

# IDA Pro: lc command reference manual

## Table of Contents

1. Introduction .....	1
2. Command-line options .....	1
3. Commands .....	3
3.1. Operating with metadata .....	3
3.1.1. <code>hist show</code> .....	3
3.1.2. <code>hist pushes</code> .....	7
3.1.3. <code>hist del</code> .....	8
3.2. Various information .....	12
3.2.1. <code>info</code> .....	12
3.2.2. <code>users</code> .....	12
3.2.3. <code>stats</code> .....	12
3.3. Administrative commands .....	14
3.3.1. Managing users .....	14
3.3.1.1. <code>user add</code> .....	14
3.3.1.2. <code>user edit</code> .....	14
3.3.1.3. <code>user del</code> .....	14
3.3.1.4. <code>passwd</code> .....	15
3.3.2. Managing sessions .....	15
3.3.2.1. <code>session list</code> .....	15
3.3.2.2. <code>session kill</code> .....	15
4. Concepts .....	17
4.1. What is the Lumina server .....	17
4.1.1. Functions metadata .....	17
4.1.2. Metadata contents .....	17
4.1.3. Pushing & overriding metadata .....	17
4.1.4. Metadata history .....	17
4.1.5. File contents .....	17
Appendix A: Commands .....	18
A.1. String patterns .....	18
A.2. Timerange formats .....	18

Last updated on October 07, 2022 – v8.1

## 1. Introduction

The `lc` executable provides a command-line interface to interact with a Lumina server and its contents.

**NOTE** | virtually all the commands described in this document, require administrator rights; "regular" users will typically not have the necessary privileges.

## 2. Command-line options

**-h, --host HOSTNAME[:PORT]**

Lumina host name and port (if port is omitted, it defaults to 65432)

**-u USERNAME**

specify username

**-p PASSWORD**

specify password

**-v, --verbose**

verbose output

In order to connect to a Lumina server, lc must at least be provided with a hostname and a valid user-password pair.

**TIP**

In order to keep the various commands' syntax as clear as possible, we will omit the login options from commands for the rest of this manual.

Other options exists, specific to each lc command (see [Commands](#)).

## 3. Commands

The following commands are accepted by lc:

### 3.1. Operating with metadata

Commands in this section let users view metadata stored in the Lumina server and their history.

#### 3.1.1. `hist show`

`hist show [OPTION]`

Queries history of changes for function(s).

The following informations will be displayed for each change:

- The ID of the change
- The timestamp of the change
- The ID of the push that contains the change
- The name of the function at that change (+ (\*) if it has been modified past this change)
- optional: The username of the user that pushed this change
- optional: The name of the license associated with the push for this change
- optional: The email of the license associated with the push for this change
- optional: The ID of the license associated with the push for this change
- optional: The ID of the function
- optional: The effective address (EA) of the function in the input file for the change
- optional: The hash of the function
- optional: The path of the idb file where the change came from
- optional: The hash of the file where the change came from
- optional: The path of the file where the change came from

**TIP** | Wilcards can be used to facilitate the usage of options that take strings as input. See the [appendix](#).

*Options*

**-a, --additional-fields LIST**

Comma-delimited list of additional info to display (`username`, `license_name`, `license_email`, `license_id`, `func_id`, `func_ea`, `calcrel_hash`, `idb_path`, `input_hash`, `input_path`, `all`)

**-d, --details**

Show details (diff-like) for each change

**--chronological**

Display entries in chronological order (defaults to reverse-chronological).

