

CLD

0.1git

Generated by Doxygen 1.7.3

Wed Feb 9 2011 09:57:35



# 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	chunk_check_status Struct Reference . . . . .	5
3.1.1	Field Documentation . . . . .	5
3.1.1.1	count . . . . .	5
3.1.1.2	lastdone . . . . .	5
3.1.1.3	pad . . . . .	5
3.1.1.4	state . . . . .	5
3.2	chunksrv_req Struct Reference . . . . .	5
3.2.1	Field Documentation . . . . .	6
3.2.1.1	data_len . . . . .	6
3.2.1.2	flags . . . . .	6
3.2.1.3	key_len . . . . .	6
3.2.1.4	magic . . . . .	6
3.2.1.5	nonce . . . . .	6
3.2.1.6	op . . . . .	6
3.2.1.7	sig . . . . .	6
3.3	chunksrv_resp Struct Reference . . . . .	6
3.3.1	Field Documentation . . . . .	7
3.3.1.1	data_len . . . . .	7
3.3.1.2	hash . . . . .	7
3.3.1.3	magic . . . . .	7
3.3.1.4	nonce . . . . .	7
3.3.1.5	resp_code . . . . .	7
3.3.1.6	rsv1 . . . . .	7
3.4	chunksrv_resp_chkstat Struct Reference . . . . .	7
3.4.1	Field Documentation . . . . .	7
3.4.1.1	chkstat . . . . .	7
3.4.1.2	resp . . . . .	7
3.5	chunksrv_resp_get Struct Reference . . . . .	7
3.5.1	Field Documentation . . . . .	8
3.5.1.1	mtime . . . . .	8
3.5.1.2	resp . . . . .	8
3.6	cld_dirent_cur Struct Reference . . . . .	8

3.6.1	Field Documentation	8
3.6.1.1	p	8
3.6.1.2	tmp_len	8
3.7	cld_timer Struct Reference	8
3.7.1	Field Documentation	9
3.7.1.1	cb	9
3.7.1.2	expires	9
3.7.1.3	fired	9
3.7.1.4	name	9
3.7.1.5	on_list	9
3.7.1.6	userdata	9
3.8	cld_timer_list Struct Reference	9
3.8.1	Field Documentation	9
3.8.1.1	list	9
3.8.1.2	runmark	9
3.9	cldc_call_opts Struct Reference	9
3.9.1	Detailed Description	10
3.9.2	Field Documentation	10
3.9.2.1	cb	10
3.9.2.2	private	10
3.9.2.3	resp	10
3.10	cldc_fh Struct Reference	10
3.10.1	Detailed Description	10
3.10.2	Field Documentation	11
3.10.2.1	fh	11
3.10.2.2	sess	11
3.10.2.3	valid	11
3.11	cldc_host Struct Reference	11
3.11.1	Detailed Description	11
3.11.2	Field Documentation	11
3.11.2.1	host	11
3.11.2.2	port	11
3.11.2.3	prio	11
3.11.2.4	weight	11
3.12	cldc_msg Struct Reference	12
3.12.1	Detailed Description	12
3.12.2	Field Documentation	12
3.12.2.1	cb	12
3.12.2.2	cb_private	12
3.12.2.3	copts	12
3.12.2.4	done	12
3.12.2.5	expire_time	12
3.12.2.6	n_pkts	12
3.12.2.7	op	12
3.12.2.8	pkt_info	12
3.12.2.9	sess	12
3.12.2.10	xid	12
3.13	cldc_node_metadata Struct Reference	13
3.13.1	Field Documentation	13
3.13.1.1	flags	13

3.13.1.2	inode_name	13
3.13.1.3	inum	13
3.13.1.4	time_create	13
3.13.1.5	time_modify	13
3.13.1.6	vers	13
3.14	cldc_ops Struct Reference	13
3.14.1	Detailed Description	14
3.14.2	Field Documentation	14
3.14.2.1	event	14
3.14.2.2	pkt_send	14
3.14.2.3	timer_ctl	14
3.15	cldc_pkt_info Struct Reference	14
3.15.1	Field Documentation	15
3.15.1.1	data	15
3.15.1.2	hdr_len	15
3.15.1.3	pkt_len	15
3.15.1.4	retries	15
3.15.1.5	user	15
3.16	cldc_session Struct Reference	15
3.16.1	Detailed Description	16
3.16.2	Field Documentation	17
3.16.2.1	addr	17
3.16.2.2	addr_len	17
3.16.2.3	cfh	17
3.16.2.4	confirmed	17
3.16.2.5	expire_time	17
3.16.2.6	expired	17
3.16.2.7	inode_name_temp	17
3.16.2.8	log	17
3.16.2.9	msg_buf	17
3.16.2.10	msg_buf_len	17
3.16.2.11	msg_buf_op	17
3.16.2.12	msg_scan_time	17
3.16.2.13	next_seqid_in	17
3.16.2.14	next_seqid_in_tr	17
3.16.2.15	next_seqid_out	17
3.16.2.16	ops	17
3.16.2.17	out_msg	17
3.16.2.18	payload	17
3.16.2.19	private	17
3.16.2.20	secret_key	17
3.16.2.21	sid	17
3.16.2.22	user	17
3.17	cldc_udp Struct Reference	18
3.17.1	Detailed Description	18
3.17.2	Field Documentation	18
3.17.2.1	addr	18
3.17.2.2	addr_len	18
3.17.2.3	cb	18
3.17.2.4	cb_private	18

3.17.2.5	fd	18
3.17.2.6	sess	18
3.18	hail_log Struct Reference	18
3.18.1	Field Documentation	19
3.18.1.1	debug	19
3.18.1.2	func	19
3.18.1.3	verbose	19
3.19	hstor_blist Struct Reference	19
3.19.1	Field Documentation	19
3.19.1.1	list	19
3.19.1.2	own_id	19
3.19.1.3	own_name	19
3.20	hstor_bucket Struct Reference	19
3.20.1	Field Documentation	20
3.20.1.1	name	20
3.20.1.2	time_create	20
3.21	hstor_client Struct Reference	20
3.21.1	Field Documentation	20
3.21.1.1	acc	20
3.21.1.2	curl	20
3.21.1.3	host	20
3.21.1.4	key	20
3.21.1.5	user	20
3.21.1.6	verbose	20
3.22	hstor_keylist Struct Reference	21
3.22.1	Field Documentation	21
3.22.1.1	common_pfx	21
3.22.1.2	contents	21
3.22.1.3	delim	21
3.22.1.4	marker	21
3.22.1.5	max_keys	21
3.22.1.6	name	21
3.22.1.7	prefix	21
3.22.1.8	trunc	21
3.23	hstor_object Struct Reference	21
3.23.1	Field Documentation	22
3.23.1.1	etag	22
3.23.1.2	key	22
3.23.1.3	own_id	22
3.23.1.4	own_name	22
3.23.1.5	size	22
3.23.1.6	storage	22
3.23.1.7	time_mod	22
3.24	http_hdr Struct Reference	22
3.24.1	Field Documentation	22
3.24.1.1	key	22
3.24.1.2	val	22
3.25	http_req Struct Reference	23
3.25.1	Field Documentation	23
3.25.1.1	hdr	23

3.25.1.2	major	23
3.25.1.3	method	23
3.25.1.4	minor	23
3.25.1.5	n_hdr	23
3.25.1.6	orig_path	23
3.25.1.7	uri	23
3.26	http_uri Struct Reference	23
3.26.1	Field Documentation	24
3.26.1.1	fragment	24
3.26.1.2	fragment_len	24
3.26.1.3	hostname	24
3.26.1.4	hostname_len	24
3.26.1.5	path	24
3.26.1.6	path_len	24
3.26.1.7	port	24
3.26.1.8	query	24
3.26.1.9	query_len	24
3.26.1.10	scheme	24
3.26.1.11	scheme_len	24
3.26.1.12	userinfo	24
3.26.1.13	userinfo_len	24
3.27	list_head Struct Reference	24
3.27.1	Field Documentation	25
3.27.1.1	next	25
3.27.1.2	prev	25
3.28	ncld_fh Struct Reference	25
3.28.1	Field Documentation	26
3.28.1.1	errc	26
3.28.1.2	event_arg	26
3.28.1.3	event_func	26
3.28.1.4	event_mask	26
3.28.1.5	fh	26
3.28.1.6	is_open	26
3.28.1.7	nios	26
3.28.1.8	sess	26
3.29	ncld_read Struct Reference	26
3.29.1	Field Documentation	27
3.29.1.1	errc	27
3.29.1.2	fh	27
3.29.1.3	is_done	27
3.29.1.4	length	27
3.29.1.5	meta	27
3.29.1.6	ptr	27
3.30	ncld_sess Struct Reference	27
3.30.1	Field Documentation	28
3.30.1.1	cond	28
3.30.1.2	errc	28
3.30.1.3	event	28
3.30.1.4	event_arg	28
3.30.1.5	handles	28

3.30.1.6	host	28
3.30.1.7	is_up	28
3.30.1.8	mutex	28
3.30.1.9	open_done	28
3.30.1.10	port	28
3.30.1.11	thread	28
3.30.1.12	tlst	28
3.30.1.13	to_thread	28
3.30.1.14	udp	28
3.30.1.15	udp_timer	28
3.31	objcache Struct Reference	28
3.31.1	Field Documentation	29
3.31.1.1	lock	29
3.31.1.2	table	29
3.32	objcache_entry Struct Reference	29
3.32.1	Field Documentation	29
3.32.1.1	flags	29
3.32.1.2	hash	29
3.32.1.3	ref	29
3.33	st_client Struct Reference	29
3.33.1	Field Documentation	30
3.33.1.1	fd	30
3.33.1.2	host	30
3.33.1.3	key	30
3.33.1.4	req_buf	30
3.33.1.5	ssl	30
3.33.1.6	ssl_ctx	30
3.33.1.7	user	30
3.33.1.8	verbose	30
3.34	st_keylist Struct Reference	30
3.34.1	Field Documentation	30
3.34.1.1	contents	30
3.34.1.2	name	30
3.35	st_object Struct Reference	31
3.35.1	Field Documentation	31
3.35.1.1	etag	31
3.35.1.2	name	31
3.35.1.3	owner	31
3.35.1.4	size	31
3.35.1.5	time_mod	31
<b>4</b>	<b>File Documentation</b>	<b>33</b>
4.1	include/chunk-private.h File Reference	33
4.1.1	Define Documentation	33
4.1.1.1	BAD_TPATH_FMT	33
4.1.1.2	MDB_TPATH_FMT	33
4.1.1.3	PREFIX_LEN	33
4.2	include/chunk_msg.h File Reference	33
4.2.1	Define Documentation	34
4.2.1.1	CHUNKD_MAGIC	34

4.2.2	Enumeration Type Documentation	34
4.2.2.1	"@0	34
4.2.2.2	chunk_check_state	35
4.2.2.3	chunk_errcode	35
4.2.2.4	chunk_flags	35
4.2.2.5	chunksrv_ops	35
4.3	include/chunkc.h File Reference	36
4.3.1	Function Documentation	39
4.3.1.1	stc_check_start	39
4.3.1.2	stc_check_status	39
4.3.1.3	stc_cp	39
4.3.1.4	stc_del	39
4.3.1.5	stc_free	39
4.3.1.6	stc_free_keylist	39
4.3.1.7	stc_free_object	39
4.3.1.8	stc_get	39
4.3.1.9	stc_get_inline	39
4.3.1.10	stc_get_recv	39
4.3.1.11	stc_get_start	39
4.3.1.12	stc_init	39
4.3.1.13	stc_keys	39
4.3.1.14	stc_new	39
4.3.1.15	stc_ping	39
4.3.1.16	stc_put	39
4.3.1.17	stc_put_inline	39
4.3.1.18	stc_put_send	39
4.3.1.19	stc_put_start	39
4.3.1.20	stc_put_sync	39
4.3.1.21	stc_readport	39
4.3.1.22	stc_table_open	39
4.4	include/chunksrv.h File Reference	39
4.4.1	Function Documentation	40
4.4.1.1	chreq_sign	40
4.4.1.2	req_len	40
4.5	include/cld-private.h File Reference	40
4.6	include/cld_common.h File Reference	40
4.6.1	Define Documentation	41
4.6.1.1	CLD_ALIGN8	41
4.6.1.2	CLD_PKT_FTR_LEN	41
4.6.1.3	PKT_HDR_TO_STR_SCRATCH_LEN	41
4.6.1.4	SIDARG	41
4.6.1.5	SIDFMT	41
4.6.2	Function Documentation	41
4.6.2.1	__attribute__	41
4.6.2.2	__cld_dump_buf	42
4.6.2.3	cld_authcheck	42
4.6.2.4	cld_authsign	42
4.6.2.5	cld_errstr	42
4.6.2.6	cld_opstr	42
4.6.2.7	cld_pkt_hdr_to_str	42

4.6.2.8	cld_rand64	42
4.6.2.9	cld_readport	42
4.6.2.10	cld_sid2llu	42
4.6.2.11	cld_timer_add	42
4.6.2.12	cld_timer_del	42
4.6.2.13	cld_timers_run	42
4.7	include/cldc.h File Reference	42
4.7.1	Function Documentation	46
4.7.1.1	cldc_close	46
4.7.1.2	cldc_copts_get_data	46
4.7.1.3	cldc_copts_get_metadata	46
4.7.1.4	cldc_del	46
4.7.1.5	cldc_dirent_count	46
4.7.1.6	cldc_dirent_cur_fini	46
4.7.1.7	cldc_dirent_cur_init	46
4.7.1.8	cldc_dirent_first	46
4.7.1.9	cldc_dirent_name	46
4.7.1.10	cldc_dirent_next	46
4.7.1.11	cldc_end_sess	46
4.7.1.12	cldc_get	46
4.7.1.13	cldc_getaddr	46
4.7.1.14	cldc_init	46
4.7.1.15	cldc_kill_sess	46
4.7.1.16	cldc_lock	46
4.7.1.17	cldc_new_sess	46
4.7.1.18	cldc_nop	46
4.7.1.19	cldc_open	46
4.7.1.20	cldc_put	46
4.7.1.21	cldc_receive_pkt	46
4.7.1.22	cldc_saveaddr	47
4.7.1.23	cldc_udp_free	47
4.7.1.24	cldc_udp_new	47
4.7.1.25	cldc_udp_pkt_send	47
4.7.1.26	cldc_udp_receive_pkt	47
4.7.1.27	cldc_unlock	47
4.8	include/elist.h File Reference	47
4.8.1	Define Documentation	48
4.8.1.1	INIT_LIST_HEAD	48
4.8.1.2	list_entry	48
4.8.1.3	list_for_each	48
4.8.1.4	list_for_each_entry	49
4.8.1.5	list_for_each_entry_continue	49
4.8.1.6	list_for_each_entry_safe	49
4.8.1.7	list_for_each_prev	49
4.8.1.8	list_for_each_safe	50
4.8.1.9	LIST_HEAD	50
4.8.1.10	LIST_HEAD_INIT	50
4.9	include/hail_log.h File Reference	50
4.9.1	Define Documentation	51
4.9.1.1	ATTR_PRINTF	51

