uint8_t [res](#) [6]

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

3.10.2.5 uint8_t cld_msg_open::res[6]

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
- [uint32_t](#) `data_size`
total size of data
- [uint8_t](#) `res` [4]

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 [uint8_t](#) `cld_msg_put::res`[4]

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
- uint32_t [flags](#)
CPF_xxx flags.
- uint8_t [res](#) [4]
- 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 uint32_t cld_packet::flags

CPF_xxx flags.

3.15.2.2 uint8_t cld_packet::magic[CLD_MAGIC_SZ]

magic number; constant

3.15.2.3 uint8_t cld_packet::res[4]

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`
 - `const 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 `const 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

- 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 unsigned short `cldc_host::port`

3.18.2.3 unsigned int `cldc_host::prio`

3.18.2.4 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 [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 [data_len](#)
- int [n_pkts](#)
- struct [cldc_pkt_info](#) * [pkt_info](#) [CLD_MAX_PKT_MSG]
- 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 `int cldc_msg::n_pkts`

3.19.2.9 `struct cldc_pkt_info* cldc_msg::pkt_info[CLD_MAX_PKT_MSG]` [**read**]

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

3.19.2.11 `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(* errlog)(int prio, const char *fmt,...)`

3.20.1 Detailed Description

application-supplied facilities

3.20.2 Field Documentation

3.20.2.1 `void(* cldc_ops::errlog)(int prio, const char *fmt,...)`

3.20.2.2 `void(* cldc_ops::event)(void *private, struct cldc_session *, struct cldc_fh *, uint32_t)`

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

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_pkt_info Struct Reference

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

Data Fields

- int [pkt_len](#)
- int [retries](#)
- struct [cld_packet](#) [pkt](#)
- uint8_t [data](#) [0]

3.21.1 Field Documentation

3.21.1.1 uint8_t cldc_pkt_info::data[0]

3.21.1.2 struct cld_packet cldc_pkt_info::pkt [read]

3.21.1.3 int cldc_pkt_info::pkt_len

3.21.1.4 int cldc_pkt_info::retries

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

- include/[cldc.h](#)

3.22 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](#))(int prio, const char *fmt,...)
- void * [private](#)
- uint8_t [addr](#) [64]
- size_t [addr_len](#)
- GArray * [fh](#)
- GList * [out_msg](#)
- time_t [msg_scan_time](#)
- 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](#)
- unsigned int [msg_buf_len](#)
- char [msg_buf](#) [CLD_MAX_MSG_SZ]

3.22.1 Detailed Description

a single CLD client session

3.22.2 Field Documentation

- 3.22.2.1 void(* cldc_session::act_log)(int prio, const char *fmt,...)
- 3.22.2.2 uint8_t cldc_session::addr[64]
- 3.22.2.3 size_t cldc_session::addr_len
- 3.22.2.4 bool cldc_session::confirmed
- 3.22.2.5 time_t cldc_session::expire_time
- 3.22.2.6 bool cldc_session::expired
- 3.22.2.7 GArray* cldc_session::fh
- 3.22.2.8 char cldc_session::msg_buf[CLD_MAX_MSG_SZ]
- 3.22.2.9 unsigned int cldc_session::msg_buf_len
- 3.22.2.10 time_t cldc_session::msg_scan_time
- 3.22.2.11 uint64_t cldc_session::next_seqid_in
- 3.22.2.12 uint64_t cldc_session::next_seqid_in_tr
- 3.22.2.13 uint64_t cldc_session::next_seqid_out
- 3.22.2.14 struct cldc_ops* cldc_session::ops [read]
- 3.22.2.15 GList* cldc_session::out_msg
- 3.22.2.16 void* cldc_session::private
- 3.22.2.17 char cldc_session::secret_key[CLD_MAX_SECRET_KEY]
- 3.22.2.18 uint8_t cldc_session::sid[CLD_SID_SZ]
- 3.22.2.19 char cldc_session::user[CLD_MAX_USERNAME]
- 3.22.2.20 bool cldc_session::verbose

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-private.h File Reference

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

```
#include <glib.h>
```

4.2 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_ack_frag](#)
ACK-FRAG 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_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_PKT_MSG_SZ` = 1024, `CLD_MAX_PKT_MSG` = 128, `CLD_MAX_MSG_SZ` = `CLD_MAX_PKT_MSG` * 1024 }
- enum `cld_msg_ops` {
`cmo_nop` = 0, `cmo_new_sess` = 1, `cmo_open` = 2, `cmo_get_meta` = 3,
`cmo_get` = 4, `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_ack_frag` = 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) }
available 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.