**-m, --max-entries NUMBER**

maximum number of entries to fetch (defaults to 100)

**-l, --license-id LICENSE**

license ID (XX-XXXX-XXXX-XX format) to operate on

**-r, --history-id-range RANGE**

history ID range(s) to operate on (`start0..end0 […]`)

**-t, --time-range RANGE**

time range to operate on (`start..end`) see the [appendix](#)

**-i, --idb IDB**

IDB name(s) to operate on

**-f, --input FILE**

input file(s) to operate on

**-u, --username USERNAME**

username(s) to operate on

**-n, --func NAME**

function name(s) to operate on

**-h, --input-hash HASH**

input file hash(es) to operate on

**-p, --pushes-id-range RANGE**Pushes ID range(s) to operate on (`start0..end0 [...]`)**-c, --calcrel-hash HASH**

function hash(es) to operate on

**--last-change**

Select only the last change for a function

Examples:

*List the last 8 changes ("`-m 8`" specifies the number of changes to show; the default order is reverse-chronological)*

```
alice@alice_PC$ lc hist show -m 8
Change Time          Push Func name
-----
507  2022-09-15 14:48:18 5  math_things
506  2022-09-15 14:48:17 4  calc_things (*)
505  2022-09-15 14:48:17 4  start
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a
503  2022-09-15 14:48:16 3  display_keygen_window
502  2022-09-15 14:48:15 2  fstat
501  2022-09-15 14:48:15 2  __umoddi3
500  2022-09-15 14:48:15 2  __udivdi3
# Shown 8 results (more are available...)
```

*List changes from id 9 up to (but excluding) id 14*

```
alice@alice_PC$ lc hist show -r9..14
Change Time          Push Func name
-----
13  2022-09-15 14:48:05 2  AllocateBucketGroup
12  2022-09-15 14:48:05 2  addr_map_find
11  2022-09-15 14:48:05 2  addr_map_insert
10  2022-09-15 14:48:05 2  addr_map_compare_func
9   2022-09-15 14:48:05 2  addr_map_free_func
# Shown 5 results
```

*List changes using multiple ranges (`9..14,505..515`; in this case, there were no changes after 507 in the database)*

```
alice@alice_PC$ lc hist show -r9..14,505..515
Change Time          Push Func name
-----
507  2022-09-15 14:48:18 5  math_things
506  2022-09-15 14:48:17 4  calc_things (*)
505  2022-09-15 14:48:17 4  start
13  2022-09-15 14:48:05 2  AllocateBucketGroup
12  2022-09-15 14:48:05 2  addr_map_find
11  2022-09-15 14:48:05 2  addr_map_insert
```

```

10  2022-09-15 14:48:05 2  addr_map_compare_func
9   2022-09-15 14:48:05 2  addr_map_free_func
# Shown 8 results

```

Find changes from a specific idb file ("-i"), showing the function hash and ea ("-a" adds additional columns to the output)

```

alice@alice_PC$ lc hist show -i C:\malware\apts\pc_keygenme_3.pe.idb -a calcrel_hash,func_ea
Change Time          Push Func name      Func EA Func hash
-----
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a 4011e2 420A0485EDB7C6774E1822953FB785D4
503  2022-09-15 14:48:16 3  display_keygen_window      401099 ACA673D198FBC0DF24C20A15FF7F25CE
# Shown 2 results

```

Show the first 4 changes ("-m4") with their input file(s) (in "--chronological" order; "-a" adds an additional column)

```

alice@alice_PC$ lc hist show --chronological -m 4 -a input_path
Change Time          Push Func name      Input path
-----
1   2022-09-15 14:47:44 1  .init_proc           /home/alice/work/pc_dwarf_arrays.elf
2   2022-09-15 14:47:44 1  _start               /home/alice/work/pc_dwarf_arrays.elf
3   2022-09-15 14:47:44 1  __do_global_dtors_aux /home/alice/work/pc_dwarf_arrays.elf
4   2022-09-15 14:47:44 1  frame_dummy          /home/alice/work/pc_dwarf_arrays.elf
# Shown 4 results (more are available...)

```

Show the last change by a user ("-u" indicates the user, "-m1" means show 1 change only, "-a" adds an additional column)

```

alice@alice_PC$ lc hist show -ubob -m1 -a username
Change Time          Push Func name      Username
-----
502  2022-09-15 14:48:15 2  fstat                bob
# Shown 1 results (more are available...)