4.9.1.2	HAIL_CRIT	51
4.9.1.3	HAIL_DEBUG	51
4.9.1.4	HAIL_ERR	51
4.9.1.5	HAIL_INFO	51
4.9.1.6	HAIL_VERBOSE	51
4.9.1.7	HAIL_WARN	51
4.10	include/hail_private.h File Reference	51
4.10.1	Function Documentation	52
4.10.1.1	xdr_sizeof	52
4.11	include/hstor.h File Reference	52
4.11.1	Define Documentation	54
4.11.1.1	ARRAY_SIZE	54
4.11.1.2	PATH_ESCAPE_MASK	54
4.11.1.3	QUERY_ESCAPE_MASK	54
4.11.2	Enumeration Type Documentation	54
4.11.2.1	"@1	54
4.11.2.2	ReqACLC	54
4.11.2.3	ReqQ	54
4.11.3	Function Documentation	56
4.11.3.1	hreq_acl_canned	56
4.11.3.2	hreq_free	56
4.11.3.3	hreq_hdr	56
4.11.3.4	hreq_hdr_push	56
4.11.3.5	hreq_is_query	56
4.11.3.6	hreq_query	56
4.11.3.7	hreq_sign	56
4.11.3.8	hstor_add_bucket	56
4.11.3.9	hstor_del	56
4.11.3.10	hstor_del_bucket	56
4.11.3.11	hstor_free	56
4.11.3.12	hstor_free_blist	56
4.11.3.13	hstor_free_bucket	56
4.11.3.14	hstor_free_keylist	56
4.11.3.15	hstor_free_object	56
4.11.3.16	hstor_get	56
4.11.3.17	hstor_get_inline	56
4.11.3.18	hstor_keys	56
4.11.3.19	hstor_list_buckets	56
4.11.3.20	hstor_new	56
4.11.3.21	hstor_put	56
4.11.3.22	hstor_put_inline	56
4.11.3.23	huri_field_escape	56
4.11.3.24	huri_field_unescape	56
4.11.3.25	huri_parse	56
4.11.3.26	hutil_str2time	56
4.11.3.27	hutil_time2str	56
4.12	include/ncld.h File Reference	56
4.12.1	Function Documentation	58
4.12.1.1	ncld_close	58
4.12.1.2	ncld_del	58

4.12.1.3	<code>ncld_get</code>	58
4.12.1.4	<code>ncld_get_meta</code>	58
4.12.1.5	<code>ncld_init</code>	58
4.12.1.6	<code>ncld_open</code>	58
4.12.1.7	<code>ncld_qlock</code>	58
4.12.1.8	<code>ncld_read_free</code>	58
4.12.1.9	<code>ncld_sess_close</code>	58
4.12.1.10	<code>ncld_sess_open</code>	58
4.12.1.11	<code>ncld_trylock</code>	58
4.12.1.12	<code>ncld_unlock</code>	58
4.12.1.13	<code>ncld_write</code>	58
4.13	<code>include/objcache.h</code> File Reference	58
4.13.1	Define Documentation	59
4.13.1.1	<code>objcache_get</code>	59
4.13.1.2	<code>objcache_get_dirty</code>	59
4.13.1.3	<code>OC_F_DIRTY</code>	59
4.13.2	Function Documentation	59
4.13.2.1	<code>__objcache_get</code>	59
4.13.2.2	<code>objcache_count</code>	59
4.13.2.3	<code>objcache_fini</code>	59
4.13.2.4	<code>objcache_init</code>	59
4.13.2.5	<code>objcache_put</code>	59
4.13.2.6	<code>objcache_test_dirty</code>	59

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">chunk_check_status</a>	5
<a href="#">chunksrv_req</a>	5
<a href="#">chunksrv_resp</a>	6
<a href="#">chunksrv_resp_chkstat</a>	7
<a href="#">chunksrv_resp_get</a>	7
<a href="#">cld_dirent_cur</a>	8
<a href="#">cld_timer</a>	8
<a href="#">cld_timer_list</a>	9
<a href="#">cldc_call_opts</a> (Per-operation application options )	9
<a href="#">cldc_fh</a> (Open file handle associated with a session )	10
<a href="#">cldc_host</a> (Information for a single CLD server host )	11
<a href="#">cldc_msg</a> (Outgoing message, from client to server )	12
<a href="#">cldc_node_metadata</a>	13
<a href="#">cldc_ops</a> (Application-supplied facilities )	13
<a href="#">cldc_pkt_info</a>	14
<a href="#">cldc_session</a> (Single CLD client session )	15
<a href="#">cldc_udp</a> (A UDP implementation of the CLD client protocol )	18
<a href="#">hail_log</a>	18
<a href="#">hstor_blist</a>	19
<a href="#">hstor_bucket</a>	19
<a href="#">hstor_client</a>	20
<a href="#">hstor_keylist</a>	21
<a href="#">hstor_object</a>	21
<a href="#">http_hdr</a>	22
<a href="#">http_req</a>	23
<a href="#">http_uri</a>	23
<a href="#">list_head</a>	24
<a href="#">ncld_fh</a>	25
<a href="#">ncld_read</a>	26

<a href="#">nclد_sess</a>	<a href="#">27</a>
<a href="#">objcache</a>	<a href="#">28</a>
<a href="#">objcache_entry</a>	<a href="#">29</a>
<a href="#">st_client</a>	<a href="#">29</a>
<a href="#">st_keylist</a>	<a href="#">30</a>
<a href="#">st_object</a>	<a href="#">31</a>

## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

include/ <a href="#">chunk-private.h</a> . . . . .	33
include/ <a href="#">chunk_msg.h</a> . . . . .	33
include/ <a href="#">chunkc.h</a> . . . . .	36
include/ <a href="#">chunksrv.h</a> . . . . .	39
include/ <a href="#">cld-private.h</a> . . . . .	40
include/ <a href="#">cld_common.h</a> . . . . .	40
include/ <a href="#">cldc.h</a> . . . . .	42
include/ <a href="#">elist.h</a> . . . . .	47
include/ <a href="#">hail_log.h</a> . . . . .	50
include/ <a href="#">hail_private.h</a> . . . . .	51
include/ <a href="#">hstor.h</a> . . . . .	52
include/ <a href="#">ncld.h</a> . . . . .	56
include/ <a href="#">objcache.h</a> . . . . .	58



## Chapter 3

# Data Structure Documentation

### 3.1 `chunk_check_status` Struct Reference

```
#include <chunk_msg.h>
```

#### Data Fields

- `uint8_t` [state](#)
- `uint8_t` [pad](#) [3]
- `uint32_t` [count](#)
- `uint64_t` [lastdone](#)

#### 3.1.1 Field Documentation

3.1.1.1 `uint32_t chunk_check_status::count`

3.1.1.2 `uint64_t chunk_check_status::lastdone`

3.1.1.3 `uint8_t chunk_check_status::pad`[3]

3.1.1.4 `uint8_t chunk_check_status::state`

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

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

### 3.2 `chunksrv_req` Struct Reference

```
#include <chunk_msg.h>
```

### Data Fields

- uint8\_t [magic](#) [CHD\_MAGIC\_SZ]
- uint8\_t [op](#)
- uint8\_t [flags](#)
- uint16\_t [key\\_len](#)
- uint32\_t [nonce](#)
- uint64\_t [data\\_len](#)
- char [sig](#) [CHD\_SIG\_SZ]

### 3.2.1 Field Documentation

3.2.1.1 uint64\_t [chunksrv\\_req::data\\_len](#)

3.2.1.2 uint8\_t [chunksrv\\_req::flags](#)

3.2.1.3 uint16\_t [chunksrv\\_req::key\\_len](#)

3.2.1.4 uint8\_t [chunksrv\\_req::magic](#)[CHD\_MAGIC\_SZ]

3.2.1.5 uint32\_t [chunksrv\\_req::nonce](#)

3.2.1.6 uint8\_t [chunksrv\\_req::op](#)

3.2.1.7 char [chunksrv\\_req::sig](#)[CHD\_SIG\_SZ]

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

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

## 3.3 chunksrv\_resp Struct Reference

```
#include <chunk_msg.h>
```

### Data Fields

- uint8\_t [magic](#) [CHD\_MAGIC\_SZ]
- uint8\_t [resp\\_code](#)
- uint8\_t [rsv1](#) [3]
- uint32\_t [nonce](#)
- uint64\_t [data\\_len](#)
- unsigned char [hash](#) [CHD\_CSUM\_SZ]

### 3.3.1 Field Documentation

3.3.1.1 `uint64_t chunksrv_resp::data_len`

3.3.1.2 `unsigned char chunksrv_resp::hash[CHD_CSUM_SZ]`

3.3.1.3 `uint8_t chunksrv_resp::magic[CHD_MAGIC_SZ]`

3.3.1.4 `uint32_t chunksrv_resp::nonce`

3.3.1.5 `uint8_t chunksrv_resp::resp_code`

3.3.1.6 `uint8_t chunksrv_resp::rsv1[3]`

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

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

## 3.4 chunksrv\_resp\_chkstat Struct Reference

```
#include <chunk_msg.h>
```

### Data Fields

- struct [chunksrv\\_resp](#) `resp`
- struct [chunk\\_check\\_status](#) `chkstat`

### 3.4.1 Field Documentation

3.4.1.1 `struct chunk_check_status chunksrv_resp_chkstat::chkstat`

3.4.1.2 `struct chunksrv_resp chunksrv_resp_chkstat::resp`

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

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

## 3.5 chunksrv\_resp\_get Struct Reference

```
#include <chunk_msg.h>
```

### Data Fields

- struct [chunksrv\\_resp](#) `resp`

- [uint64\\_t mtime](#)

### 3.5.1 Field Documentation

#### 3.5.1.1 [uint64\\_t chunksrv\\_resp\\_get::mtime](#)

#### 3.5.1.2 [struct chunksrv\\_resp chunksrv\\_resp\\_get::resp](#)

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

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

## 3.6 [cld\\_dirent\\_cur](#) Struct Reference

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

### Data Fields

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

### 3.6.1 Field Documentation

#### 3.6.1.1 [const void\\* cld\\_dirent\\_cur::p](#)

#### 3.6.1.2 [size\\_t cld\\_dirent\\_cur::tmp\\_len](#)

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

- [include/cldc.h](#)

## 3.7 [cld\\_timer](#) Struct Reference

```
#include <cld_common.h>
```

### Data Fields

- [bool fired](#)
- [bool on\\_list](#)
- [void\(\\* cb\)\(struct cld\\_timer \\*\)](#)
- [void \\* userdata](#)
- [time\\_t expires](#)
- [char name \[32\]](#)

### 3.7.1 Field Documentation

3.7.1.1 `void(* cld_timer::cb)(struct cld_timer *)`

3.7.1.2 `time_t cld_timer::expires`

3.7.1.3 `bool cld_timer::fired`

3.7.1.4 `char cld_timer::name[32]`

3.7.1.5 `bool cld_timer::on_list`

3.7.1.6 `void* cld_timer::userdata`

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

- `include/cld_common.h`

## 3.8 cld\_timer\_list Struct Reference

```
#include <cld_common.h>
```

### Data Fields

- `GList * list`
- `time_t runmark`

### 3.8.1 Field Documentation

3.8.1.1 `GList* cld_timer_list::list`

3.8.1.2 `time_t cld_timer_list::runmark`

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

- `include/cld_common.h`

## 3.9 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`
- `struct cld\_msg\_get\_resp resp`

### 3.9.1 Detailed Description

per-operation application options

### 3.9.2 Field Documentation

**3.9.2.1** `int(* cldc\_call\_opts::cb)(struct cldc\_call\_opts *, enum cle\_err\_codes)`

**3.9.2.2** `void* cldc\_call\_opts::private`

**3.9.2.3** `struct cld\_msg\_get\_resp cldc\_call\_opts::resp`

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

- `include/cldc.h`

## 3.10 [cldc\\_fh](#) Struct Reference

an open file handle associated with a session

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

## Data Fields

- `uint64_t fh`
- `struct cldc\_session * sess`
- `bool valid`

### 3.10.1 Detailed Description

an open file handle associated with a session

### 3.10.2 Field Documentation

3.10.2.1 `uint64_t cldc_fh::fh`

3.10.2.2 `struct cldc_session* cldc_fh::sess`

3.10.2.3 `bool cldc_fh::valid`

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

- `include/cldc.h`

## 3.11 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`

### 3.11.1 Detailed Description

Information for a single CLD server host.

### 3.11.2 Field Documentation

3.11.2.1 `char* cldc_host::host`

3.11.2.2 `unsigned short cldc_host::port`

3.11.2.3 `unsigned int cldc_host::prio`

3.11.2.4 `unsigned int cldc_host::weight`

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

- `include/cldc.h`

## 3.12 cldc\_msg Struct Reference

an outgoing message, from client to server

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

### Data Fields

- uint64\_t [xid](#)
- enum cld\_msg\_op [op](#)
- struct [cldc\\_session](#) \* [sess](#)
- ssize\_t(\* [cb](#))(struct [cldc\\_msg](#) \*, const void \*, size\_t, enum cle\_err\_codes)
- void \* [cb\\_private](#)
- struct [cldc\\_call\\_opts](#) [copts](#)
- bool [done](#)
- time\_t [expire\\_time](#)
- int [n\\_pkts](#)
- struct [cldc\\_pkt\\_info](#) \* [pkt\\_info](#) [0]

### 3.12.1 Detailed Description

an outgoing message, from client to server

### 3.12.2 Field Documentation

**3.12.2.1** `ssize_t(* cldc_msg::cb)(struct cldc_msg *, const void *, size_t, enum cle_err_codes)`

**3.12.2.2** `void* cldc_msg::cb_private`

**3.12.2.3** `struct cldc_call_opts cldc_msg::copts`

**3.12.2.4** `bool cldc_msg::done`

**3.12.2.5** `time_t cldc_msg::expire_time`

**3.12.2.6** `int cldc_msg::n_pkts`

**3.12.2.7** `enum cld_msg_op cldc_msg::op`

**3.12.2.8** `struct cldc_pkt_info* cldc_msg::pkt_info[0]`

**3.12.2.9** `struct cldc_session* cldc_msg::sess`

**3.12.2.10** `uint64_t cldc_msg::xid`

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

- [include/cldc.h](#)

## 3.13 cldc\_node\_metadata Struct Reference

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

### Data Fields

- [quad\\_t inum](#)
- [quad\\_t vers](#)
- [quad\\_t time\\_create](#)
- [quad\\_t time\\_modify](#)
- [int flags](#)
- [const char \\* inode\\_name](#)

#### 3.13.1 Field Documentation

3.13.1.1 [int cldc\\_node\\_metadata::flags](#)

3.13.1.2 [const char\\* cldc\\_node\\_metadata::inode\\_name](#)