- enum `cld_packet_flags` { `CPF_FIRST` = (1 << 0), `CPF_LAST` = (1 << 1) }
CLD packet flags.

Functions

- unsigned long long `cld_sid2llu` (const uint8_t *sid)
- void `__cld_rand64` (void *p)

4.2.1 Define Documentation

4.2.1.1 `#define CLD_ALIGN8(n) ((8 - ((n) & 7)) & 7)`

4.2.1.2 `#define CLD_MSG_MAGIC "CLDc1msg"`

4.2.1.3 `#define CLD_PKT_MAGIC "CLDc1pkt"`

4.2.1.4 `#define SIDARG(sid) cld_sid2llu(sid)`

4.2.1.5 `#define SIDFMT "%016llx"`

4.2.2 Enumeration Type Documentation

4.2.2.1 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_PKT_MSG_SZ
CLD_MAX_PKT_MSG
CLD_MAX_MSG_SZ maximum total msg size, including all packets

4.2.2.2 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

4.2.2.3 enum cld_lock_flags

LOCK flags.

Enumerator:

CLF_SHARED a shared (read) lock

4.2.2.4 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_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_ack_frag ack partial msg

4.2.2.5 enum cld_open_modes

availble 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

4.2.2.6 enum cld_packet_flags

CLD packet flags.

Enumerator:

CPF_FIRST first fragment

CPF_LAST last fragment

4.2.2.7 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.

4.2.3 Function Documentation

4.2.3.1 void __cld_rand64 (void *p)

4.2.3.2 unsigned long long cld_sid2llu (const uint8_t *sid)

4.3 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_pkt_info](#)
- 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.
- void [cldc_init](#) (void)
- 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)(int prio, 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)(int prio, const char *fmt,...))

4.3.1 Function Documentation

- 4.3.1.1 `int cldc_close (struct cldc_fh *fh, const struct cldc_call_opts *copts)`
- 4.3.1.2 `int cldc_del (struct cldc_session *sess, const struct cldc_call_opts *copts, const char *pathname)`
- 4.3.1.3 `int cldc_dirent_count (const void *data, size_t data_len)`
- 4.3.1.4 `void cldc_dirent_cur_fini (struct cld_dirent_cur *dc)`
- 4.3.1.5 `void cldc_dirent_cur_init (struct cld_dirent_cur *dc, const void *buf, size_t buflen)`
- 4.3.1.6 `int cldc_dirent_first (struct cld_dirent_cur *dc)`
- 4.3.1.7 `char* cldc_dirent_name (struct cld_dirent_cur *dc)`
- 4.3.1.8 `int cldc_dirent_next (struct cld_dirent_cur *dc)`
- 4.3.1.9 `int cldc_end_sess (struct cldc_session *sess, const struct cldc_call_opts *copts)`
- 4.3.1.10 `int cldc_get (struct cldc_fh *fh, const struct cldc_call_opts *copts, bool metadata_only)`
- 4.3.1.11 `int cldc_getaddr (GList **host_list, const char *thishost, bool verbose, void(*)(int prio, const char *fmt,...) act_log)`
- 4.3.1.12 `void cldc_init (void)`
- 4.3.1.13 `void cldc_kill_sess (struct cldc_session *sess)`
- 4.3.1.14 `bool cldc_levent_timer (void *private, bool add, int(*)(struct cldc_session *, void *) cb, void *cb_private, time_t secs)`
- 4.3.1.15 `int cldc_lock (struct cldc_fh *fh, const struct cldc_call_opts *copts, uint32_t lock_flags, bool wait_for_lock)`
- 4.3.1.16 `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)`
- 4.3.1.17 `int cldc_nop (struct cldc_session *sess, const struct cldc_call_opts *copts)`
- 4.3.1.18 `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)`
- 4.3.1.19 `int cldc_put (struct cldc_fh *fh, const struct cldc_call_opts *copts, const void *data, size_t data_len)`
- 4.3.1.20 `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

- 4.3.1.21 `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(*) (int prio, const char *fmt,...) act_log)`
- 4.3.1.22 `void cldc_udp_free (struct cldc_udp * udp)`
- 4.3.1.23 `int cldc_udp_new (const char * hostname, int port, struct cldc_udp ** udp_out)`
- 4.3.1.24 `int cldc_udp_pkt_send (void * private, const void * addr, size_t addrlen, const void * buf, size_t buflen)`
- 4.3.1.25 `int cldc_udp_receive_pkt (struct cldc_udp * udp)`
- 4.3.1.26 `int cldc_unlock (struct cldc_fh * fh, const struct cldc_call_opts * copts)`