```

Show up to 4 changes ("-m4") between two dates ("-t" indicates a range of "YYYY-MM-DD" dates)

```

alice@alice_PC$ lc hist show -m4 -t2022-09-02..2022-12-03
Change Time          Push Func name
-----
507  2022-09-15 14:48:18 5  math_things
506  2022-09-15 14:48:17 4  calc_things (*)
505  2022-09-15 14:48:17 4  start
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a
# Shown 4 results (more are available...)

```

Show up to 4 changes ("-m4") between two "now"-relative dates ("-t", from 2 weeks ago to 5 minutes ago) ("-a" adds an additional column)

```

alice@alice_PC$ lc hist show -m4 -t-2w..-5M -a func_id
Change Time          Push Func name      Func ID
-----
507  2022-09-15 14:48:18 5  math_things         506
506  2022-09-15 14:48:17 4  calc_things (*)     506
505  2022-09-15 14:48:17 4  start                505
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a 504
# Shown 4 results (more are available...)

```

Show up to 4 changes ("-m4") between two "now"-relative dates ("-t", from 2 weeks ago to 5 minutes ago) ("-a" adds an additional column), only select the last change for each function

```

alice@alice_PC$ lc hist show -m4 -t-2w..-5M -a func_id --last-change
Change Time          Push Func name      Func ID
-----
507  2022-09-15 14:48:18 5  math_things         506
505  2022-09-15 14:48:17 4  start                505
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a 504
503  2022-09-15 14:48:16 3  display_keygen_window 503

```

```
# Shown 4 results (more are available...)
```

Show up to 4 changes ("-m4"), occurring after a "now"-relative date ("-t", from 6 hours ago up to now)

```
alice@alice_PC$ lc hist show -m4 -t-6H
Change Time          Push Func name
-----
507  2022-09-15 14:48:18 5  math_things
506  2022-09-15 14:48:17 4  calc_things (*)
505  2022-09-15 14:48:17 4  start
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a
# Shown 4 results (more are available...)
```

Show changes about a specific function name ("-n")

```
alice@alice_PC$ lc hist show -n print_relocinfo
Change Time          Push Func name
-----
175  2022-09-15 14:48:08 2  print_relocinfo
# Shown 1 results
```

Show changes about a function name ("-n") searched by wildcard ("like:...")

```
alice@alice_PC$ lc hist show -n "like:%reloc%info%"
Change Time          Push Func name
-----
179  2022-09-15 14:48:08 2  print_reloc_information_32
178  2022-09-15 14:48:08 2  print_reloc_information_64
177  2022-09-15 14:48:08 2  print_relocinfo_32
176  2022-09-15 14:48:08 2  print_relocinfo_64
175  2022-09-15 14:48:08 2  print_relocinfo
# Shown 5 results
```

Show metadata details ("--details") in changes for a function ("-n"); this particular change added a new function

```
alice@alice_PC$ lc hist show -n is_location_form --details
Change Time          Push Func name
-----
126  2022-09-15 14:48:07 2  is_location_form
>> Score
>>   - 0
>>   + 1445
>> Name
>>   - None
>>   + is_location_form
>> Prototype
>>   -
>>   + 0C303D08626F6F6C65616E0207
>> Member @ 0x8
>>   .type
>>     - None
>>     + 07
>>   .cmt
>>     - None
>>     +
>>   .rptcmt
>>     - None
>>     +
>> Insn operands @ 0+0x3
>>   - [<no ops repr>]
>>   + [op0=0xb, op1=0x0]
>> Insn operands @ 0+0x9
>>   - [<no ops repr>]
>>   + [op0=0xb, op1=0x0]
>> Insn operands @ 0+0x15
>>   - [<no ops repr>]
>>   + [op0=0xb, op1=0x0]
```

```

>> Insn operands @ 0+0x21
>> - [<no ops repr>]
>> + [op0=0xb, op1=0x0]
>> Insn operands @ 0+0x27
>> - [<no ops repr>]
>> + [op0=0xb, op1=0x0]
>> Insn operands @ 0+0x33
>> - [<no ops repr>]
>> + [op0=0xb, op1=0x0]
>> Insn operands @ 0+0x39
>> - [<no ops repr>]
>> + [op0=0xb, op1=0x0]
# Shown 1 results

```

### 3.1.2. hist pushes

#### hist pushes [OPTION]

Shows pushes to the Lumina server.

