

LWL Reference Manual

1.0

Generated by Doxygen 1.3.5

Wed Jan 5 16:54:39 2005

Contents

1	LWL Module Index	1
1.1	LWL Modules	1
2	LWL File Index	3
2.1	LWL File List	3
3	LWL Module Documentation	5
3.1	Log Writer Library	5
4	LWL File Documentation	13
4.1	lwl.h File Reference	13

Chapter 1

LWL Module Index

1.1 LWL Modules

Here is a list of all modules:

Log Writer Library	5
------------------------------	---

Chapter 2

LWL File Index

2.1 LWL File List

Here is a list of all files with brief descriptions:

lwl.h	13
------------------------	-----------

Chapter 3

LWL Module Documentation

3.1 Log Writer Library

Defines

- `#define lwl_get_log_file(h) lwl_get_default_log_file(h)`
Alias for `lwl_get_default_log_file()` (p. 10).

Typedefs

- `typedef _lwl_handle * lwlh_t`
LWL handle type.

Enumerations

- `enum lwl_tag_t {`
 `LWL_TAG_DONE = 0, LWL_TAG_LEVEL, LWL_TAG_`
 `PREFIX, LWL_TAG_OPTIONS,`
 `LWL_TAG_FILE, LWL_TAG_FILE_EMERG, LWL_TAG_`
 `FILE_ALERT, LWL_TAG_FILE_CRIT,`
 `LWL_TAG_FILE_ERR, LWL_TAG_FILE_WARNING,`
 `LWL_TAG_FILE_NOTICE, LWL_TAG_FILE_INFO,`
 `LWL_TAG_FILE_DEBUG }`
LWL Tags.
- `enum lwl_option_t {`
 `LWL_OPT_NONE = 0, LWL_OPT_PERROR = 1, LWL_`
 `OPT_PID = 2, LWL_OPT_DATE = 4,`

```
LWL_OPT_TIME = 8, LWL_OPT_IP = 16, LWL_OPT_ADR
= 32, LWL_OPT_PRIORITY = 64,
LWL_OPT_PREFIX = 128, LWL_OPT_NO_FLUSH = 256,
LWL_OPT_USE_LOCALE = 512 }
```

LWL options.

- enum `lwl_priority_t` {
`LWL_PRI_EMERG = 0, LWL_PRI_ALERT = 1, LWL_PRI_CRIT = 2, LWL_PRI_ERR = 3,`
`LWL_PRI_WARNING = 4, LWL_PRI_NOTICE = 5, LWL_PRI_INFO = 6, LWL_PRI_DEBUG = 7` }

LWL priority.

Functions

- `lwlh_t lwl_alloc` (void)
Create a new LWL handle.
- `void lwl_free` (`lwlh_t HANDLE`)
Destroy a LWL handle.
- `void lwl_write_log` (`lwlh_t HANDLE, lwl_priority_t PRI, char *FORMAT,...`)
Log a message to the files associated to a LWL handle.
- `int lwl_set_attributes` (`lwlh_t HANDLE, lwl_tag_t TAGS,...`)
Modify the attributes of a LWL handle.
- `lwl_priority_t lwl_get_log_level` (`const lwlh_t HANDLE`)
Get the actual log level of a LWL handle.
- `FILE * lwl_get_default_log_file` (`const lwlh_t HANDLE`)
Get the actual default file where a LWL handle is logging into.
- `const char * lwl_get_hostname` (`lwlh_t HANDLE`)
Get the hostname of a LWL handle.

3.1.1 Define Documentation

3.1.1.1 `#define lwl_get_log_file(h) lwl_get_default_log_file(h)`

Alias for `lwl_get_default_log_file()`(p. 10).

See also:

`lwl_get_default_log_file()`(p.10)

Definition at line 309 of file `lwl.h`.

3.1.2 Typedef Documentation

3.1.2.1 typedef struct `_lwl_handle*` `lwlh_t`

LWL handle type.