3.13.1.3 [quad\\_t cldc\\_node\\_metadata::inum](#)

3.13.1.4 [quad\\_t cldc\\_node\\_metadata::time\\_create](#)

3.13.1.5 [quad\\_t cldc\\_node\\_metadata::time\\_modify](#)

3.13.1.6 [quad\\_t cldc\\_node\\_metadata::vers](#)

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

- [include/cldc.h](#)

## 3.14 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)`

### 3.14.1 Detailed Description

application-supplied facilities

### 3.14.2 Field Documentation

- 3.14.2.1 `void(* cldc\_ops::event )(void *private, struct cldc\_session *, struct cldc\_fh *, uint32_t)`
- 3.14.2.2 `int(* cldc\_ops::pkt\_send )(void *private, const void *addr, size_t addrlen, const void *buf, size_t buflen)`
- 3.14.2.3 `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`

## 3.15 [cldc\\_pkt\\_info](#) Struct Reference

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

### Data Fields

- `int pkt\_len`
- `int hdr\_len`
- `int retries`
- `char user [CLD_MAX_USERNAME]`
- `char data [0]`

### 3.15.1 Field Documentation

3.15.1.1 char cldc\_pkt\_info::data[0]

3.15.1.2 int cldc\_pkt\_info::hdr\_len

3.15.1.3 int cldc\_pkt\_info::pkt\_len

3.15.1.4 int cldc\_pkt\_info::retries

3.15.1.5 char cldc\_pkt\_info::user[CLD\_MAX\_USERNAME]

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

- include/cldc.h

## 3.16 cldc\_session Struct Reference

a single CLD client session

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

### Data Fields

- uint8\_t sid [CLD\_SID\_SZ]
- struct cldc\_ops \* ops
- struct hail\_log log
- void \* private
- uint8\_t addr [64]
- size\_t addr\_len
- GList \* cfh
- 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
- enum cld\_msg\_op msg\_buf\_op
- unsigned int msg\_buf\_len
- char msg\_buf [CLD\_MAX\_MSG\_SZ]
- char payload [CLD\_MAX\_PAYLOAD\_SZ]
- char inode\_name\_temp [CLD\_INODE\_NAME\_MAX]

### 3.16.1 Detailed Description

a single CLD client session

### 3.16.2 Field Documentation

- 3.16.2.1 `uint8_t cldc_session::addr[64]`
- 3.16.2.2 `size_t cldc_session::addr_len`
- 3.16.2.3 `GList* cldc_session::cfh`
- 3.16.2.4 `bool cldc_session::confirmed`
- 3.16.2.5 `time_t cldc_session::expire_time`
- 3.16.2.6 `bool cldc_session::expired`
- 3.16.2.7 `char cldc_session::inode_name_temp[CLD_INODE_NAME_MAX]`
- 3.16.2.8 `struct hail_log cldc_session::log`
- 3.16.2.9 `char cldc_session::msg_buf[CLD_MAX_MSG_SZ]`
- 3.16.2.10 `unsigned int cldc_session::msg_buf_len`
- 3.16.2.11 `enum cld_msg_op cldc_session::msg_buf_op`
- 3.16.2.12 `time_t cldc_session::msg_scan_time`
- 3.16.2.13 `uint64_t cldc_session::next_seqid_in`
- 3.16.2.14 `uint64_t cldc_session::next_seqid_in_tr`
- 3.16.2.15 `uint64_t cldc_session::next_seqid_out`
- 3.16.2.16 `struct cldc_ops* cldc_session::ops`
- 3.16.2.17 `GList* cldc_session::out_msg`
- 3.16.2.18 `char cldc_session::payload[CLD_MAX_PAYLOAD_SZ]`
- 3.16.2.19 `void* cldc_session::private`
- 3.16.2.20 `char cldc_session::secret_key[CLD_MAX_SECRET_KEY]`
- 3.16.2.21 `uint8_t cldc_session::sid[CLD_SID_SZ]`
- 3.16.2.22 `char cldc_session::user[CLD_MAX_USERNAME]`

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

- [include/cldc.h](#)

## 3.17 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 cldc_session * sess`
- `int(* cb)(struct cldc_session *, void *)`
- `void * cb_private`

### 3.17.1 Detailed Description

A UDP implementation of the CLD client protocol.

### 3.17.2 Field Documentation

**3.17.2.1** `uint8_t cldc_udp::addr[64]`

**3.17.2.2** `size_t cldc_udp::addr_len`

**3.17.2.3** `int(* cldc_udp::cb)(struct cldc_session *, void *)`

**3.17.2.4** `void* cldc_udp::cb_private`

**3.17.2.5** `int cldc_udp::fd`

**3.17.2.6** `struct cldc_session* cldc_udp::sess`

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

- `include/cldc.h`

## 3.18 hail\_log Struct Reference

```
#include <hail_log.h>
```

### Data Fields

- `void(* func)(int prio, const char *fmt,...) ATTR_PRINTF(2`

- void(\*) boo [debug](#) )
- bool [verbose](#)

### 3.18.1 Field Documentation

3.18.1.1 void(\*) boo `hail_log::debug`)

3.18.1.2 void(\*) `hail_log::func`)(int prio, const char \*fmt,...) ATTR\_PRINTF(2

3.18.1.3 bool `hail_log::verbose`

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

- include/[hail\\_log.h](#)

## 3.19 hstor\_blist Struct Reference

```
#include <hstor.h>
```

### Data Fields

- char \* [own\\_id](#)
- char \* [own\\_name](#)
- GList \* [list](#)

### 3.19.1 Field Documentation

3.19.1.1 GList\* `hstor_blist::list`

3.19.1.2 char\* `hstor_blist::own_id`

3.19.1.3 char\* `hstor_blist::own_name`

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

- include/[hstor.h](#)

## 3.20 hstor\_bucket Struct Reference

```
#include <hstor.h>
```

### Data Fields

- char \* [name](#)
- char \* [time\\_create](#)

#### 3.20.1 Field Documentation

3.20.1.1 char\* hstor\_bucket::name

3.20.1.2 char\* hstor\_bucket::time\_create

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

- include/[hstor.h](#)

### 3.21 hstor\_client Struct Reference

```
#include <hstor.h>
```

### Data Fields

- CURL \* [curl](#)
- char \* [acc](#)
- char \* [host](#)
- char \* [user](#)
- char \* [key](#)
- bool [verbose](#)

#### 3.21.1 Field Documentation

3.21.1.1 char\* hstor\_client::acc

3.21.1.2 CURL\* hstor\_client::curl

3.21.1.3 char\* hstor\_client::host

3.21.1.4 char\* hstor\_client::key

3.21.1.5 char\* hstor\_client::user

3.21.1.6 bool hstor\_client::verbose

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

- include/[hstor.h](#)

## 3.22 hstor\_keylist Struct Reference

```
#include <hstor.h>
```

### Data Fields

- char \* [name](#)
- char \* [prefix](#)
- char \* [marker](#)
- char \* [delim](#)
- unsigned int [max\\_keys](#)
- bool [trunc](#)
- GList \* [contents](#)
- GList \* [common\\_pfx](#)

### 3.22.1 Field Documentation

3.22.1.1 GList\* hstor\_keylist::common\_pfx

3.22.1.2 GList\* hstor\_keylist::contents

3.22.1.3 char\* hstor\_keylist::delim

3.22.1.4 char\* hstor\_keylist::marker

3.22.1.5 unsigned int hstor\_keylist::max\_keys

3.22.1.6 char\* hstor\_keylist::name

3.22.1.7 char\* hstor\_keylist::prefix

3.22.1.8 bool hstor\_keylist::trunc

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

- include/[hstor.h](#)

## 3.23 hstor\_object Struct Reference

```
#include <hstor.h>
```

### Data Fields

- char \* [key](#)

- char \* [time\\_mod](#)
- char \* [etag](#)
- uint64\_t [size](#)
- char \* [storage](#)
- char \* [own\\_id](#)
- char \* [own\\_name](#)

### 3.23.1 Field Documentation

3.23.1.1 char\* hstor\_object::etag

3.23.1.2 char\* hstor\_object::key

3.23.1.3 char\* hstor\_object::own\_id

3.23.1.4 char\* hstor\_object::own\_name

3.23.1.5 uint64\_t hstor\_object::size

3.23.1.6 char\* hstor\_object::storage

3.23.1.7 char\* hstor\_object::time\_mod

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

- include/[hstor.h](#)

## 3.24 http\_hdr Struct Reference

```
#include <hstor.h>
```

### Data Fields

- char \* [key](#)
- char \* [val](#)

### 3.24.1 Field Documentation

3.24.1.1 char\* http\_hdr::key

3.24.1.2 char\* http\_hdr::val

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

- include/[hstor.h](#)

## 3.25 http\_req Struct Reference

```
#include <hstor.h>
```

### Data Fields

- char \* [method](#)
- struct [http\\_uri](#) uri
- int [major](#)
- int [minor](#)
- char \* [orig\\_path](#)
- unsigned int [n\\_hdr](#)
- struct [http\\_hdr](#) [hdr](#) [HREQ\_MAX\_HDR]

### 3.25.1 Field Documentation

3.25.1.1 struct [http\\_hdr](#) [http\\_req::hdr](#)[HREQ\_MAX\_HDR]

3.25.1.2 int [http\\_req::major](#)

3.25.1.3 char\* [http\\_req::method](#)

3.25.1.4 int [http\\_req::minor](#)

3.25.1.5 unsigned int [http\\_req::n\\_hdr](#)

3.25.1.6 char\* [http\\_req::orig\\_path](#)

3.25.1.7 struct [http\\_uri](#) [http\\_req::uri](#)

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

- include/[hstor.h](#)

## 3.26 http\_uri Struct Reference

```
#include <hstor.h>
```

### Data Fields

- char \* [scheme](#)
- unsigned int [scheme\\_len](#)
- char \* [userinfo](#)
- unsigned int [userinfo\\_len](#)

- char \* [hostname](#)
- unsigned int [hostname\\_len](#)
- unsigned int [port](#)
- char \* [path](#)
- unsigned int [path\\_len](#)
- char \* [query](#)
- unsigned int [query\\_len](#)
- char \* [fragment](#)
- unsigned int [fragment\\_len](#)

### 3.26.1 Field Documentation

3.26.1.1 char\* [http\\_uri::fragment](#)

3.26.1.2 unsigned int [http\\_uri::fragment\\_len](#)

3.26.1.3 char\* [http\\_uri::hostname](#)

3.26.1.4 unsigned int [http\\_uri::hostname\\_len](#)

3.26.1.5 char\* [http\\_uri::path](#)

3.26.1.6 unsigned int [http\\_uri::path\\_len](#)

3.26.1.7 unsigned int [http\\_uri::port](#)

3.26.1.8 char\* [http\\_uri::query](#)

3.26.1.9 unsigned int [http\\_uri::query\\_len](#)

3.26.1.10 char\* [http\\_uri::scheme](#)

3.26.1.11 unsigned int [http\\_uri::scheme\\_len](#)

3.26.1.12 char\* [http\\_uri::userinfo](#)

3.26.1.13 unsigned int [http\\_uri::userinfo\\_len](#)

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

- include/[hstor.h](#)

## 3.27 list\_head Struct Reference

```
#include <elist.h>
```

## Data Fields

- struct [list\\_head](#) \* [next](#)
- struct [list\\_head](#) \* [prev](#)

### 3.27.1 Field Documentation

#### 3.27.1.1 struct [list\\_head](#)\* [list\\_head::next](#)

#### 3.27.1.2 struct [list\\_head](#) \* [list\\_head::prev](#)

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

- include/[elist.h](#)

## 3.28 ncld\_fh Struct Reference

```
#include <ncld.h>
```

## Data Fields

- struct [ncld\\_sess](#) \* [sess](#)
- struct [cldc\\_fh](#) \* [fh](#)
- bool [is\\_open](#)
- int [errc](#)
- int [nios](#)
- unsigned int [event\\_mask](#)
- void(\* [event\\_func](#) )(void \*, unsigned int)
- void \* [event\\_arg](#)

### 3.28.1 Field Documentation

3.28.1.1 `int ncld_fh::errc`

3.28.1.2 `void* ncld_fh::event_arg`

3.28.1.3 `void(* ncld_fh::event_func)(void *, unsigned int)`

3.28.1.4 `unsigned int ncld_fh::event_mask`

3.28.1.5 `struct cldc_fh* ncld_fh::fh`

3.28.1.6 `bool ncld_fh::is_open`

3.28.1.7 `int ncld_fh::nios`

3.28.1.8 `struct ncld_sess* ncld_fh::sess`

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

- `include/ncld.h`

## 3.29 ncld\_read Struct Reference

```
#include <ncld.h>
```

### Data Fields

- `const void * ptr`
- `long length`
- `struct cldc_node_metadata meta`
- `struct ncld_fh * fh`
- `bool is_done`
- `int errc`

### 3.29.1 Field Documentation

3.29.1.1 `int ncld_read::errc`

3.29.1.2 `struct ncld_fh* ncld_read::fh`

3.29.1.3 `bool ncld_read::is_done`

3.29.1.4 `long ncld_read::length`

3.29.1.5 `struct cldc_node_metadata ncld_read::meta`

3.29.1.6 `const void* ncld_read::ptr`

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

- `include/ncld.h`

## 3.30 ncld\_sess Struct Reference

```
#include <ncld.h>
```

### Data Fields

- `char * host`
- `unsigned short port`
- `GMutex * mutex`
- `GCond * cond`
- `GThread * thread`
- `bool is_up`
- `bool open_done`
- `int errc`
- `GList * handles`
- `int to_thread [2]`
- `struct cldc_udp * udp`
- `struct cld_timer udp_timer`
- `struct cld_timer_list tlist`
- `void(* event)(void *, unsigned int)`
- `void * event_arg`

### 3.30.1 Field Documentation

3.30.1.1 `GCond* ncld_sess::cond`

3.30.1.2 `int ncld_sess::errc`

3.30.1.3 `void(* ncld_sess::event)(void *, unsigned int)`

3.30.1.4 `void* ncld_sess::event_arg`

3.30.1.5 `GList* ncld_sess::handles`

3.30.1.6 `char* ncld_sess::host`

3.30.1.7 `bool ncld_sess::is_up`

3.30.1.8 `GMutex* ncld_sess::mutex`

3.30.1.9 `bool ncld_sess::open_done`

3.30.1.10 `unsigned short ncld_sess::port`

3.30.1.11 `GThread* ncld_sess::thread`

3.30.1.12 `struct cld_timer_list ncld_sess::tlist`

3.30.1.13 `int ncld_sess::to_thread[2]`

3.30.1.14 `struct clde_udp* ncld_sess::udp`

3.30.1.15 `struct cld_timer ncld_sess::udp_timer`

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

- `include/ncld.h`

## 3.31 objcache Struct Reference

```
#include <objcache.h>
```

### Data Fields

- `GMutex *` [lock](#)
- `GHashTable *` [table](#)