##### Options

##### -a, --additional-fields LIST

Comma-delimited list of additional info to display (`license_name`, `license_email`, `license_id`, `all`)

##### -t, --time-range TIMESTAMP

timestamp

##### -u, --username USERNAME

username(s) to operate on

##### -l, --license-id LICENSE

license ID (XX-XXXX-XXXX-XX format) to operate on

##### -m, --max-entries NUMBER

maximum number of entries to operate on (defaults to 100)

##### --chronological

Display entries in chronological order (defaults to reverse-chronological).

Examples:

```

alice@alice_PC$ lc hist pushes
  Push ID Time                User name IDB path
-----
  5      2022-09-15 14:48:18  alice      /home/alice/work/pc_math_b_64.elf.i64
  4      2022-09-15 14:48:17  alice      /home/alice/work/pc_math_a_64.elf.i64
  3      2022-09-15 14:48:16  damian     C:\malware\apts\pc_keygenme_3.pe.idb
  2      2022-09-15 14:48:05  bob        /Users/bob/idbs/pc_dwarfdump.elf.idb
  1      2022-09-15 14:47:44  alice      /home/alice/work/pc_dwarf_arrays.elf.idb
# Shown 5 results

```

List all pushes from a specific license ID

```

alice@alice_PC$ lc hist pushes -l BB-0B0B-AC8E-01 -a license_email
  Push ID Time                User name License email IDB path
-----
  2      2022-09-15 14:48:05  bob        bob@acme.com /Users/bob/idbs/pc_dwarfdump.elf.idb
# Shown 1 results

```

List all pushes from licenses with IDs matching a pattern ("-a" adds an additional column)

```
alice@alice_PC$ lc hist pushes -l like:AA-% -a license_id
Push ID Time          User name License ID      IDB path
-----
5      2022-09-15 14:48:18 alice    AA-A11C-AC8E-01 /home/alice/work/pc_math_b_64.elf.i64
4      2022-09-15 14:48:17 alice    AA-A11C-AC8E-01 /home/alice/work/pc_math_a_64.elf.i64
1      2022-09-15 14:47:44 alice    AA-A11C-AC8E-01 /home/alice/work/pc_dwarf_arrays.elf.idb
# Shown 3 results
```

Show the first push

```
alice@alice_PC$ lc hist pushes --chronological -m 1
Push ID Time          User name IDB path
-----
1      2022-09-15 14:47:44 alice    /home/alice/work/pc_dwarf_arrays.elf.idb
# Shown 1 results (more are available...)
```

List all pushes between two timestamps

```
alice@alice_PC$ lc hist pushes -t"2022-09-07 10:20:00..2022-12-31"
Push ID Time          User name IDB path
-----
5      2022-09-15 14:48:18 alice    /home/alice/work/pc_math_b_64.elf.i64
4      2022-09-15 14:48:17 alice    /home/alice/work/pc_math_a_64.elf.i64
3      2022-09-15 14:48:16 damian   C:\malware\apts\pc_keygenme_3.pe.idb
2      2022-09-15 14:48:05 bob      /Users/bob/idbs/pc_dwarfdump.elf.idb
1      2022-09-15 14:47:44 alice    /home/alice/work/pc_dwarf_arrays.elf.idb
# Shown 5 results
```

List all pushes from two users ("-a" adds an additional column)

```
alice@alice_PC$ lc hist pushes -u"bob damian" -a"license_name license_id"
Push ID Time          User name License name License ID      IDB path
-----
3      2022-09-15 14:48:16 damian   Damian    DD-DA81-A000-01 C:\malware\apts\pc_keygenme_3.pe.idb
2      2022-09-15 14:48:05 bob      Bob      BB-0B0B-AC8E-01 /Users/bob/idbs/pc_dwarfdump.elf.idb
# Shown 2 results
```

### 3.1.3. hist del

#### hist del [OPTION]

Deletes history and metadata for functions.