This type is voluntary opaque. Variables of this kind could't be directly used, but by the functions of this module.

Definition at line 181 of file `lwl.h`.

3.1.3 Enumeration Type Documentation

3.1.3.1 enum `lwl_option_t`

LWL options.

This is the list of available LWL options for use with `lwl_set_attributes()`(p.11). The options could be XOR-ed to use more than one option at the same time.

Example: `LWL_OPT_ADR | LWL_OPT_PID | LWL_OPT_DATE | LWL_OPT_TIME` means you want to use all these options for each logged message.

See also:

`lwl_set_attributes()`(p.11)

Enumeration values:

`LWL_OPT_NONE` No option (default)

`LWL_OPT_PERROR` The message is also logged to `stderr`

`LWL_OPT_PID` Write the pid of the process

`LWL_OPT_DATE` Write the date

`LWL_OPT_TIME` Write the time

`LWL_OPT_IP` Write host's IP address

`LWL_OPT_ADR` Write host's FQDN

`LWL_OPT_PRIORITY` Write the `LWL_PRIORITY` of the message

`LWL_OPT_PREFIX` Write prefix specified by `LWL_TAG_PREFIX`

`LWL_OPT_NO_FLUSH` Don't flush the file after log writing

`LWL_OPT_USE_LOCALE` Use locale environment settings to display date and time

Definition at line 103 of file `lwl.h`.

3.1.3.2 enum `lwl_priority_t`

LWL priority.

This is the list of available LWL priorities for use with `lwl_write_log()`(p. 11)

See also:

`lwl_write_log()`(p. 11)

Enumeration values:

LWL_PRI_EMERG EMERGENCY level

LWL_PRI_ALERT ALERT level

LWL_PRI_CRIT CRITICAL level

LWL_PRI_ERR ERROR level

LWL_PRI_WARNING WARNING level

LWL_PRI_NOTICE NOTICE level

LWL_PRI_INFO INFORMATION level

LWL_PRI_DEBUG DEBUG level

Definition at line 147 of file `lwl.h`.

3.1.3.3 enum `lwl_tag_t`

LWL Tags.

This is the list of available `lwl_tag_t` tags for use with `lwl_set_attributes()`(p. 11).

See also:

`lwl_set_attributes()`(p. 11)

Enumeration values:

LWL_TAG_DONE Special tag to end the tag list

LWL_TAG_LEVEL Tag to modify the level of a LWL handle

LWL_TAG_PREFIX Tag to set the prefix of the logged messages

LWL_TAG_OPTIONS Tag to set options to use in a LWL handle

LWL_TAG_FILE Tag to set the default file used to write messages in

LWL_TAG_FILE_EMERG Tag to set the file for EMERG level

LWL_TAG_FILE_ALERT Tag to set the file for ALERT level

LWL_TAG_FILE_CRIT Tag to set the file for CRIT level

LWL_TAG_FILE_ERR Tag to set the file for ERR level

LWL_TAG_FILE_WARNING Tag to set the file for WARNING level

LWL_TAG_FILE_NOTICE Tag to set the file for NOTICE level
LWL_TAG_FILE_INFO Tag to set the file for INFO level
LWL_TAG_FILE_DEBUG Tag to set the file for DEBUG level

Definition at line 49 of file lwl.h.

3.1.4 Function Documentation

3.1.4.1 `lwlh_t lwl_alloc (void)`

Create a new LWL handle.

Allocate a new LWL handle data structure. The initial log file used is stdout and the initial level is `LWL_PRI_DEBUG`. To change them and other attributes, uses `lwl_set_attributes()`(p. 11) function.

Default values are:

- `LWL_PRI_DEBUG` for `LWL_TAG_LEVEL`
- `LWL_OPT_NONE` for `LWL_TAG_OPTIONS`
- `NULL` for `LWL_TAG_PREFIX`
- `stdout` for `LWL_TAG_FILE`
- `NULL` for all `LWL_TAG_FILE_XXX`

Precondition:

nothing

Returns:

the newly allocated LWL handle in case of success.
`NULL` in case of insufficient memory.