Index

- `__cld_rand64`
 - `cld_msg.h`, [36](#)
- `act_log`
 - `cldc_session`, [29](#)
- `addr`
 - `cldc_session`, [29](#)
 - `cldc_udp`, [30](#)
- `addr_len`
 - `cldc_session`, [29](#)
 - `cldc_udp`, [30](#)
- `buf`
 - `cldc_call_opts`, [22](#)
- `cb`
 - `cldc_call_opts`, [22](#)
 - `cldc_msg`, [25](#)
 - `cldc_udp`, [30](#)
- `cb_private`
 - `cldc_msg`, [25](#)
 - `cldc_udp`, [30](#)
- `CE_DELETED`
 - `cld_msg.h`, [34](#)
- `CE_LOCKED`
 - `cld_msg.h`, [34](#)
- `CE_MASTER_FAILOVER`
 - `cld_msg.h`, [34](#)
- `CE_SESS_FAILED`
 - `cld_msg.h`, [34](#)
- `CE_UPDATED`
 - `cld_msg.h`, [34](#)
- `CLD_INODE_NAME_MAX`
 - `cld_msg.h`, [34](#)
- `CLD_MAGIC_SZ`
 - `cld_msg.h`, [34](#)
- `CLD_MAX_MSG_SZ`
 - `cld_msg.h`, [34](#)
- `CLD_MAX_PKT_MSG`
 - `cld_msg.h`, [34](#)
- `CLD_MAX_PKT_MSG_SZ`
 - `cld_msg.h`, [34](#)
- `CLD_MAX_SECRET_KEY`
 - `cld_msg.h`, [34](#)
- `CLD_MAX_USERNAME`
 - `cld_msg.h`, [34](#)

- cmo_nop, 35
- cmo_not_master, 35
- cmo_open, 35
- cmo_ping, 35
- cmo_put, 35
- cmo_trylock, 35
- cmo_unlock, 35
- COM_ACL, 35
- COM_CREATE, 35
- COM_DIRECTORY, 35
- COM_EXCL, 35
- COM_LOCK, 35
- COM_READ, 35
- COM_WRITE, 35
- CPF_FIRST, 36
- CPF_LAST, 36
- CLD_SID_SZ
 - cld_msg.h, 34
- CLD_ALIGN8
 - cld_msg.h, 34
- cld_dirent_cur, 5
 - p, 5
 - tmp_len, 5
- cld_events
 - cld_msg.h, 34
- cld_lock_flags
 - cld_msg.h, 34
- cld_msg.h
 - __cld_rand64, 36
 - CLD_ALIGN8, 34
 - cld_events, 34
 - cld_lock_flags, 34
 - CLD_MSG_MAGIC, 34
 - cld_msg_ops, 35
 - cld_open_modes, 35
 - cld_packet_flags, 35
 - CLD_PKT_MAGIC, 34
 - cld_sid2llu, 36
 - cle_err_codes, 36
 - SIDARG, 34
 - SIDFMT, 34
- cld_msg_ack_frag, 6
 - hdr, 6
 - seqid, 6
- cld_msg_close, 7
 - fh, 7
 - hdr, 7
- cld_msg_del, 8
 - hdr, 8
 - name_len, 8
 - res, 8
- cld_msg_event, 9
 - events, 9
 - fh, 9
 - hdr, 9
 - res, 9
- cld_msg_get, 10
 - fh, 10
 - hdr, 10
- cld_msg_get_resp, 11
 - flags, 11
 - ino_len, 11
 - inum, 11
 - res, 11
 - resp, 12
 - size, 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
 - res, 14
- CLD_MSG_MAGIC
 - cld_msg.h, 34
- cld_msg_open, 15
 - events, 15
 - hdr, 15
 - mode, 15
 - name_len, 15
 - res, 15
- cld_msg_open_resp, 16
 - fh, 16
 - resp, 16
- cld_msg_ops
 - cld_msg.h, 35
- cld_msg_put, 17
 - data_size, 17
 - fh, 17
 - hdr, 17
 - res, 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, 35
- cld_packet, 20