Options

**-s, --silent**

Do not ask for confirmation before deleting history

**-l, --license-id LICENSE**

license ID (XX-XXXX-XXXX-XX format) to operate on

**-r, --history-id-range RANGE**

history ID range(s) to operate on (**start0..end0** [...])

**-t, --time-range RANGE**

time range to operate on (**start..end**) see the [appendix](#)

**-i, --idb IDB**

IDB name(s) to operate on



**-f, --input FILE**

input file(s) to operate on

**-u, --username USERNAME**

username(s) to operate on

**-n, --func NAME**

function name(s) to operate on

**-h, --input-hash HASH**

input file hash(es) to operate on

**-p, --pushes-id-range RANGE**Pushes ID range(s) to operate on (`start0..end0 [...]`)**-c, --calcrel-hash HASH**

function hash(es) to operate on

**--last-change**

Select only the last change for a function

Examples:

Display the last 10 changes, with their input file(s) and function ID(s)

```
alice@alice_PC$ show -m10 -a input_path
```

Change	Time	Push	Func name	Func ID	Input path
507	2022-09-15 14:48:18	5	math_things	506	/home/alice/work/pc_math_b_64.elf
506	2022-09-15 14:48:17	4	calc_things (*)	506	/home/alice/work/pc_math_a_64.elf
505	2022-09-15 14:48:17	4	start	505	/home/alice/work/pc_math_a_64.elf
504	2022-09-15 14:48:16	3	keygen_window_dialog_proc_a	504	C:\malware\apts\pc_keygenme_3.pe
503	2022-09-15 14:48:16	3	display_keygen_window	503	C:\malware\apts\pc_keygenme_3.pe
502	2022-09-15 14:48:15	2	fstat	502	/Users/bob/idbs/pc_dwarfdump.elf
501	2022-09-15 14:48:15	2	__umoddi3	501	/Users/bob/idbs/pc_dwarfdump.elf
500	2022-09-15 14:48:15	2	__udivdi3	500	/Users/bob/idbs/pc_dwarfdump.elf
499	2022-09-15 14:48:14	2	dwarf_get_ADDR_name	499	/Users/bob/idbs/pc_dwarfdump.elf
498	2022-09-15 14:48:14	2	dwarf_get_FRAME_name	498	/Users/bob/idbs/pc_dwarfdump.elf

# Shown 10 results (more are available...)

Delete all changes for functions matching a pattern

```
alice@alice_PC$ lc hist del -s -n like:%dwarf%
267 entries deleted from history
```

```
alice@alice_PC$ hist show -m10 -a input_path,func_id
```

Change	Time	Push	Func name	Func ID	Input path
507	2022-09-15 14:48:18	5	math_things	506	/home/alice/work/pc_math_b_64.elf
506	2022-09-15 14:48:17	4	calc_things (*)	506	/home/alice/work/pc_math_a_64.elf
505	2022-09-15 14:48:17	4	start	505	/home/alice/work/pc_math_a_64.elf
504	2022-09-15 14:48:16	3	keygen_window_dialog_proc_a	504	C:\malware\apts\pc_keygenme_3.pe
503	2022-09-15 14:48:16	3	display_keygen_window	503	C:\malware\apts\pc_keygenme_3.pe
502	2022-09-15 14:48:15	2	fstat	502	/Users/bob/idbs/pc_dwarfdump.elf
501	2022-09-15 14:48:15	2	__umoddi3	501	/Users/bob/idbs/pc_dwarfdump.elf
500	2022-09-15 14:48:15	2	__udivdi3	500	/Users/bob/idbs/pc_dwarfdump.elf
474	2022-09-15 14:48:14	2	free_macro_stack	474	/Users/bob/idbs/pc_dwarfdump.elf
468	2022-09-15 14:48:14	2	print_line_detail	468	/Users/bob/idbs/pc_dwarfdump.elf