See also:

`lwl_free()`(p.9)

3.1.4.2 `void lwl_free (lwlh_t HANDLE)`

Destroy a LWL handle.

Deallocate and destroy the lwl HANDLE.

Precondition:

HANDLE must be a valid lwlh_t.

Parameters:

HANDLE The lwl handle to deallocate.

See also:

`lwl_alloc()`(p.9)

3.1.4.3 FILE* lwl_get_default_log_file (const lwlh_t *HANDLE*)

Get the actual default file where a LWL handle is logging into.

Precondition:

HANDLE must be a valid lwlh_t.

Parameters:

HANDLE The LWL handle to use.

Returns:

the default file where *HANDLE* is actually logging.

3.1.4.4 const char* lwl_get_hostname (lwlh_t *HANDLE*)

Get the hostname of a LWL handle.

Precondition:

HANDLE must be a valid lwlh_t.

Parameters:

HANDLE The LWL handle to use.

Returns:

the hostname if *HANDLE*'s options includes LWL_OPT_IP and/or LWL_OPT_ADR.
NULL if *HANDLE*'s options does not include LWL_OPT_IP and/or LWL_OPT_ADR.

3.1.4.5 lwl_priority_t lwl_get_log_level (const lwlh_t *HANDLE*)

Get the actual log level of a LWL handle.

Precondition:

HANDLE must be a valid lwlh_t.

Returns:

the actual log level of *HANDLE*.

Parameters:

HANDLE The LWL handle to use.

3.1.4.6 `int lwl_set_attributes (lwlh_t HANDLE, lwl_tag_t TAGS, ...)`

Modify the attributes of a LWL handle.

To use this function, you must use LWL_TAGs to specify each tag you want to set. The list of tags is fully determined by the `lwl_tag_t` type. Each tag must be followed by its new value, and the tag list must be terminated by LWL_TAG_DONE. If the tag list is not terminated by LWL_TAG_DONE the result of this function is not determined.

Example:

```
lwl_set_attributes (my_handle, LWL_TAG_PREFIX, "my example app",
LWL_TAG_FILE, stderr, LWL_TAG_OPTIONS, LWL_OPT_PRIORITY
| LWL_OPT_DATE, LWL_TAG_DONE);
```

Precondition:

HANDLE must be a valid `lwlh_t`.

Parameters:

HANDLE The LWL handle to modify.

TAGS,... A coma separated list of (LWL_TAG_XXX, value) couples, terminated by the special LWL_TAG_DONE `lwl_tag_t`.

Returns:

0 in case of success.

-1 in case of failure.

See also:

`lwl_tag_t`(p.8)

3.1.4.7 `void lwl_write_log (lwlh_t HANDLE, lwl_priority_t PRI, char * FORMAT, ...)`

Log a message to the files associated to a LWL handle.

Write the given formatted message of priority *PRI* to the *HANDLE*'s files. To change the *HANDLE*'s files, or log level, you can use `lwl_set_attributes`(p.11) function. If no files were specified by LWL_TAG_FILE_XXX tags, then all logs are written to the file specified by LWL_TAG_FILE (or to the default one if no LWL_TAG_FILE was specified). If a file was specified for a given level, by use of tags like LWL_TAG_FILE_XXX, then the specified file is used instead of the default one. The message *FORMAT* must be used like `printf()`. The message will be logged only if *PRI* is <= *HANDLE*'s log level (wich you can consult with `lwl_get_log_level`(p.10)).

Example:

```
lwl_write_log (my_handle, LWL_PRI_DEBUG, "%s:%d-%s ERROR", __-
FILE__, __LINE__, __FUNCTION__);
```

Precondition:

`HANDLE` must be a valid `lwlh_t`.

Parameters:

HANDLE The LWL handle to use.

PRI The priority of the message to log.