### 3.31.1 Field Documentation

#### 3.31.1.1 GMutex\* objcache::lock

#### 3.31.1.2 GHashTable\* objcache::table

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

- [include/objcache.h](#)

## 3.32 objcache\_entry Struct Reference

```
#include <objcache.h>
```

### Data Fields

- unsigned int [hash](#)
- unsigned int [flags](#)
- int [ref](#)

### 3.32.1 Field Documentation

#### 3.32.1.1 unsigned int objcache\_entry::flags

#### 3.32.1.2 unsigned int objcache\_entry::hash

#### 3.32.1.3 int objcache\_entry::ref

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

- [include/objcache.h](#)

## 3.33 st\_client Struct Reference

```
#include <chunkc.h>
```

### Data Fields

- char \* [host](#)
- char \* [user](#)
- char \* [key](#)
- bool [verbose](#)
- int [fd](#)

- `SSL_CTX * ssl_ctx`
- `SSL * ssl`
- `char req_buf [sizeof(struct chunksrv_req)+CHD_KEY_SZ]`

### 3.33.1 Field Documentation

3.33.1.1 `int st_client::fd`

3.33.1.2 `char* st_client::host`

3.33.1.3 `char* st_client::key`

3.33.1.4 `char st_client::req_buf[sizeof(struct chunksrv_req)+CHD_KEY_SZ]`

3.33.1.5 `SSL* st_client::ssl`

3.33.1.6 `SSL_CTX* st_client::ssl_ctx`

3.33.1.7 `char* st_client::user`

3.33.1.8 `bool st_client::verbose`

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

- `include/chunkc.h`

## 3.34 st\_keylist Struct Reference

```
#include <chunkc.h>
```

### Data Fields

- `char * name`
- `GList * contents`

### 3.34.1 Field Documentation

3.34.1.1 `GList* st_keylist::contents`

3.34.1.2 `char* st_keylist::name`

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

- `include/chunkc.h`

## 3.35 st\_object Struct Reference

```
#include <chunkc.h>
```

### Data Fields

- char \* [name](#)
- char \* [time\\_mod](#)
- char \* [etag](#)
- uint64\_t [size](#)
- char \* [owner](#)

### 3.35.1 Field Documentation

3.35.1.1 char\* st\_object::etag

3.35.1.2 char\* st\_object::name

3.35.1.3 char\* st\_object::owner

3.35.1.4 uint64\_t st\_object::size

3.35.1.5 char\* st\_object::time\_mod

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

- include/[chunkc.h](#)



## Chapter 4

# File Documentation

### 4.1 include/chunk-private.h File Reference

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

#### Defines

- #define [MDB\\_TPATH\\_FMT](#) "%s/%X"
- #define [BAD\\_TPATH\\_FMT](#) "%s/bad"
- #define [PREFIX\\_LEN](#) 3

#### 4.1.1 Define Documentation

4.1.1.1 #define [BAD\\_TPATH\\_FMT](#) "%s/bad"

4.1.1.2 #define [MDB\\_TPATH\\_FMT](#) "%s/%X"

4.1.1.3 #define [PREFIX\\_LEN](#) 3

### 4.2 include/chunk\_msg.h File Reference

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

#### Data Structures

- struct [chunksrv\\_req](#)
- struct [chunksrv\\_resp](#)
- struct [chunksrv\\_resp\\_get](#)
- struct [chunk\\_check\\_status](#)

- struct `chunksrv_resp_chkstat`

## Defines

- #define `CHUNKD_MAGIC` "CHUNKDv1"

## Enumerations

- enum {  
`CHD_MAGIC_SZ` = 8, `CHD_USER_SZ` = 64, `CHD_KEY_SZ` = 1024, `CHD_CSUM_SZ` = 20,  
`CHD_SIG_SZ` = 64 }
- enum `chunksrv_ops` {  
`CHO_NOP` = 0, `CHO_GET` = 1, `CHO_GET_META` = 2, `CHO_PUT` = 3,  
`CHO_DEL` = 4, `CHO_LIST` = 5, `CHO_LOGIN` = 6, `CHO_TABLE_OPEN` = 7,  
`CHO_CHECK_START` = 8, `CHO_CHECK_STATUS` = 9, `CHO_START_TLS` = 10, `CHO_CP` = 11 }
- enum `chunk_errcode` {  
`che_Success` = 0, `che_AccessDenied` = 1, `che_InternalError` = 2, `che_InvalidArgument` = 3,  
`che_InvalidURI` = 4, `che_NoSuchKey` = 5, `che_SignatureDoesNotMatch` = 6,  
`che_InvalidKey` = 7,  
`che_InvalidTable` = 8, `che_Busy` = 9, `che_KeyExists` = 10 }
- enum `chunk_flags` { `CHF_SYNC` = (1 << 0), `CHF_TBL_CREAT` = (1 << 1),  
`CHF_TBL_EXCL` = (1 << 2) }
- enum `chunk_check_state` { `chk_Off`, `chk_Idle`, `chk_Active` }

### 4.2.1 Define Documentation

#### 4.2.1.1 #define `CHUNKD_MAGIC` "CHUNKDv1"

### 4.2.2 Enumeration Type Documentation

#### 4.2.2.1 anonymous enum

##### Enumerator:

*`CHD_MAGIC_SZ`*

*`CHD_USER_SZ`*

*`CHD_KEY_SZ`*

*`CHD_CSUM_SZ`*

*`CHD_SIG_SZ`*

## 4.2.2.2 enum chunk\_check\_state

Enumerator:

*chk\_Off*  
*chk\_Idle*  
*chk\_Active*

## 4.2.2.3 enum chunk\_errcode

Enumerator:

*che\_Success*  
*che\_AccessDenied*  
*che\_InternalError*  
*che\_InvalidArgument*  
*che\_InvalidURI*  
*che\_NoSuchKey*  
*che\_SignatureDoesNotMatch*  
*che\_InvalidKey*  
*che\_InvalidTable*  
*che\_Busy*  
*che\_KeyExists*

## 4.2.2.4 enum chunk\_flags

Enumerator:

*CHF\_SYNC*  
*CHF\_TBL\_CREAT*  
*CHF\_TBL\_EXCL*

## 4.2.2.5 enum chunksrv\_ops

Enumerator:

*CHO\_NOP*  
*CHO\_GET*  
*CHO\_GET\_META*  
*CHO\_PUT*  
*CHO\_DEL*  
*CHO\_LIST*

*CHO\_LOGIN*  
*CHO\_TABLE\_OPEN*  
*CHO\_CHECK\_START*  
*CHO\_CHECK\_STATUS*  
*CHO\_START\_TLS*  
*CHO\_CP*

### 4.3 include/chunkc.h File Reference

```
#include <sys/types.h>
#include <openssl/ssl.h>
#include <stdbool.h>
#include <stdint.h>
#include <string.h>
#include <glib.h>
#include <chunk_msg.h>
```

#### Data Structures

- struct [st\\_object](#)
- struct [st\\_keylist](#)
- struct [st\\_client](#)

#### Functions

- void [stc\\_free](#) (struct [st\\_client](#) \*stc)
- void [stc\\_free\\_keylist](#) (struct [st\\_keylist](#) \*keylist)
- void [stc\\_free\\_object](#) (struct [st\\_object](#) \*obj)
- void [stc\\_init](#) (void)
- struct [st\\_client](#) \* [stc\\_new](#) (const char \*service\_host, int port, const char \*user, const char \*secret\_key, bool encrypt)
- bool [stc\\_table\\_open](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len, uint32\_t flags)
- bool [stc\\_get](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len, size\_t(\*write\_cb)(void \*, size\_t, size\_t, void \*), void \*user\_data)
- void \* [stc\\_get\\_inline](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len, size\_t \*len)
- bool [stc\\_get\\_start](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len, int \*pfd, uint64\_t \*len)
- size\_t [stc\\_get\\_recv](#) (struct [st\\_client](#) \*stc, void \*data, size\_t len)
- bool [stc\\_put](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len, size\_t(\*read\_cb)(void \*, size\_t, size\_t, void \*), uint64\_t len, void \*user\_data, uint32\_t flags)

- bool [stc\\_put\\_start](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len, uint64\_t cont\_len, int \*pfd, uint32\_t flags)
- size\_t [stc\\_put\\_send](#) (struct [st\\_client](#) \*stc, void \*data, size\_t len)
- bool [stc\\_put\\_sync](#) (struct [st\\_client](#) \*stc)
- bool [stc\\_put\\_inline](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len, void \*data, uint64\_t len, uint32\_t flags)
- bool [stc\\_cp](#) (struct [st\\_client](#) \*stc, const void \*dest\_key, size\_t dest\_key\_len, const void \*src\_key, size\_t src\_key\_len)
- bool [stc\\_del](#) (struct [st\\_client](#) \*stc, const void \*key, size\_t key\_len)
- bool [stc\\_ping](#) (struct [st\\_client](#) \*stc)
- bool [stc\\_check\\_start](#) (struct [st\\_client](#) \*stc)
- bool [stc\\_check\\_status](#) (struct [st\\_client](#) \*stc, struct [chunk\\_check\\_status](#) \*out)
- struct [st\\_keylist](#) \* [stc\\_keys](#) (struct [st\\_client](#) \*stc)
- int [stc\\_readport](#) (const char \*fname)



### 4.3.1 Function Documentation

- 4.3.1.1 `bool stc_check_start ( struct st_client * stc )`
- 4.3.1.2 `bool stc_check_status ( struct st_client * stc, struct chunk_check_status * out )`
- 4.3.1.3 `bool stc_cp ( struct st_client * stc, const void * dest_key, size_t dest_key_len, const void * src_key, size_t src_key_len )`
- 4.3.1.4 `bool stc_del ( struct st_client * stc, const void * key, size_t key_len )`
- 4.3.1.5 `void stc_free ( struct st_client * stc )`
- 4.3.1.6 `void stc_free_keylist ( struct st_keylist * keylist )`
- 4.3.1.7 `void stc_free_object ( struct st_object * obj )`
- 4.3.1.8 `bool stc_get ( struct st_client * stc, const void * key, size_t key_len, size_t(*) (void *, size_t, size_t, void *) write_cb, void * user_data )`
- 4.3.1.9 `void* stc_get_inline ( struct st_client * stc, const void * key, size_t key_len, size_t * len )`
- 4.3.1.10 `size_t stc_get_recv ( struct st_client * stc, void * data, size_t len )`
- 4.3.1.11 `bool stc_get_start ( struct st_client * stc, const void * key, size_t key_len, int * pfd, uint64_t * len )`
- 4.3.1.12 `void stc_init ( void )`
- 4.3.1.13 `struct st_keylist* stc_keys ( struct st_client * stc )` [read]
- 4.3.1.14 `struct st_client* stc_new ( const char * service_host, int port, const char * user, const char * secret_key, bool encrypt )` [read]
- 4.3.1.15 `bool stc_ping ( struct st_client * stc )`
- 4.3.1.16 `bool stc_put ( struct st_client * stc, const void * key, size_t key_len, size_t(*) (void *, size_t, size_t, void *) read_cb, uint64_t len, void * user_data, uint32_t flags )`
- 4.3.1.17 `bool stc_put_inline ( struct st_client * stc, const void * key, size_t key_len, void * data, uint64_t len, uint32_t flags )`
- 4.3.1.18 `size_t stc_put_send ( struct st_client * stc, void * data, size_t len )`
- 4.3.1.19 `bool stc_put_start ( struct st_client * stc, const void * key, size_t key_len, uint64_t cont_len, int * pfd, uint32_t flags )`
- 4.3.1.20 `bool stc_put_sync ( struct st_client * stc )`
- 4.3.1.21 `int stc_readport ( const char * fname )`  
Generated on Wed Feb 9 2011 09:57:35 for CLD by Doxygen
- 4.3.1.22 `bool stc_table_open ( struct st_client * stc, const void * key, size_t key_len, uint32_t flags )`

## 4.4 include/chunksrv.h File Reference

```
#include <chunk_msg.h>
```

## Functions

- `size_t req_len` (const struct `chunksrv_req` \*req)
- `void chreq_sign` (struct `chunksrv_req` \*req, const char \*key, char \*b64hmac\_out)

### 4.4.1 Function Documentation

4.4.1.1 `void chreq_sign ( struct chunksrv_req * req, const char * key, char * b64hmac_out )`

4.4.1.2 `size_t req_len ( const struct chunksrv_req * req )`

## 4.5 include/cld-private.h File Reference

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

## 4.6 include/cld\_common.h File Reference

```
#include <stdint.h>
#include <stdbool.h>
#include <string.h>
#include <time.h>
#include <glib.h>
#include <openssl/sha.h>
#include <cld_msg_rpc.h>
```

## Data Structures

- struct `cld_timer`
- struct `cld_timer_list`

## Defines

- `#define CLD_ALIGN8(n) ((8 - ((n) & 7)) & 7)`
- `#define SIDFMT "%016lX"`
- `#define SIDARG(sid) cld_sid2llu(sid)`
- `#define CLD_PKT_FTR_LEN sizeof(struct cld_pkt_ftr)`  
*Length of the packet footer.*
- `#define PKT_HDR_TO_STR_SCRATCH_LEN 128`

## Functions

- void `cld_timer_add` (struct `cld_timer_list` \*tlist, struct `cld_timer` \*timer, time\_t expires)
- void `cld_timer_del` (struct `cld_timer_list` \*tlist, struct `cld_timer` \*timer)
- time\_t `cld_timers_run` (struct `cld_timer_list` \*tlist)
- unsigned long long `cld_sid2llu` (const uint8\_t \*sid)
- void `cld_rand64` (void \*p)
- const char \* `cld_errstr` (enum `cle_err_codes` ecode)
- int `cld_readport` (const char \*fname)
- int `cld_authcheck` (struct `hail_log` \*log, const char \*key, const void \*buf, size\_t buf\_len, const void \*sha)
- int `cld_authsign` (struct `hail_log` \*log, const char \*key, const void \*buf, size\_t buf\_len, void \*sha)
- const char \* `cld_opstr` (enum `cld_msg_op`)
- const char \* `cld_pkt_hdr_to_str` (char \*scratch, const char \*pkt\_hdr, size\_t pkt\_len)
- void `__cld_dump_buf` (const void \*buf, size\_t len)
- struct `__attribute__` ((packed)) `cld_pkt_ftr`

*Footer that appears at the end of each packet.*

### 4.6.1 Define Documentation

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

4.6.1.2 `#define CLD_PKT_FTR_LEN sizeof(struct cld_pkt_ftr)`

Length of the packet footer.

This size is fixed

4.6.1.3 `#define PKT_HDR_TO_STR_SCRATCH_LEN 128`

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

4.6.1.5 `#define SIDFMT "%016llx"`

### 4.6.2 Function Documentation

4.6.2.1 struct `__attribute__` ((packed)) [read]

Footer that appears at the end of each packet.