# Shown 10 results (more are available...)

Delete all changes stemming from a specific file

```
alice@alice_PC$ lc hist del -s -f/Users/bob/idbs/pc_dwarfdump.elf
```

```
228 entries deleted from history
```

List the changes from pushes 1 to 5 (not included), showing their input file(s)

```
alice@alice_PC$ lc show -p1..5 -a input_path
Change Time          Push Func name      Input path
-----
506  2022-09-15 14:48:17 4  math_things        /home/alice/work/pc_math_a_64.elf
505  2022-09-15 14:48:17 4  start              /home/alice/work/pc_math_a_64.elf
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a C:\malware\apts\pc_keygenme_3.pe
503  2022-09-15 14:48:16 3  display_keygen_window C:\malware\apts\pc_keygenme_3.pe
7    2022-09-15 14:47:44 1  __do_global_ctors_aux /home/alice/work/pc_dwarf_arrays.elf
6    2022-09-15 14:47:44 1  __libc_csu_init      /home/alice/work/pc_dwarf_arrays.elf
5    2022-09-15 14:47:44 1  main                 /home/alice/work/pc_dwarf_arrays.elf
4    2022-09-15 14:47:44 1  frame_dummy         /home/alice/work/pc_dwarf_arrays.elf
3    2022-09-15 14:47:44 1  __do_global_dtors_aux /home/alice/work/pc_dwarf_arrays.elf
2    2022-09-15 14:47:44 1  _start              /home/alice/work/pc_dwarf_arrays.elf
1    2022-09-15 14:47:44 1  .init_proc          /home/alice/work/pc_dwarf_arrays.elf
# Shown 11 results
```

Delete one single change

```
alice@alice_PC$ lc hist del -s -r505..506
1 entries deleted from history
```

Delete all the changes from push 1

```
alice@alice_PC$ lc hist del -s -p 1..2
7 entries deleted from history

alice@alice_PC$ hist show -a username
Change Time          Push Func name      Username
-----
507  2022-09-15 14:48:18 5  math_things        alice
506  2022-09-15 14:48:17 4  calc_things (*)    alice
504  2022-09-15 14:48:16 3  keygen_window_dialog_proc_a damian
503  2022-09-15 14:48:16 3  display_keygen_window damian
# Shown 4 results
```

Delete all changes by a user

```
alice@alice_PC$ lc hist del -s -udamian
2 entries deleted from history

alice@alice_PC$ hist show -a func_id
Change Time          Push Func name      Func ID
-----
507  2022-09-15 14:48:18 5  math_things        506
506  2022-09-15 14:48:17 4  calc_things (*)    506
# Shown 2 results
```

Delete the last change for a function

```
alice@alice_PC$ lc hist del -s --func math_things --last-change
1 entries deleted from history

alice@alice_PC$ hist show -a func_id
Change Time          Push Func name      Func ID
-----
506  2022-09-15 14:48:17 4  calc_things 506
# Shown 1 results
```

Delete all remaining changes for a function by name

```
alice@alice_PC$ lc hist del -s --func calc_things
```

1 entries deleted from history

```
alice@alice_PC$ hist show
# Shown 0 results
```

---

## 3.2. Various information

### 3.2.1. info

#### info

Shows lumina connection information.

Example:

```
alice@alice_PC$ lc info
Hex-Rays Lumina Server v8.0
Lumina time: 2022-08-29 10:13:37, up since 2022-08-21 21:00:05
MAC address: FF:32:67:FF:D3:00
Client name: alice *ADMIN*
Client host: 127.0.0.1
```

### 3.2.2. users

#### users

Shows users.