- flags, 20
- magic, 20
- res, 20
- seqid, 20
- sid, 20
- user, 20
- cld_packet_flags
 - cld_msg.h, 35
- CLD_PKT_MAGIC
 - cld_msg.h, 34
- cld_sid2llu
 - cld_msg.h, 36
- cldc.h
 - cldc_close, 39
 - cldc_del, 39
 - cldc_dirent_count, 39
 - cldc_dirent_cur_fini, 39
 - cldc_dirent_cur_init, 39
 - cldc_dirent_first, 39
 - cldc_dirent_name, 39
 - cldc_dirent_next, 39
 - cldc_end_sess, 39
 - cldc_get, 39
 - cldc_getaddr, 39
 - cldc_init, 39
 - cldc_kill_sess, 39
 - cldc_levent_timer, 39
 - cldc_lock, 39
 - cldc_new_sess, 39
 - cldc_nop, 39
 - cldc_open, 39
 - cldc_put, 39
 - cldc_receive_pkt, 39
 - cldc_saveaddr, 40
 - cldc_udp_free, 40
 - cldc_udp_new, 40
 - cldc_udp_pkt_send, 40
 - cldc_udp_receive_pkt, 40
 - cldc_unlock, 40
- 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, 39
- cldc_del
 - cldc.h, 39
- cldc_dirent_count
 - cldc.h, 39
- cldc_dirent_cur_fini
 - cldc.h, 39
- cldc_dirent_cur_init
 - cldc.h, 39
- cldc_dirent_first
 - cldc.h, 39
- cldc_dirent_name
 - cldc.h, 39
- cldc_dirent_next
 - cldc.h, 39
- cldc_end_sess
 - cldc.h, 39
- cldc_fh, 23
 - fh_le, 23
 - sess, 23
 - valid, 23
- cldc_get
 - cldc.h, 39
- cldc_getaddr
 - cldc.h, 39
- cldc_host, 24
 - host, 24
 - port, 24
 - prio, 24
 - weight, 24
- cldc_init
 - cldc.h, 39
- cldc_kill_sess
 - cldc.h, 39
- cldc_levent_timer
 - cldc.h, 39
- cldc_lock
 - cldc.h, 39
- cldc_msg, 25
 - cb, 25
 - cb_private, 25
 - copts, 25
 - data, 25
 - data_len, 25
 - done, 25
 - expire_time, 25
 - n_pkts, 25
 - pkt_info, 25
 - sess, 25
 - xid, 25
- cldc_new_sess
 - cldc.h, 39
- cldc_nop
 - cldc.h, 39
- cldc_open
 - cldc.h, 39
- cldc_ops, 26
 - errlog, 26

- event, 26
- pkt_send, 26
- timer_ctl, 26
- cldc_pkt_info, 27
 - data, 27
 - pkt, 27
 - pkt_len, 27
 - retries, 27
- cldc_put
 - cldc.h, 39
- cldc_receive_pkt
 - cldc.h, 39
- cldc_saveaddr
 - cldc.h, 40
- cldc_session, 28
 - act_log, 29
 - addr, 29
 - addr_len, 29
 - confirmed, 29
 - expire_time, 29
 - expired, 29
 - fh, 29
 - msg_buf, 29
 - msg_buf_len, 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
 - user, 29
 - verbose, 29
- cldc_udp, 30
 - addr, 30
 - addr_len, 30
 - cb, 30
 - cb_private, 30
 - fd, 30
 - sess, 30
 - timer_ev, 30
- cldc_udp_free
 - cldc.h, 40
- cldc_udp_new
 - cldc.h, 40
- cldc_udp_pkt_send
 - cldc.h, 40
- cldc_udp_receive_pkt
 - cldc.h, 40
- cldc_unlock
 - cldc.h, 40
- CLE_BAD_PKT
 - cld_msg.h, 36
- CLE_DATA_INVALID
 - cld_msg.h, 36
- CLE_DB_ERR
 - cld_msg.h, 36
- CLE_DIR_NOTEMPTY
 - cld_msg.h, 36
- CLE_FH_INVALID
 - cld_msg.h, 36
- CLE_INODE_EXISTS
 - cld_msg.h, 36
- CLE_INODE_INVALID
 - cld_msg.h, 36
- CLE_INTERNAL_ERR
 - cld_msg.h, 36
- CLE_LOCK_CONFLICT
 - cld_msg.h, 36
- CLE_LOCK_INVALID
 - cld_msg.h, 36
- CLE_LOCK_PENDING
 - cld_msg.h, 36
- CLE_MODE_INVALID
 - cld_msg.h, 36
- CLE_NAME_INVALID
 - cld_msg.h, 36
- CLE_OK
 - cld_msg.h, 36
- CLE_OOM
 - cld_msg.h, 36
- CLE_SESS_EXISTS
 - cld_msg.h, 36
- CLE_SESS_INVALID
 - cld_msg.h, 36
- CLE_SIG_INVALID
 - cld_msg.h, 36
- CLE_TIMEOUT
 - cld_msg.h, 36
- cle_err_codes
 - cld_msg.h, 36
- CLF_SHARED
 - cld_msg.h, 35
- cmo_ack
 - cld_msg.h, 35
- cmo_ack_frag
 - cld_msg.h, 35
- cmo_close
 - cld_msg.h, 35
- cmo_del
 - cld_msg.h, 35
- cmo_end_sess
 - cld_msg.h, 35
- cmo_event
 - cld_msg.h, 35
- cmo_get