< packet sequence ID

< packet signature

- 4.6.2.2 void `_cld_dump_buf` ( const void \* *buf*, size\_t *len* )
- 4.6.2.3 int `cld_authcheck` ( struct hail\_log \* *log*, const char \* *key*, const void \* *buf*, size\_t *buf\_len*, const void \* *sha* )
- 4.6.2.4 int `cld_authsign` ( struct hail\_log \* *log*, const char \* *key*, const void \* *buf*, size\_t *buf\_len*, void \* *sha* )
- 4.6.2.5 const char\* `cld_errstr` ( enum cle\_err\_codes *ecode* )
- 4.6.2.6 const char\* `cld_opstr` ( enum *cld\_msg\_op* )
- 4.6.2.7 const char\* `cld_pkt_hdr_to_str` ( char \* *scratch*, const char \* *pkt\_hdr*, size\_t *pkt\_len* )
- 4.6.2.8 void `cld_rand64` ( void \* *p* )
- 4.6.2.9 int `cld_readport` ( const char \* *fname* )
- 4.6.2.10 unsigned long long `cld_sid2llu` ( const uint8\_t \* *sid* )
- 4.6.2.11 void `cld_timer_add` ( struct cld\_timer\_list \* *tlist*, struct cld\_timer \* *timer*, time\_t *expires* )
- 4.6.2.12 void `cld_timer_del` ( struct cld\_timer\_list \* *tlist*, struct cld\_timer \* *timer* )
- 4.6.2.13 time\_t `cld_timers_run` ( struct cld\_timer\_list \* *tlist* )

## 4.7 include/cldc.h File Reference

```
#include <sys/types.h>
#include <stdbool.h>
#include <glib.h>
#include <cld_msg_rpc.h>
#include <cld_common.h>
#include <hail_log.h>
```

### Data Structures

- struct [cldc\\_call\\_opts](#)  
*per-operation application options*
- struct [cldc\\_node\\_metadata](#)
- 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_copts_get_data` (const struct `cldc_call_opts` \*copts, char \*\*data, size\_t \*data\_len)
- void `cldc_copts_get_metadata` (const struct `cldc_call_opts` \*copts, struct `cldc_node_metadata` \*md)
- 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)
- int `cldc_getaddr` (GList \*\*host\_list, const char \*thishost, struct `hail_log` \*log)
- int `cldc_saveaddr` (struct `cldc_host` \*hp, unsigned int priority, unsigned int weight, unsigned int port, unsigned int nlen, const char \*name, struct `hail_log` \*log)



### 4.7.1 Function Documentation

- 4.7.1.1 `int cldc_close ( struct cldc_fh * fh, const struct cldc_call_opts * copts )`
- 4.7.1.2 `void cldc_copts_get_data ( const struct cldc_call_opts * copts, char ** data, size_t * data_len )`
- 4.7.1.3 `void cldc_copts_get_metadata ( const struct cldc_call_opts * copts, struct cldc_node_metadata * md )`
- 4.7.1.4 `int cldc_del ( struct cldc_session * sess, const struct cldc_call_opts * copts, const char * pathname )`
- 4.7.1.5 `int cldc_dirent_count ( const void * data, size_t data_len )`
- 4.7.1.6 `void cldc_dirent_cur_fini ( struct cld_dirent_cur * dc )`
- 4.7.1.7 `void cldc_dirent_cur_init ( struct cld_dirent_cur * dc, const void * buf, size_t buflen )`
- 4.7.1.8 `int cldc_dirent_first ( struct cld_dirent_cur * dc )`
- 4.7.1.9 `char* cldc_dirent_name ( struct cld_dirent_cur * dc )`
- 4.7.1.10 `int cldc_dirent_next ( struct cld_dirent_cur * dc )`
- 4.7.1.11 `int cldc_end_sess ( struct cldc_session * sess, const struct cldc_call_opts * copts )`
- 4.7.1.12 `int cldc_get ( struct cldc_fh * fh, const struct cldc_call_opts * copts, bool metadata_only )`
- 4.7.1.13 `int cldc_getaddr ( GList ** host_list, const char * thishost, struct hail_log * log )`
- 4.7.1.14 `void cldc_init ( void )`
- 4.7.1.15 `void cldc_kill_sess ( struct cldc_session * sess )`
- 4.7.1.16 `int cldc_lock ( struct cldc_fh * fh, const struct cldc_call_opts * copts, uint32_t lock_flags, bool wait_for_lock )`
- 4.7.1.17 `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 )`
- 4.7.1.18 `int cldc_nop ( struct cldc_session * sess, const struct cldc_call_opts * copts )`
- 4.7.1.19 `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 )`
- 4.7.1.20 `int cldc_put ( struct cldc_fh * fh, const struct cldc_call_opts * copts, const void * data, size_t data_len )`
- 4.7.1.21 `int cldc_receive_pkt ( struct cldc_session * sess, const void * net_addr, size_t net_addrlen, const void * buf, size_t buflen )`

Generated on Wed Feb 9 2011 09:57:35 for CLD by Doxygen

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

#### Parameters

<i>sess</i>	Session associated with received packet
<i>net_addr</i>	Opaque network address
<i>net_addrlen</i>	Size of opaque network address
<i>buf</i>	Pointer to data buffer containing packet
<i>buflen</i>	Length of received packet

#### Returns

Zero for success, non-zero on error

4.7.1.22 `int cldc_saveaddr ( struct cldc_host * hp, unsigned int priority, unsigned int weight, unsigned int port, unsigned int nlen, const char * name, struct hail_log * log )`

4.7.1.23 `void cldc_udp_free ( struct cldc_udp * udp )`

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

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

4.7.1.26 `int cldc_udp_receive_pkt ( struct cldc_udp * udp )`

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

## 4.8 include/elist.h File Reference

### Data Structures

- struct [list\\_head](#)

### Defines

- `#define LIST_HEAD_INIT(name) { &(amp;name), &(name) }`
- `#define LIST_HEAD(name) struct list_head name = LIST_HEAD_INIT(name)`
- `#define INIT_LIST_HEAD(ptr)`
- `#define list_entry(ptr, type, member) ((type *)((char *)(ptr)-(unsigned long)&((type *)0)->member))`

*list\_entry* - get the struct for this entry : the &struct [list\\_head](#) pointer.

- `#define list_for_each(pos, head)`  
*list\_for\_each* - iterate over a list : the &struct [list\\_head](#) to use as a loop counter.
- `#define list_for_each_prev(pos, head)`

*list\_for\_each\_prev* - iterate over a list backwards : the &struct [list\\_head](#) to use as a loop counter.

- **#define [list\\_for\\_each\\_safe](#)(pos, n, head)**

*list\_for\_each\_safe* - iterate over a list safe against removal of list entry : the &struct [list\\_head](#) to use as a loop counter.

- **#define [list\\_for\\_each\\_entry](#)(pos, head, member)**

*list\_for\_each\_entry* - iterate over list of given type : the type \* to use as a loop counter.

- **#define [list\\_for\\_each\\_entry\\_safe](#)(pos, n, head, member)**

*list\_for\_each\_entry\_safe* - iterate over list of given type safe against removal of list entry : the type \* to use as a loop counter.

- **#define [list\\_for\\_each\\_entry\\_continue](#)(pos, head, member)**

*list\_for\_each\_entry\_continue* - iterate over list of given type continuing after existing point : the type \* to use as a loop counter.

## 4.8.1 Define Documentation

### 4.8.1.1 **#define INIT\_LIST\_HEAD( ptr )**

**Value:**

```
do { \
    (ptr)->next = (ptr); (ptr)->prev = (ptr); \
} while (0)
```

### 4.8.1.2 **#define list\_entry( ptr, type, member ) ((type \*)((char \*)(ptr)-(unsigned long)(&((type \*)0)->member)))**

*list\_entry* - get the struct for this entry : the &struct [list\\_head](#) pointer.

: the type of the struct this is embedded in. : the name of the list\_struct within the struct.

### 4.8.1.3 **#define list\_for\_each( pos, head )**

**Value:**

```
for (pos = (head)->next; pos != (head); \
     pos = pos->next)
```

*list\_for\_each* - iterate over a list : the &struct [list\\_head](#) to use as a loop counter.

: the head for your list.

**4.8.1.4 #define list\_for\_each\_entry( pos, head, member )****Value:**

```
for (pos = list_entry((head)->next, typeof(*pos), member); \
    &pos->member != (head); \
    pos = list_entry(pos->member.next, typeof(*pos), member))
```

list\_for\_each\_entry - iterate over list of given type : the type \* to use as a loop counter.

: the head for your list. : the name of the list\_struct within the struct.

**4.8.1.5 #define list\_for\_each\_entry\_continue( pos, head, member )****Value:**

```
for (pos = list_entry(pos->member.next, typeof(*pos), member), \
    prefetch(pos->member.next); \
    &pos->member != (head); \
    pos = list_entry(pos->member.next, typeof(*pos), member), \
    prefetch(pos->member.next))
```

list\_for\_each\_entry\_continue - iterate over list of given type continuing after existing point : the type \* to use as a loop counter.

: the head for your list. : the name of the list\_struct within the struct.

**4.8.1.6 #define list\_for\_each\_entry\_safe( pos, n, head, member )****Value:**

```
for (pos = list_entry((head)->next, typeof(*pos), member), \
    n = list_entry(pos->member.next, typeof(*pos), member); \
    &pos->member != (head); \
    pos = n, n = list_entry(n->member.next, typeof(*n), member))
```

list\_for\_each\_entry\_safe - iterate over list of given type safe against removal of list entry : the type \* to use as a loop counter.

: another type \* to use as temporary storage : the head for your list. : the name of the list\_struct within the struct.

**4.8.1.7 #define list\_for\_each\_prev( pos, head )****Value:**

```
for (pos = (head)->prev; pos != (head); \
    pos = pos->prev)
```

list\_for\_each\_prev - iterate over a list backwards : the &struct [list\\_head](#) to use as a loop counter.

: the head for your list.

#### 4.8.1.8 #define list\_for\_each\_safe( pos, n, head )

**Value:**

```
for (pos = (head)->next, n = pos->next; pos != (head); \
     pos = n, n = pos->next)
```

list\_for\_each\_safe - iterate over a list safe against removal of list entry : the &struct [list\\_head](#) to use as a loop counter.

: another &struct [list\\_head](#) to use as temporary storage : the head for your list.

#### 4.8.1.9 #define LIST\_HEAD( name ) struct list\_head name = LIST\_HEAD\_INIT(name)

#### 4.8.1.10 #define LIST\_HEAD\_INIT( name ) { &(amp;name), &(name) }

## 4.9 include/hail\_log.h File Reference

```
#include <stdbool.h>
```

### Data Structures

- struct [hail\\_log](#)

### Defines

- #define [ATTR\\_PRINTF](#)(x, y)
- #define [HAIL\\_VERBOSE](#)(log,...)  
*Print out a CLD session debug message if enabled.*
- #define [HAIL\\_DEBUG](#)(log,...)  
*Print out an application debug message if enabled.*
- #define [HAIL\\_INFO](#)(log,...) (log)->func(LOG\_INFO, \_\_VA\_ARGS\_\_)  
*Print out an informational log message.*
- #define [HAIL\\_WARN](#)(log,...) (log)->func(LOG\_WARNING, \_\_VA\_ARGS\_\_-  
)  
*Print out a warning message.*
- #define [HAIL\\_ERR](#)(log,...) (log)->func(LOG\_ERR, \_\_VA\_ARGS\_\_)  
*Print out an error message.*
- #define [HAIL\\_CRIT](#)(log,...) (log)->func(LOG\_CRIT, \_\_VA\_ARGS\_\_)  
*Print out a critical warning message.*

### 4.9.1 Define Documentation

4.9.1.1 **#define ATTR\_PRINTF( *x*, *y* )**

4.9.1.2 **#define HAIL\_CRIT( *log*, ... )** (log)->func(LOG\_CRIT, \_\_VA\_ARGS\_\_)

Print out a critical warning message.

4.9.1.3 **#define HAIL\_DEBUG( *log*, ... )**

**Value:**

```
if ((log)->debug) { \
    (log)->func(LOG_DEBUG, __VA_ARGS__); \
}
```

Print out an application debug message if enabled.

4.9.1.4 **#define HAIL\_ERR( *log*, ... )** (log)->func(LOG\_ERR, \_\_VA\_ARGS\_\_)

Print out an error message.

4.9.1.5 **#define HAIL\_INFO( *log*, ... )** (log)->func(LOG\_INFO, \_\_VA\_ARGS\_\_)

Print out an informational log message.

4.9.1.6 **#define HAIL\_VERBOSE( *log*, ... )**

**Value:**

```
if ((log)->verbose) { \
    (log)->func(LOG_DEBUG, __VA_ARGS__); \
}
```

Print out a CLD session debug message if enabled.

4.9.1.7 **#define HAIL\_WARN( *log*, ... )** (log)->func(LOG\_WARNING, \_\_VA\_ARGS\_\_)

Print out a warning message.