Example:

```
alice@alice_PC$ lc users
LastActive Adm Login License User name Email
-----
2022-08-29 * bob XX-XXXX-XXXX-XX bob bob@acme.com
2022-08-29 * alice XX-XXXX-XXXX-XX alice alice@acme.com
2022-08-27 damian XX-XXXX-XXXX-XX damian damian@acme.com
```

### 3.2.3. stats

#### stats [OPTION]

Shows the numbers of functions, pushes, history records, IDBs and input files stored in the Lumina server database.

*Options*

**-u, --username USERNAME**  
username(s) to operate on

Examples:

```
alice@alice_PC$ lc stats
Consolidated statistics from lumina_server:
Number of functions: 4
Number of pushes: 5
Number of history records: 6
Number of IDBs: 3
Number of input files: 3
-----
```

*Retrieve the statistics for a list of users*

```
alice@alice_PC$ lc stats -ualice,bob,russ
```

Statistics for alice:

Number of functions: 2

Number of pushes: 3

Number of history records: 4

Number of IDBs: 1

Number of input files: 1

-----  
Statistics for bob:

Number of functions: 1

Number of pushes: 1

Number of history records: 1

Number of IDBs: 1

Number of input files: 1  
-----

## 3.3. Administrative commands

These commands require that the user executing them has admin privileges.

### 3.3.1. Managing users

The following commands allow the administrator to manipulate users known to the Lumina server.

#### 3.3.1.1. `user add`

```
user add USERNAME EMAIL IS_ADMIN LICENSE_ID
```

Adds a user.

The `passwd` command must be used after creating a user; otherwise the new user will not be able to login to the Lumina server and use it.

##### Parameters

<b>USERNAME</b>	The username of the user.
<b>EMAIL</b>	The email address of the user.
<b>IS_ADMIN</b>	Should be 1 if the user is admin, otherwise 0.
<b>LICENSE_ID</b>	The license of the user.

Example:

```
alice@alice_PC$ lc user add damiank damian@acme.com 0 XX-XXXX-XXXX-XX
```

#### 3.3.1.2. `user edit`

```
user edit USERNAME EMAIL IS_ADMIN LICENSE_ID
```

Edits a user definition.

##### Parameters

<b>USERNAME</b>	The username of the user to modify.
<b>EMAIL</b>	The email address of the user.
<b>IS_ADMIN</b>	Should be 1 if the user is admin, otherwise 0.
<b>LICENSE_ID</b>	The license of the user.

Example:

*For the given username, set the email, admin flag and license ID*

```
alice@alice_PC$ lc user edit damiank damiank@acme.com 0 XX-XXXX-XXXX-XX
```

#### 3.3.1.3. `user del`

```
user del USERNAME EMAIL LICENSE_ID
```

Deletes a user.

##### Parameters

<b>USERNAME</b>	The name of the user to delete.
<b>EMAIL</b>	The email address of the user.
<b>LICENSE_ID</b>	The license of the user.

Example:

```
alice@alice_PC$ lc user del damiank damiank@acme.com XX-XXXX-XXXX-XX
```

### 3.3.1.4. passwd

#### passwd PASS USER

Modifies the password for a user.

*Parameters*

<b>PASS</b>	The new password.
<b>USER</b>	The username whose password should be changed Only admins can change other users' passwords.

Example:

```
alice@alice_PC$ lc passwd secret_password john
```

The host name (and port) of the Hex-Rays Vault server need to be present in the `lumina.conf` file.

## 3.3.2. Managing sessions

### 3.3.2.1. session list

#### session list

Lists the current connections to the Lumina server. For each connection, the currently executed query (if any) is shown.

Example:

```
alice@alice_PC$ lc session list
id=1642 peer="127.0.0.1", user="...", license="...", e-mail="...", established="2022-08-16 17:13:21"
current_query="INSERT INTO pushes (fk_idb, fk_user) VALUES (?, ?)" (0msec)
```

### 3.3.2.2. session kill

#### session kill ID

Kills an existing connection to the Lumina server.