- cld_msg.h, 35
- cmo_get_meta
 - cld_msg.h, 35
- cmo_lock
 - cld_msg.h, 35
- cmo_new_sess
 - cld_msg.h, 35
- cmo_nop
 - cld_msg.h, 35
- cmo_not_master
 - cld_msg.h, 35
- cmo_open
 - cld_msg.h, 35
- cmo_ping
 - cld_msg.h, 35
- cmo_put
 - cld_msg.h, 35
- cmo_trylock
 - cld_msg.h, 35
- cmo_unlock
 - cld_msg.h, 35
- code
 - cld_msg_resp, 18
- COM_ACL
 - cld_msg.h, 35
- COM_CREATE
 - cld_msg.h, 35
- COM_DIRECTORY
 - cld_msg.h, 35
- COM_EXCL
 - cld_msg.h, 35
- COM_LOCK
 - cld_msg.h, 35
- COM_READ
 - cld_msg.h, 35
- COM_WRITE
 - cld_msg.h, 35
- confirmed
 - cldc_session, 29
- copts
 - cldc_msg, 25
- CPF_FIRST
 - cld_msg.h, 36
- CPF_LAST
 - cld_msg.h, 36
- data
 - cldc_msg, 25
 - cldc_pkt_info, 27
- data_len
 - cldc_msg, 25
- data_size
 - cld_msg_put, 17
- done
 - cldc_msg, 25
- errlog
 - cldc_ops, 26
- event
 - cldc_ops, 26
- events
 - cld_msg_event, 9
 - cld_msg_open, 15
- expire_time
 - cldc_msg, 25
 - cldc_session, 29
- expired
 - cldc_session, 29
- fd
 - cldc_udp, 30
- fh
 - cld_msg_close, 7
 - 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
 - cld_packet, 20
- get
 - cldc_call_opts, 22
- hdr
 - cld_msg_ack_frag, 6
 - cld_msg_close, 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-private.h, 31
- include/cld_msg.h, 32
- include/cldc.h, 37
- ino_len
 - cld_msg_get_resp, 11

- inode_name
 - cldc_call_opts, 22
- inum
 - cld_msg_get_resp, 11
- magic
 - cld_msg_hdr, 13
 - cld_packet, 20
- mode
 - cld_msg_open, 15
- msg_buf
 - cldc_session, 29
- msg_buf_len
 - cldc_session, 29
- msg_scan_time
 - cldc_session, 29
- n_pkts
 - cldc_msg, 25
- name_len
 - cld_msg_del, 8
 - cld_msg_open, 15
- 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_pkt_info, 27
- pkt_info
 - cldc_msg, 25
- pkt_len
 - cldc_pkt_info, 27
- pkt_send
 - cldc_ops, 26
- port
 - cldc_host, 24
- prio
 - cldc_host, 24
- private
 - cldc_call_opts, 22
 - cldc_session, 29
- res
 - cld_msg_del, 8
 - cld_msg_event, 9
 - cld_msg_get_resp, 11
 - cld_msg_lock, 14
 - cld_msg_open, 15
 - cld_msg_put, 17
 - cld_packet, 20
- resl
 - cld_msg_hdr, 13
- resp
 - cld_msg_get_resp, 12
 - cld_msg_open_resp, 16
 - cldc_call_opts, 22
- retries
 - cldc_pkt_info, 27
- rsv
 - cld_msg_resp, 18
- secret_key
 - cldc_session, 29
- seqid
 - cld_msg_ack_frag, 6
 - cld_packet, 20
- sess
 - cldc_fh, 23
 - cldc_msg, 25
 - cldc_udp, 30
- sid
 - cld_packet, 20
 - cldc_session, 29
- SIDARG
 - cld_msg.h, 34
- SIDFMT
 - cld_msg.h, 34
- size
 - cld_msg_get_resp, 12
 - cldc_call_opts, 22
- time_create
 - cld_msg_get_resp, 12
- time_modify
 - cld_msg_get_resp, 12
- timer_ctl
 - cldc_ops, 26
- timer_ev
 - cldc_udp, 30
- 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, [25](#)
- xid_in
 - cld_msg_resp, [18](#)