## 4.10 include/hail\_private.h File Reference

```
#include "hail-config.h"
```

```
#include <rpc/xdr.h>
```

## Functions

- u\_long [xdr\\_sizeof](#) (xdrproc\_t, void \*)

### 4.10.1 Function Documentation

#### 4.10.1.1 u\_long xdr\_sizeof ( xdrproc\_t , void \* )

## 4.11 include/hstor.h File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include <curl/curl.h>
#include <glib.h>
```

## Data Structures

- struct [hstor\\_client](#)
- struct [hstor\\_bucket](#)
- struct [hstor\\_blist](#)
- struct [hstor\\_object](#)
- struct [hstor\\_keylist](#)
- struct [http\\_uri](#)
- struct [http\\_hdr](#)
- struct [http\\_req](#)

## Defines

- #define [ARRAY\\_SIZE](#)(arr) (sizeof(arr) / sizeof((arr)[0]))
- #define [PATH\\_ESCAPE\\_MASK](#) 0x02
- #define [QUERY\\_ESCAPE\\_MASK](#) 0x04

## Enumerations

- enum { [HREQ\\_MAX\\_HDR](#) = 128 }
- enum [ReqQ](#) {  
    [URIQ\\_ACL](#), [URIQ\\_LOCATION](#), [URIQ\\_LOGGING](#), [URIQ\\_TORRENT](#),  
    [URIQNUM](#) }
- enum [ReqACLC](#) {  
    [ACLC\\_PRIV](#), [ACLC\\_PUB\\_R](#), [ACLC\\_PUB\\_RW](#), [ACLC\\_AUTH\\_R](#),  
    [ACLCNUM](#) }

## Functions

- char \* [hutil\\_time2str](#) (char \*buf, int len, time\_t time)
- time\_t [hutil\\_str2time](#) (const char \*timestr)
- int [hreq\\_hdr\\_push](#) (struct [http\\_req](#) \*req, char \*key, char \*val)
- char \* [hreq\\_hdr](#) (struct [http\\_req](#) \*req, const char \*key)
- void [hreq\\_sign](#) (struct [http\\_req](#) \*req, const char \*bucket, const char \*key, char \*b64hmac\_out)
- GHashTable \* [hreq\\_query](#) (struct [http\\_req](#) \*req)
- int [hreq\\_is\\_query](#) (struct [http\\_req](#) \*req)
- void [hreq\\_free](#) (struct [http\\_req](#) \*req)
- int [hreq\\_acl\\_canned](#) (struct [http\\_req](#) \*req)
- struct [http\\_uri](#) \* [huri\\_parse](#) (struct [http\\_uri](#) \*uri\_dest, char \*uri\_src\_text)
- int [huri\\_field\\_unescape](#) (char \*s, int s\_len)
- char \* [huri\\_field\\_escape](#) (char \*signed\_str, unsigned char mask)
- void [hstor\\_free](#) (struct [hstor\\_client](#) \*hstor)
- void [hstor\\_free\\_blist](#) (struct [hstor\\_blist](#) \*blist)
- void [hstor\\_free\\_bucket](#) (struct [hstor\\_bucket](#) \*buck)
- void [hstor\\_free\\_object](#) (struct [hstor\\_object](#) \*obj)
- void [hstor\\_free\\_keylist](#) (struct [hstor\\_keylist](#) \*keylist)
- struct [hstor\\_client](#) \* [hstor\\_new](#) (const char \*service\_acc, const char \*service\_host, const char \*user, const char \*secret\_key)
- bool [hstor\\_add\\_bucket](#) (struct [hstor\\_client](#) \*hstor, const char \*name)
- bool [hstor\\_del\\_bucket](#) (struct [hstor\\_client](#) \*hstor, const char \*name)
- struct [hstor\\_blist](#) \* [hstor\\_list\\_buckets](#) (struct [hstor\\_client](#) \*hstor)
- bool [hstor\\_get](#) (struct [hstor\\_client](#) \*hstor, const char \*bucket, const char \*key, size\_t(\*write\_cb)(void \*, size\_t, size\_t, void \*), void \*user\_data, bool want\_headers)
- void \* [hstor\\_get\\_inline](#) (struct [hstor\\_client](#) \*hstor, const char \*bucket, const char \*key, bool want\_headers, size\_t \*len)
- bool [hstor\\_put](#) (struct [hstor\\_client](#) \*hstor, const char \*bucket, const char \*key, size\_t(\*read\_cb)(void \*, size\_t, size\_t, void \*), uint64\_t len, void \*user\_data, char \*\*user\_hdrs)
- bool [hstor\\_put\\_inline](#) (struct [hstor\\_client](#) \*hstor, const char \*bucket, const char \*key, void \*data, uint64\_t len, char \*\*user\_hdrs)
- bool [hstor\\_del](#) (struct [hstor\\_client](#) \*hstor, const char \*bucket, const char \*key)
- struct [hstor\\_keylist](#) \* [hstor\\_keys](#) (struct [hstor\\_client](#) \*hstor, const char \*bucket, const char \*prefix, const char \*marker, const char \*delim, unsigned int max\_keys)

### 4.11.1 Define Documentation

4.11.1.1 `#define ARRAY_SIZE( arr ) (sizeof(arr) / sizeof((arr)[0]))`

4.11.1.2 `#define PATH_ESCAPE_MASK 0x02`

4.11.1.3 `#define QUERY_ESCAPE_MASK 0x04`

### 4.11.2 Enumeration Type Documentation

4.11.2.1 anonymous enum

Enumerator:

*HREQ\_MAX\_HDR*

4.11.2.2 enum ReqACLC

Enumerator:

*ACLC\_PRIV*

*ACLC\_PUB\_R*

*ACLC\_PUB\_RW*

*ACLC\_AUTH\_R*

*ACLCNUM*

4.11.2.3 enum ReqQ

Enumerator:

*URIQ\_ACL*

*URIQ\_LOCATION*

*URIQ\_LOGGING*

*URIQ\_TORRENT*

*URIQNUM*



### 4.11.3 Function Documentation

- 4.11.3.1 `int hreq_acl_canned ( struct http_req * req )`
- 4.11.3.2 `void hreq_free ( struct http_req * req )`
- 4.11.3.3 `char* hreq_hdr ( struct http_req * req, const char * key )`
- 4.11.3.4 `int hreq_hdr_push ( struct http_req * req, char * key, char * val )`
- 4.11.3.5 `int hreq_is_query ( struct http_req * req )`
- 4.11.3.6 `GHashTable* hreq_query ( struct http_req * req )`
- 4.11.3.7 `void hreq_sign ( struct http_req * req, const char * bucket, const char * key, char * b64hmac_out )`
- 4.11.3.8 `bool hstor_add_bucket ( struct hstor_client * hstor, const char * name )`
- 4.11.3.9 `bool hstor_del ( struct hstor_client * hstor, const char * bucket, const char * key )`
- 4.11.3.10 `bool hstor_del_bucket ( struct hstor_client * hstor, const char * name )`
- 4.11.3.11 `void hstor_free ( struct hstor_client * hstor )`
- 4.11.3.12 `void hstor_free_blist ( struct hstor_blist * blist )`
- 4.11.3.13 `void hstor_free_bucket ( struct hstor_bucket * buck )`
- 4.11.3.14 `void hstor_free_keylist ( struct hstor_keylist * keylist )`
- 4.11.3.15 `void hstor_free_object ( struct hstor_object * obj )`
- 4.11.3.16 `bool hstor_get ( struct hstor_client * hstor, const char * bucket, const char * key, size_t*(void *, size_t, size_t, void *) write_cb, void * user_data, bool want_headers )`
- 4.11.3.17 `void* hstor_get_inline ( struct hstor_client * hstor, const char * bucket, const char * key, bool want_headers, size_t * len )`
- 4.11.3.18 `struct hstor_keylist* hstor_keys ( struct hstor_client * hstor, const char * bucket, const char * prefix, const char * marker, const char * delim, unsigned int max_keys ) [read]`
- 4.11.3.19 `struct hstor_blist* hstor_list_buckets ( struct hstor_client * hstor ) [read]`
- 4.11.3.20 `struct hstor_client* hstor_new ( const char * service_acc, const char * service_host, const char * user, const char * secret_key ) [read]`
- 4.11.3.21 `bool hstor_put ( struct hstor_client * hstor, const char * bucket, const char * key, size_t*(void *, size_t, size_t, void *) read_cb, uint64_t len, void * user_data, char ** user_hdrs )`
- 
- 4.11.3.22 `bool hstor_put_inline ( struct hstor_client * hstor, const char * bucket, const char * key, void * data, uint64_t len, char ** user_hdrs )`
- 4.11.3.23 `char* huri_field_escape ( char * signed_str, unsigned char mask )`
- 4.11.3.24 `int huri_field_unescape ( char * s, int s_len )`

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

## Data Structures

- struct [ncld\\_sess](#)
- struct [ncld\\_fh](#)
- struct [ncld\\_read](#)

## Functions

- struct [ncld\\_sess](#) \* [ncld\\_sess\\_open](#) (const char \*host, int port, int \*error, void(\*event)(void \*, unsigned int), void \*ev\_arg, const char \*cld\_user, const char \*cld\_key, struct [hail\\_log](#) \*log)
- struct [ncld\\_fh](#) \* [ncld\\_open](#) (struct [ncld\\_sess](#) \*s, const char \*fname, unsigned int mode, int \*error, unsigned int events, void(\*event)(void \*, unsigned int), void \*ev\_arg)
- int [ncld\\_del](#) (struct [ncld\\_sess](#) \*nsess, const char \*fname)
- struct [ncld\\_read](#) \* [ncld\\_get](#) (struct [ncld\\_fh](#) \*fh, int \*error)
- struct [ncld\\_read](#) \* [ncld\\_get\\_meta](#) (struct [ncld\\_fh](#) \*fh, int \*error)
- void [ncld\\_read\\_free](#) (struct [ncld\\_read](#) \*rp)
- int [ncld\\_write](#) (struct [ncld\\_fh](#) \*, const void \*data, long len)
- int [ncld\\_trylock](#) (struct [ncld\\_fh](#) \*)
- int [ncld\\_qlock](#) (struct [ncld\\_fh](#) \*)
- int [ncld\\_unlock](#) (struct [ncld\\_fh](#) \*)
- void [ncld\\_close](#) (struct [ncld\\_fh](#) \*)
- void [ncld\\_sess\\_close](#) (struct [ncld\\_sess](#) \*s)
- void [ncld\\_init](#) (void)

### 4.12.1 Function Documentation

4.12.1.1 void `nclد_close ( struct nclد_fh * )`

4.12.1.2 int `nclد_del ( struct nclد_sess * nsess, const char * fname )`

4.12.1.3 struct nclد\_read\* `nclد_get ( struct nclد_fh * fh, int * error )` [read]

4.12.1.4 struct nclد\_read\* `nclد_get_meta ( struct nclد_fh * fh, int * error )` [read]

4.12.1.5 void `nclد_init ( void )`

4.12.1.6 struct nclد\_fh\* `nclد_open ( struct nclد_sess * s, const char * fname, unsigned int mode, int * error, unsigned int events, void(*) (void *, unsigned int) event, void * ev_arg )` [read]

4.12.1.7 int `nclد_qlock ( struct nclد_fh * )`

4.12.1.8 void `nclد_read.free ( struct nclد_read * rp )`

4.12.1.9 void `nclد_sess.close ( struct nclد_sess * s )`

4.12.1.10 struct nclد\_sess\* `nclد_sess.open ( const char * host, int port, int * error, void(*) (void *, unsigned int) event, void * ev_arg, const char * cld_user, const char * cld_key, struct hail_log * log )` [read]

4.12.1.11 int `nclد_trylock ( struct nclد_fh * )`

4.12.1.12 int `nclد_unlock ( struct nclد_fh * )`

4.12.1.13 int `nclد_write ( struct nclد_fh * , const void * data, long len )`

## 4.13 include/objcache.h File Reference

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

### Data Structures

- struct [objcache](#)
- struct [objcache\\_entry](#)

### Defines

- #define [OC\\_F\\_DIRTY](#) 0x1
- #define [objcache\\_get](#)(c, k, l) \_\_objcache\_get(c, k, l, 0)

- #define `objcache_get_dirty`(c, k, l) `__objcache_get`(c, k, l, OC\_F\_DIRTY)

## Functions

- struct `objcache_entry` \* `__objcache_get` (struct `objcache` \*cache, const char \*key, int klen, unsigned int flag)
- bool `objcache_test_dirty` (struct `objcache` \*cache, struct `objcache_entry` \*entry)
- void `objcache_put` (struct `objcache` \*cache, struct `objcache_entry` \*entry)
- int `objcache_count` (struct `objcache` \*cache)
- int `objcache_init` (struct `objcache` \*cache)
- void `objcache_fini` (struct `objcache` \*cache)

### 4.13.1 Define Documentation

4.13.1.1 #define `objcache_get`( c, k, l ) `__objcache_get`(c, k, l, 0)

4.13.1.2 #define `objcache_get_dirty`( c, k, l ) `__objcache_get`(c, k, l, OC\_F\_DIRTY)

4.13.1.3 #define `OC_F_DIRTY` 0x1

### 4.13.2 Function Documentation

4.13.2.1 struct `objcache_entry`\* `__objcache_get` ( struct `objcache` \* *cache*, const char \* *key*, int *klen*, unsigned int *flag* ) [read]

4.13.2.2 int `objcache_count` ( struct `objcache` \* *cache* )

4.13.2.3 void `objcache_fini` ( struct `objcache` \* *cache* )

4.13.2.4 int `objcache_init` ( struct `objcache` \* *cache* )

4.13.2.5 void `objcache_put` ( struct `objcache` \* *cache*, struct `objcache_entry` \* *entry* )