*Parameters*

<b>ID</b>	The connection to kill, as shown by the <code>session list</code> command
-----------	---

Example:

```
alice@alice_PC$ lc session list
id=1 peer="127.0.0.1", user="...", license="...", e-mail="...", established="2022-09-20 16:47:07" current_query=""
```

```
(0msec)
alice@alice_PC$ lc session kill 1
Connection killed
alice@alice_PC$ lc session list
No connections.
```

---



## 4. Concepts

### 4.1. What is the Lumina server

The Lumina server is a "functions metadata" repository.

It is a place where IDA users can **push**, and **pull** such metadata, to ease their reverse-engineering work: metadata can be extracted from existing projects, and re-applied effortlessly to new projects, thereby reducing (sometimes dramatically) the amount of time needed to analyze binaries.

#### 4.1.1. Functions metadata

The Lumina server associates "function metadata" to functions, by means of a (md5) *hash* of those functions: whenever it wants to push information to, or pull information from the server, IDA will first have to compute hashes of the functions it wants to retrieve metadata for, and send those hashes to the Lumina server.

Similarly, when IDA **pushes** information to the Lumina server, it will first compute hashes for the corresponding functions, extract the metadata corresponding to those from the `.idb` file, and send those hash+metadata pairs to the server.

#### 4.1.2. Metadata contents

Metadata about functions can include:

- function name
- function address
- function size
- function prototype
- function [repeatable] comments
- instruction-specific [repeatable] comments
- anterior/posterior (i.e., "extra") comments
- user-defined "stack points" in the function's frame
- the function frame description and stack variables
- instructions operands representations

#### 4.1.3. Pushing & overriding metadata

When a user pushes metadata about a function whose md5 hash isn't present in the database, the Lumina server will simply create a new record for it.

However, when a user pushes metadata about a function whose md5 hash (and associated metadata) is already present in the database, the Lumina server will attempt to "score" the quality of the old metadata and the quality of the new metadata. If the score of the new metadata is higher, the new function metadata will override the previous one.

**NOTE** When a user asks IDA to push *all* functions to the Lumina server, IDA will automatically skip some functions: those that still have a "dummy" name (e.g., `sub_XXXX`), or that are below a certain size threshold (i.e., 32 bytes) will be ignored.

#### 4.1.4. Metadata history

The Lumina server retains a history of the metadata associated to functions. Using the `lc` utility, it is possible to dig into that history, and view changes (detailed diffs, too.)

#### 4.1.5. File contents

It's worth pointing out that when pushing metadata to the Lumina server, IDA will not push the binary file itself. Only the following metadata about the file itself will be sent:

- the name of the input file

- the name of the IDB file
- a md5 hash of the input file

The Lumina server cannot therefore be used as a backup/repository for binary files & IDBs

## Appendix A: Commands

### A.1. String patterns

Options that take strings as inputs can be enhanced through wildcards. The following wildcards are available:

**%**

represents zero, one or multiple characters.

**-**

represents one character.

To use wildcards in a string, it must be prefixed with **like**: e.g. `-n like:%main%`.

### A.2. Timerange formats

For timeranges, the following syntaxes are supported:

- `<ts>..<ts>` (from timestamp to (but not including) timestamp)
- `<ts>` (only one timestamp)

Where `<ts>` can be of the form:

- `yyyy-mm-dd HH:MM:SS`: e.g., `2022-09-12 11:38:22`
- `yyyy-mm-dd`: e.g., `2020-03-12`
- `+|-<count><unit>`: this is a "now-relative" timestamp, where `<unit>` must be one of `w, d, H, M, S` for weeks, days, hours, minutes or seconds respectively. E.g., `-4d, +5w, -8H, +1H, ...`

#### NOTE

when using the `<ts>` syntax (i.e., only 1 timestamp is provided, not an actual range), the final range will be either "from now to `<ts>`", or "from `<ts>` to now", depending on whether `<ts>` is before, or after, the present time.