4.13.2.6 bool `objcache_test_dirty` ( struct `objcache` \* *cache*, struct `objcache_entry` \* *entry* )

# Index

\_\_attribute\_\_  
    cld\_common.h, [41](#)  
\_\_cld\_dump\_buf  
    cld\_common.h, [41](#)  
\_\_objcache\_get  
    objcache.h, [59](#)  
  
acc  
    hstor\_client, [20](#)  
ACLC\_AUTH\_R  
    hstor.h, [54](#)  
ACLC\_PRIV  
    hstor.h, [54](#)  
ACLC\_PUB\_R  
    hstor.h, [54](#)  
ACLC\_PUB\_RW  
    hstor.h, [54](#)  
ACLCNUM  
    hstor.h, [54](#)  
addr  
    cldc\_session, [17](#)  
    cldc\_udp, [18](#)  
addr\_len  
    cldc\_session, [17](#)  
    cldc\_udp, [18](#)  
ARRAY\_SIZE  
    hstor.h, [54](#)  
ATTR\_PRINTF  
    hail\_log.h, [51](#)  
  
BAD\_TPATH\_FMT  
    chunk-private.h, [33](#)  
  
cb  
    cld\_timer, [9](#)  
    cldc\_call\_opts, [10](#)  
    cldc\_msg, [12](#)  
    cldc\_udp, [18](#)  
cb\_private  
    cldc\_msg, [12](#)  
    cldc\_udp, [18](#)  
  
cfh  
    cldc\_session, [17](#)  
CHD\_CSUM\_SZ  
    chunk\_msg.h, [34](#)  
CHD\_KEY\_SZ  
    chunk\_msg.h, [34](#)  
CHD\_MAGIC\_SZ  
    chunk\_msg.h, [34](#)  
CHD\_SIG\_SZ  
    chunk\_msg.h, [34](#)  
CHD\_USER\_SZ  
    chunk\_msg.h, [34](#)  
che\_AccessDenied  
    chunk\_msg.h, [35](#)  
che\_Busy  
    chunk\_msg.h, [35](#)  
che\_InternalError  
    chunk\_msg.h, [35](#)  
che\_InvalidArgument  
    chunk\_msg.h, [35](#)  
che\_InvalidKey  
    chunk\_msg.h, [35](#)  
che\_InvalidTable  
    chunk\_msg.h, [35](#)  
che\_InvalidURI  
    chunk\_msg.h, [35](#)  
che\_KeyExists  
    chunk\_msg.h, [35](#)  
che\_NoSuchKey  
    chunk\_msg.h, [35](#)  
che\_SignatureDoesNotMatch  
    chunk\_msg.h, [35](#)  
che\_Success  
    chunk\_msg.h, [35](#)  
CHF\_SYNC  
    chunk\_msg.h, [35](#)  
CHF\_TBL\_CREAT  
    chunk\_msg.h, [35](#)  
CHF\_TBL\_EXCL  
    chunk\_msg.h, [35](#)  
chk\_Active

- chunk\_msg.h, 35
- chk\_Idle
  - chunk\_msg.h, 35
- chk\_Off
  - chunk\_msg.h, 35
- chkstat
  - chunksrv\_resp\_chkstat, 7
- CHO\_CHECK\_START
  - chunk\_msg.h, 36
- CHO\_CHECK\_STATUS
  - chunk\_msg.h, 36
- CHO\_CP
  - chunk\_msg.h, 36
- CHO\_DEL
  - chunk\_msg.h, 35
- CHO\_GET
  - chunk\_msg.h, 35
- CHO\_GET\_META
  - chunk\_msg.h, 35
- CHO\_LIST
  - chunk\_msg.h, 35
- CHO\_LOGIN
  - chunk\_msg.h, 35
- CHO\_NOP
  - chunk\_msg.h, 35
- CHO\_PUT
  - chunk\_msg.h, 35
- CHO\_START\_TLS
  - chunk\_msg.h, 36
- CHO\_TABLE\_OPEN
  - chunk\_msg.h, 36
- chreq\_sign
  - chunksrv.h, 40
- chunk-private.h
  - BAD\_TPATH\_FMT, 33
  - MDB\_TPATH\_FMT, 33
  - PREFIX\_LEN, 33
- chunk\_msg.h
  - CHD\_CSUM\_SZ, 34
  - CHD\_KEY\_SZ, 34
  - CHD\_MAGIC\_SZ, 34
  - CHD\_SIG\_SZ, 34
  - CHD\_USER\_SZ, 34
  - che\_AccessDenied, 35
  - che\_Busy, 35
  - che\_InternalError, 35
  - che\_InvalidArgument, 35
  - che\_InvalidKey, 35
  - che\_InvalidTable, 35
  - che\_InvalidURI, 35
  - che\_KeyExists, 35
  - che\_NoSuchKey, 35
  - che\_SignatureDoesNotMatch, 35
  - che\_Success, 35
  - CHF\_SYNC, 35
  - CHF\_TBL\_CREAT, 35
  - CHF\_TBL\_EXCL, 35
  - chk\_Active, 35
  - chk\_Idle, 35
  - chk\_Off, 35
  - CHO\_CHECK\_START, 36
  - CHO\_CHECK\_STATUS, 36
  - CHO\_CP, 36
  - CHO\_DEL, 35
  - CHO\_GET, 35
  - CHO\_GET\_META, 35
  - CHO\_LIST, 35
  - CHO\_LOGIN, 35
  - CHO\_NOP, 35
  - CHO\_PUT, 35
  - CHO\_START\_TLS, 36
  - CHO\_TABLE\_OPEN, 36
- chunk\_check\_state
  - chunk\_msg.h, 34
- chunk\_check\_status, 5
  - count, 5
  - lastdone, 5
  - pad, 5
  - state, 5
- chunk\_errcode
  - chunk\_msg.h, 35
- chunk\_flags
  - chunk\_msg.h, 35
- chunk\_msg.h
  - chunk\_check\_state, 34
  - chunk\_errcode, 35
  - chunk\_flags, 35
  - CHUNKD\_MAGIC, 34
  - chunksrv\_ops, 35
- chunkc.h
  - stc\_check\_start, 39
  - stc\_check\_status, 39
  - stc\_cp, 39
  - stc\_del, 39
  - stc\_free, 39
  - stc\_free\_keylist, 39
  - stc\_free\_object, 39
  - stc\_get, 39
  - stc\_get\_inline, 39
  - stc\_get\_recv, 39

- stc\_get\_start, 39
- stc\_init, 39
- stc\_keys, 39
- stc\_new, 39
- stc\_ping, 39
- stc\_put, 39
- stc\_put\_inline, 39
- stc\_put\_send, 39
- stc\_put\_start, 39
- stc\_put\_sync, 39
- stc\_readport, 39
- stc\_table\_open, 39
- CHUNKD\_MAGIC
- chunk\_msg.h, 34
- chunksrv.h
- chreq\_sign, 40
- req\_len, 40
- chunksrv\_ops
- chunk\_msg.h, 35
- chunksrv\_req, 5
- data\_len, 6
- flags, 6
- key\_len, 6
- magic, 6
- nonce, 6
- op, 6
- sig, 6
- chunksrv\_resp, 6
- data\_len, 7
- hash, 7
- magic, 7
- nonce, 7
- resp\_code, 7
- rsv1, 7
- chunksrv\_resp\_chkstat, 7
- chkstat, 7
- resp, 7
- chunksrv\_resp\_get, 7
- mtime, 8
- resp, 8
- CLD\_ALIGN8
- cld\_common.h, 41
- cld\_authcheck
- cld\_common.h, 42
- cld\_authsign
- cld\_common.h, 42
- cld\_common.h
- \_\_attribute\_\_, 41
- \_\_cld\_dump\_buf, 41
- CLD\_ALIGN8, 41
- cld\_authcheck, 42
- cld\_authsign, 42
- cld\_errstr, 42
- cld\_opstr, 42
- CLD\_PKT\_FTR\_LEN, 41
- cld\_pkt\_hdr\_to\_str, 42
- cld\_rand64, 42
- cld\_readport, 42
- cld\_sid2llu, 42
- cld\_timer\_add, 42
- cld\_timer\_del, 42
- cld\_timers\_run, 42
- PKT\_HDR\_TO\_STR\_SCRATCH\_LEN, 41
- SIDARG, 41
- SIDFMT, 41
- cld\_dirent\_cur, 8
- p, 8
- tmp\_len, 8
- cld\_errstr
- cld\_common.h, 42
- cld\_opstr
- cld\_common.h, 42
- CLD\_PKT\_FTR\_LEN
- cld\_common.h, 41
- cld\_pkt\_hdr\_to\_str
- cld\_common.h, 42
- cld\_rand64
- cld\_common.h, 42
- cld\_readport
- cld\_common.h, 42
- cld\_sid2llu
- cld\_common.h, 42
- cld\_timer, 8
- cb, 9
- expires, 9
- fired, 9
- name, 9
- on\_list, 9
- userdata, 9
- cld\_timer\_add
- cld\_common.h, 42
- cld\_timer\_del
- cld\_common.h, 42
- cld\_timer\_list, 9
- list, 9
- runmark, 9
- cld\_timers\_run
- cld\_common.h, 42
- cldc.h

- cldc\_close, [46](#)
- cldc\_copts\_get\_data, [46](#)
- cldc\_copts\_get\_metadata, [46](#)
- cldc\_del, [46](#)
- cldc\_dirent\_count, [46](#)
- cldc\_dirent\_cur\_fini, [46](#)
- cldc\_dirent\_cur\_init, [46](#)
- cldc\_dirent\_first, [46](#)
- cldc\_dirent\_name, [46](#)
- cldc\_dirent\_next, [46](#)
- cldc\_end\_sess, [46](#)
- cldc\_get, [46](#)
- cldc\_getaddr, [46](#)
- cldc\_init, [46](#)
- cldc\_kill\_sess, [46](#)
- cldc\_lock, [46](#)
- cldc\_new\_sess, [46](#)
- cldc\_nop, [46](#)
- cldc\_open, [46](#)
- cldc\_put, [46](#)
- cldc\_receive\_pkt, [46](#)
- cldc\_saveaddr, [47](#)
- cldc\_udp\_free, [47](#)
- cldc\_udp\_new, [47](#)
- cldc\_udp\_pkt\_send, [47](#)
- cldc\_udp\_receive\_pkt, [47](#)
- cldc\_unlock, [47](#)
- cldc\_call\_opts, [9](#)
  - cb, [10](#)
  - private, [10](#)
  - resp, [10](#)
- cldc\_close
  - cldc.h, [46](#)
- cldc\_copts\_get\_data
  - cldc.h, [46](#)
- cldc\_copts\_get\_metadata
  - cldc.h, [46](#)
- cldc\_del
  - cldc.h, [46](#)
- cldc\_dirent\_count
  - cldc.h, [46](#)
- cldc\_dirent\_cur\_fini
  - cldc.h, [46](#)
- cldc\_dirent\_cur\_init
  - cldc.h, [46](#)
- cldc\_dirent\_first
  - cldc.h, [46](#)
- cldc\_dirent\_name
  - cldc.h, [46](#)
- cldc\_dirent\_next
  - cldc.h, [46](#)
- cldc.h, [46](#)
- cldc\_end\_sess
  - cldc.h, [46](#)
- cldc\_fh, [10](#)
  - fh, [11](#)
  - sess, [11](#)
  - valid, [11](#)
- cldc\_get
  - cldc.h, [46](#)
- cldc\_getaddr
  - cldc.h, [46](#)
- cldc\_host, [11](#)
  - host, [11](#)
  - port, [11](#)
  - prio, [11](#)
  - weight, [11](#)
- cldc\_init
  - cldc.h, [46](#)
- cldc\_kill\_sess
  - cldc.h, [46](#)
- cldc\_lock
  - cldc.h, [46](#)
- cldc\_msg, [12](#)
  - cb, [12](#)
  - cb\_private, [12](#)
  - copts, [12](#)
  - done, [12](#)
  - expire\_time, [12](#)
  - n\_pkts, [12](#)
  - op, [12](#)
  - pkt\_info, [12](#)
  - sess, [12](#)
  - xid, [12](#)
- cldc\_new\_sess
  - cldc.h, [46](#)
- cldc\_node\_metadata, [13](#)
  - flags, [13](#)
  - inode\_name, [13](#)
  - inum, [13](#)
  - time\_create, [13](#)
  - time\_modify, [13](#)
  - vers, [13](#)
- cldc\_nop
  - cldc.h, [46](#)
- cldc\_open
  - cldc.h, [46](#)
- cldc\_ops, [13](#)
  - event, [14](#)
  - pkt\_send, [14](#)
  - timer\_ctl, [14](#)

- cldc\_pkt\_info, 14
  - data, 15
  - hdr\_len, 15
  - pkt\_len, 15
  - retries, 15
  - user, 15
- cldc\_put
  - cldc.h, 46
- cldc\_receive\_pkt
  - cldc.h, 46
- cldc\_saveaddr
  - cldc.h, 47
- cldc\_session, 15
  - addr, 17
  - addr\_len, 17
  - cfh, 17
  - confirmed, 17
  - expire\_time, 17
  - expired, 17
  - inode\_name\_temp, 17
  - log, 17
  - msg\_buf, 17
  - msg\_buf\_len, 17
  - msg\_buf\_op, 17
  - msg\_scan\_time, 17
  - next\_seqid\_in, 17
  - next\_seqid\_in\_tr, 17
  - next\_seqid\_out, 17
  - ops, 17
  - out\_msg, 17
  - payload, 17
  - private, 17
  - secret\_key, 17
  - sid, 17
  - user, 17
- cldc\_udp, 18
  - addr, 18
  - addr\_len, 18
  - cb, 18
  - cb\_private, 18
  - fd, 18
  - sess, 18
- cldc\_udp\_free
  - cldc.h, 47
- cldc\_udp\_new
  - cldc.h, 47
- cldc\_udp\_pkt\_send
  - cldc.h, 47
- cldc\_udp\_receive\_pkt
  - cldc.h, 47
- cldc\_unlock
  - cldc.h, 47
- common\_pfx
  - hstor\_keylist, 21
- cond
  - ncld\_sess, 28
- confirmed
  - cldc\_session, 17
- contents
  - hstor\_keylist, 21
  - st\_keylist, 30
- copts
  - cldc\_msg, 12
- count
  - chunk\_check\_status, 5
- curl
  - hstor\_client, 20
- data
  - cldc\_pkt\_info, 15
- data\_len
  - chunksrv\_req, 6
  - chunksrv\_resp, 7
- debug
  - hail\_log, 19
- delim
  - hstor\_keylist, 21
- done
  - cldc\_msg, 12
- elist.h
  - INIT\_LIST\_HEAD, 48
  - list\_entry, 48
  - list\_for\_each, 48
  - list\_for\_each\_entry, 48
  - list\_for\_each\_entry\_continue, 49
  - list\_for\_each\_entry\_safe, 49
  - list\_for\_each\_prev, 49
  - list\_for\_each\_safe, 49
  - LIST\_HEAD, 50
  - LIST\_HEAD\_INIT, 50
- errc
  - ncld\_fh, 26
  - ncld\_read, 27
  - ncld\_sess, 28
- etag
  - hstor\_object, 22
  - st\_object, 31
- event
  - cldc\_ops, 14

- ncld\_sess, 28
- event\_arg
  - ncld\_fh, 26
  - ncld\_sess, 28
- event\_func
  - ncld\_fh, 26
- event\_mask
  - ncld\_fh, 26
- expire\_time
  - cldc\_msg, 12
  - cldc\_session, 17
- expired
  - cldc\_session, 17
- expires
  - cld\_timer, 9
- fd
  - cldc\_udp, 18
  - st\_client, 30
- fh
  - cldc\_fh, 11
  - ncld\_fh, 26
  - ncld\_read, 27
- fired
  - cld\_timer, 9
- flags
  - chunksrv\_req, 6
  - cldc\_node\_metadata, 13
  - objcache\_entry, 29
- fragment
  - http\_uri, 24
- fragment\_len
  - http\_uri, 24
- func
  - hail\_log, 19
- HAIL\_CRIT
  - hail\_log.h, 51
- HAIL\_DEBUG
  - hail\_log.h, 51
- HAIL\_ERR
  - hail\_log.h, 51
- HAIL\_INFO
  - hail\_log.h, 51
- hail\_log, 18
  - debug, 19
  - func, 19
  - verbose, 19
- hail\_log.h
  - ATTR\_PRINTF, 51
  - HAIL\_CRIT, 51
  - HAIL\_DEBUG, 51
  - HAIL\_ERR, 51
  - HAIL\_INFO, 51
  - HAIL\_VERBOSE, 51
  - HAIL\_WARN, 51
- hail\_private.h
  - xdr\_sizeof, 52
- HAIL\_VERBOSE
  - hail\_log.h, 51
- HAIL\_WARN
  - hail\_log.h, 51
- handles
  - ncld\_sess, 28
- hash
  - chunksrv\_resp, 7
  - objcache\_entry, 29
- hdr
  - http\_req, 23
- hdr\_len
  - cldc\_pkt\_info, 15
- host
  - cldc\_host, 11
  - hstor\_client, 20
  - ncld\_sess, 28
  - st\_client, 30
- hostname
  - http\_uri, 24
- hostname\_len
  - http\_uri, 24
- HREQ\_MAX\_HDR
  - hstor.h, 54
- hreq\_acl\_canned
  - hstor.h, 56
- hreq\_free
  - hstor.h, 56
- hreq\_hdr
  - hstor.h, 56
- hreq\_hdr\_push
  - hstor.h, 56
- hreq\_is\_query
  - hstor.h, 56
- hreq\_query
  - hstor.h, 56
- hreq\_sign
  - hstor.h, 56
- hstor.h
  - ACLC\_AUTH\_R, 54
  - ACLC\_PRIV, 54
  - ACLC\_PUB\_R, 54

- ACLC\_PUB\_RW, 54
- ACLCNUM, 54
- ARRAY\_SIZE, 54
- HREQ\_MAX\_HDR, 54
- hreq\_acl\_canned, 56
- hreq\_free, 56
- hreq\_hdr, 56
- hreq\_hdr\_push, 56
- hreq\_is\_query, 56
- hreq\_query, 56
- hreq\_sign, 56
- hstor\_add\_bucket, 56
- hstor\_del, 56
- hstor\_del\_bucket, 56
- hstor\_free, 56
- hstor\_free\_blist, 56
- hstor\_free\_bucket, 56
- hstor\_free\_keylist, 56
- hstor\_free\_object, 56
- hstor\_get, 56
- hstor\_get\_inline, 56
- hstor\_keys, 56
- hstor\_list\_buckets, 56
- hstor\_new, 56
- hstor\_put, 56
- hstor\_put\_inline, 56
- huri\_field\_escape, 56
- huri\_field\_unescape, 56
- huri\_parse, 56
- hutil\_str2time, 56
- hutil\_time2str, 56
- PATH\_ESCAPE\_MASK, 54
- QUERY\_ESCAPE\_MASK, 54
- ReqACLC, 54
- ReqQ, 54
- URIQ\_ACL, 54
- URIQ\_LOCATION, 54
- URIQ\_LOGGING, 54
- URIQ\_TORRENT, 54
- URIQNUM, 54
- hstor\_add\_bucket
  - hstor.h, 56
- hstor\_blist, 19
  - list, 19
  - own\_id, 19
  - own\_name, 19
- hstor\_bucket, 19
  - name, 20
  - time\_create, 20
- hstor\_client, 20
  - acc, 20
  - curl, 20
  - host, 20
  - key, 20
  - user, 20
  - verbose, 20
- hstor\_del
  - hstor.h, 56
- hstor\_del\_bucket
  - hstor.h, 56
- hstor\_free
  - hstor.h, 56
- hstor\_free\_blist
  - hstor.h, 56
- hstor\_free\_bucket
  - hstor.h, 56
- hstor\_free\_keylist
  - hstor.h, 56
- hstor\_free\_object
  - hstor.h, 56
- hstor\_get
  - hstor.h, 56
- hstor\_get\_inline
  - hstor.h, 56
- hstor\_keylist, 21
  - common\_pfx, 21
  - contents, 21
  - delim, 21
  - marker, 21
  - max\_keys, 21
  - name, 21
  - prefix, 21
  - trunc, 21
- hstor\_keys
  - hstor.h, 56
- hstor\_list\_buckets
  - hstor.h, 56
- hstor\_new
  - hstor.h, 56
- hstor\_object, 21
  - etag, 22
  - key, 22
  - own\_id, 22
  - own\_name, 22
  - size, 22
  - storage, 22
  - time\_mod, 22
- hstor\_put
  - hstor.h, 56
- hstor\_put\_inline

- hstor.h, 56
- http\_hdr, 22
  - key, 22
  - val, 22
- http\_req, 23
  - hdr, 23
  - major, 23
  - method, 23
  - minor, 23
  - n\_hdr, 23
  - orig\_path, 23
  - uri, 23
- http\_uri, 23
  - fragment, 24
  - fragment\_len, 24
  - hostname, 24
  - hostname\_len, 24
  - path, 24
  - path\_len, 24
  - port, 24
  - query, 24
  - query\_len, 24
  - scheme, 24
  - scheme\_len, 24
  - userinfo, 24
  - userinfo\_len, 24
- huri\_field\_escape
  - hstor.h, 56
- huri\_field\_unescape
  - hstor.h, 56
- huri\_parse
  - hstor.h, 56
- hutil\_str2time
  - hstor.h, 56
- hutil\_time2str
  - hstor.h, 56
- include/chunk-private.h, 33
- include/chunk\_msg.h, 33
- include/chunkc.h, 36
- include/chunksrv.h, 39
- include/cld-private.h, 40
- include/cld\_common.h, 40
- include/cldc.h, 42
- include/elist.h, 47
- include/hail\_log.h, 50
- include/hail\_private.h, 51
- include/hstor.h, 52
- include/ncl.d.h, 56
- include/objcache.h, 58
- INIT\_LIST\_HEAD
  - elist.h, 48
- inode\_name
  - cldc\_node\_metadata, 13
- inode\_name\_temp
  - cldc\_session, 17
- inum
  - cldc\_node\_metadata, 13
- is\_done
  - ncl.d\_read, 27
- is\_open
  - ncl.d\_fh, 26
- is\_up
  - ncl.d\_sess, 28
- key
  - hstor\_client, 20
  - hstor\_object, 22
  - http\_hdr, 22
  - st\_client, 30
- key\_len
  - chunksrv\_req, 6
- lastdone
  - chunk\_check\_status, 5
- length
  - ncl.d\_read, 27
- list
  - cld\_timer\_list, 9
  - hstor\_blist, 19
- list\_entry
  - elist.h, 48
- list\_for\_each
  - elist.h, 48
- list\_for\_each\_entry
  - elist.h, 48
- list\_for\_each\_entry\_continue
  - elist.h, 49
- list\_for\_each\_entry\_safe
  - elist.h, 49
- list\_for\_each\_prev
  - elist.h, 49
- list\_for\_each\_safe
  - elist.h, 49
- LIST\_HEAD
  - elist.h, 50
- list\_head, 24
  - next, 25
  - prev, 25
- LIST\_HEAD\_INIT

- elist.h, 50
- lock
  - objcache, 29
- log
  - cldc\_session, 17
- magic
  - chunksrv\_req, 6
  - chunksrv\_resp, 7
- major
  - http\_req, 23
- marker
  - hstor\_keylist, 21
- max\_keys
  - hstor\_keylist, 21
- MDB\_TPATH\_FMT
  - chunk-private.h, 33
- meta
  - ncld\_read, 27
- method
  - http\_req, 23
- minor
  - http\_req, 23
- msg\_buf
  - cldc\_session, 17
- msg\_buf\_len
  - cldc\_session, 17
- msg\_buf\_op
  - cldc\_session, 17
- msg\_scan\_time
  - cldc\_session, 17
- mtime
  - chunksrv\_resp\_get, 8
- mutex
  - ncld\_sess, 28
- n\_hdr
  - http\_req, 23
- n\_pkts
  - cldc\_msg, 12
- name
  - cld\_timer, 9
  - hstor\_bucket, 20
  - hstor\_keylist, 21
  - st\_keylist, 30
  - st\_object, 31
- ncld.h
  - ncld\_close, 58
  - ncld\_del, 58
  - ncld\_get, 58
  - ncld\_get\_meta, 58
  - ncld\_init, 58
  - ncld\_open, 58
  - ncld\_qlock, 58
  - ncld\_read\_free, 58
  - ncld\_sess\_close, 58
  - ncld\_sess\_open, 58
  - ncld\_trylock, 58
  - ncld\_unlock, 58
  - ncld\_write, 58
- ncld\_close
  - ncld.h, 58
- ncld\_del
  - ncld.h, 58
- ncld\_fh, 25
  - errc, 26
  - event\_arg, 26
  - event\_func, 26
  - event\_mask, 26
  - fh, 26
  - is\_open, 26
  - nios, 26
  - sess, 26
- ncld\_get
  - ncld.h, 58
- ncld\_get\_meta
  - ncld.h, 58
- ncld\_init
  - ncld.h, 58
- ncld\_open
  - ncld.h, 58
- ncld\_qlock
  - ncld.h, 58
- ncld\_read, 26
  - errc, 27
  - fh, 27
  - is\_done, 27
  - length, 27
  - meta, 27
  - ptr, 27
- ncld\_read\_free
  - ncld.h, 58
- ncld\_sess, 27
  - cond, 28
  - errc, 28
  - event, 28
  - event\_arg, 28
  - handles, 28
  - host, 28
  - is\_up, 28

- mutex, 28
- open\_done, 28
- port, 28
- thread, 28
- tlist, 28
- to\_thread, 28
- udp, 28
- udp\_timer, 28
- ncld\_sess\_close
  - ncld.h, 58
- ncld\_sess\_open
  - ncld.h, 58
- ncld\_trylock
  - ncld.h, 58
- ncld\_unlock
  - ncld.h, 58
- ncld\_write
  - ncld.h, 58
- next
  - list\_head, 25
- next\_seqid\_in
  - cldc\_session, 17
- next\_seqid\_in\_tr
  - cldc\_session, 17
- next\_seqid\_out
  - cldc\_session, 17
- nios
  - ncld\_fh, 26
- nonce
  - chunksrv\_req, 6
  - chunksrv\_resp, 7
- objcache, 28
  - lock, 29
  - table, 29
- objcache.h
  - \_\_objcache\_get, 59
  - objcache\_count, 59
  - objcache\_fini, 59
  - objcache\_get, 59
  - objcache\_get\_dirty, 59
  - objcache\_init, 59
  - objcache\_put, 59
  - objcache\_test\_dirty, 59
  - OC\_F\_DIRTY, 59
- objcache\_count
  - objcache.h, 59
- objcache\_entry, 29
  - flags, 29
  - hash, 29
  - ref, 29
- objcache\_fini
  - objcache.h, 59
- objcache\_get
  - objcache.h, 59
- objcache\_get\_dirty
  - objcache.h, 59
- objcache\_init
  - objcache.h, 59
- objcache\_put
  - objcache.h, 59
- objcache\_test\_dirty
  - objcache.h, 59
- OC\_F\_DIRTY
  - objcache.h, 59
- on\_list
  - cld\_timer, 9
- op
  - chunksrv\_req, 6
  - cldc\_msg, 12
- open\_done
  - ncld\_sess, 28
- ops
  - cldc\_session, 17
- orig\_path
  - http\_req, 23
- out\_msg
  - cldc\_session, 17
- own\_id
  - hstor\_blist, 19
  - hstor\_object, 22
- own\_name
  - hstor\_blist, 19
  - hstor\_object, 22
- owner
  - st\_object, 31
- p
  - cld\_dirent\_cur, 8
- pad
  - chunk\_check\_status, 5
- path
  - http\_uri, 24
- PATH\_ESCAPE\_MASK
  - hstor.h, 54
- path\_len
  - http\_uri, 24
- payload
  - cldc\_session, 17
- PKT\_HDR\_TO\_STR\_SCRATCH\_LEN

- cld\_common.h, 41
- pkt\_info
  - cldc\_msg, 12
- pkt\_len
  - cldc\_pkt\_info, 15
- pkt\_send
  - cldc\_ops, 14
- port
  - cldc\_host, 11
  - http\_uri, 24
  - ncld\_sess, 28
- prefix
  - hstor\_keylist, 21
- PREFIX\_LEN
  - chunk-private.h, 33
- prev
  - list\_head, 25
- prio
  - cldc\_host, 11
- private
  - cldc\_call\_opts, 10
  - cldc\_session, 17
- ptr
  - ncld\_read, 27
- query
  - http\_uri, 24
- QUERY\_ESCAPE\_MASK
  - hstor.h, 54
- query\_len
  - http\_uri, 24
- ref
  - objcache\_entry, 29
- req\_buf
  - st\_client, 30
- req\_len
  - chunksrv.h, 40
- ReqACLC
  - hstor.h, 54
- ReqQ
  - hstor.h, 54
- resp
  - chunksrv\_resp\_chkstat, 7
  - chunksrv\_resp\_get, 8
  - cldc\_call\_opts, 10
- resp\_code
  - chunksrv\_resp, 7
- retries
  - cldc\_pkt\_info, 15
- rsv1
  - chunksrv\_resp, 7
- runmark
  - cld\_timer\_list, 9
- scheme
  - http\_uri, 24
- scheme\_len
  - http\_uri, 24
- secret\_key
  - cldc\_session, 17
- sess
  - cldc\_fh, 11
  - cldc\_msg, 12
  - cldc\_udp, 18
  - ncld\_fh, 26
- sid
  - cldc\_session, 17
- SIDARG
  - cld\_common.h, 41
- SIDFMT
  - cld\_common.h, 41
- sig
  - chunksrv\_req, 6
- size
  - hstor\_object, 22
  - st\_object, 31
- ssl
  - st\_client, 30
- ssl\_ctx
  - st\_client, 30
- st\_client, 29
  - fd, 30
  - host, 30
  - key, 30
  - req\_buf, 30
  - ssl, 30
  - ssl\_ctx, 30
  - user, 30
  - verbose, 30
- st\_keylist, 30
  - contents, 30
  - name, 30
- st\_object, 31
  - etag, 31
  - name, 31
  - owner, 31
  - size, 31
  - time\_mod, 31
- state

- chunk\_check\_status, 5
- stc\_check\_start
  - chunkc.h, 39
- stc\_check\_status
  - chunkc.h, 39
- stc\_cp
  - chunkc.h, 39
- stc\_del
  - chunkc.h, 39
- stc\_free
  - chunkc.h, 39
- stc\_free\_keylist
  - chunkc.h, 39
- stc\_free\_object
  - chunkc.h, 39
- stc\_get
  - chunkc.h, 39
- stc\_get\_inline
  - chunkc.h, 39
- stc\_get\_recv
  - chunkc.h, 39
- stc\_get\_start
  - chunkc.h, 39
- stc\_init
  - chunkc.h, 39
- stc\_keys
  - chunkc.h, 39
- stc\_new
  - chunkc.h, 39
- stc\_ping
  - chunkc.h, 39
- stc\_put
  - chunkc.h, 39
- stc\_put\_inline
  - chunkc.h, 39
- stc\_put\_send
  - chunkc.h, 39
- stc\_put\_start
  - chunkc.h, 39
- stc\_put\_sync
  - chunkc.h, 39
- stc\_readport
  - chunkc.h, 39
- stc\_table\_open
  - chunkc.h, 39
- storage
  - hstor\_object, 22
- table
  - objcache, 29
- thread
  - ncld\_sess, 28
- time\_create
  - cldc\_node\_metadata, 13
  - hstor\_bucket, 20
- time\_mod
  - hstor\_object, 22
  - st\_object, 31
- time\_modify
  - cldc\_node\_metadata, 13
- timer\_ctl
  - cldc\_ops, 14
- tlist
  - ncld\_sess, 28
- tmp\_len
  - cld\_dirent\_cur, 8
- to\_thread
  - ncld\_sess, 28
- trunc
  - hstor\_keylist, 21
- udp
  - ncld\_sess, 28
- udp\_timer
  - ncld\_sess, 28
- uri
  - http\_req, 23
- URIQ\_ACL
  - hstor.h, 54
- URIQ\_LOCATION
  - hstor.h, 54
- URIQ\_LOGGING
  - hstor.h, 54
- URIQ\_TORRENT
  - hstor.h, 54
- URIQNUM
  - hstor.h, 54
- user
  - cldc\_pkt\_info, 15
  - cldc\_session, 17
  - hstor\_client, 20
  - st\_client, 30
- userdata
  - cld\_timer, 9
- userinfo
  - http\_uri, 24
- userinfo\_len
  - http\_uri, 24
- val

- http\_hdr, [22](#)
- valid
  - cldc\_fh, [11](#)
- verbose
  - hail\_log, [19](#)
  - hstor\_client, [20](#)
  - st\_client, [30](#)
- vers
  - cldc\_node\_metadata, [13](#)
- weight
  - cldc\_host, [11](#)
- xdr\_sizeof
  - hail\_private.h, [52](#)
- xid
  - cldc\_msg, [12](#)