FORMAT The message to log.

... Variables to include into the message.

See also:

`lwl_get_log_level`(p. 10)

`lwl_set_attributes`(p. 11)

Chapter 4

LWL File Documentation

4.1 lwl.h File Reference

Defines

- `#define lwl_get_log_file(h) lwl_get_default_log_file(h)`
Alias for `lwl_get_default_log_file()` (p.10).

Typedefs

- `typedef _lwl_handle * lwlh_t`
LWL handle type.

Enumerations

- `enum lwl_tag_t {`
 `LWL_TAG_DONE = 0, LWL_TAG_LEVEL, LWL_TAG_`
 `PREFIX, LWL_TAG_OPTIONS,`
 `LWL_TAG_FILE, LWL_TAG_FILE_EMERG, LWL_TAG_`
 `FILE_ALERT, LWL_TAG_FILE_CRIT,`
 `LWL_TAG_FILE_ERR, LWL_TAG_FILE_WARNING,`
 `LWL_TAG_FILE_NOTICE, LWL_TAG_FILE_INFO,`
 `LWL_TAG_FILE_DEBUG }`
LWL Tags.
- `enum lwl_option_t {`
 `LWL_OPT_NONE = 0, LWL_OPT_PERROR = 1, LWL_`
 `OPT_PID = 2, LWL_OPT_DATE = 4,`

```
LWL_OPT_TIME = 8, LWL_OPT_IP = 16, LWL_OPT_ADR
= 32, LWL_OPT_PRIORITY = 64,
LWL_OPT_PREFIX = 128, LWL_OPT_NO_FLUSH = 256,
LWL_OPT_USE_LOCALE = 512 }
```

LWL options.

- enum `lwl_priority_t` {


```
LWL_PRI_EMERG = 0, LWL_PRI_ALERT = 1, LWL_
PRI_CRIT = 2, LWL_PRI_ERR = 3,
LWL_PRI_WARNING = 4, LWL_PRI_NOTICE = 5, LWL_
PRI_INFO = 6, LWL_PRI_DEBUG = 7 }
```

LWL priority.

Functions

- `lwlh_t lwl_alloc` (void)

Create a new LWL handle.
- `void lwl_free` (`lwlh_t HANDLE`)

Destroy a LWL handle.
- `void lwl_write_log` (`lwlh_t HANDLE`, `lwl_priority_t PRI`, `char *FORMAT`,...)

Log a message to the files associated to a LWL handle.
- `int lwl_set_attributes` (`lwlh_t HANDLE`, `lwl_tag_t TAGS`,...)

Modify the attributes of a LWL handle.
- `lwl_priority_t lwl_get_log_level` (const `lwlh_t HANDLE`)

Get the actual log level of a LWL handle.
- `FILE * lwl_get_default_log_file` (const `lwlh_t HANDLE`)

Get the actual default file where a LWL handle is logging into.
- `const char * lwl_get_hostname` (`lwlh_t HANDLE`)

Get the hostname of a LWL handle.

Index

Log Writer Library, 5

LWL

- lwl_alloc, 9
- lwl_free, 9
- lwl_get_default_log_file, 9
- lwl_get_hostname, 10
- lwl_get_log_file, 6
- lwl_get_log_level, 10
- LWL_OPT_ADR, 7
- LWL_OPT_DATE, 7
- LWL_OPT_IP, 7
- LWL_OPT_NO_FLUSH, 7
- LWL_OPT_NONE, 7
- LWL_OPT_PERROR, 7
- LWL_OPT_PID, 7
- LWL_OPT_PREFIX, 7
- LWL_OPT_PRIORITY, 7
- LWL_OPT_TIME, 7
- LWL_OPT_USE_LOCALE, 7
- lwl_option_t, 7
- LWL_PRI_ALERT, 8
- LWL_PRI_CRIT, 8
- LWL_PRI_DEBUG, 8
- LWL_PRI_EMERG, 8
- LWL_PRI_ERR, 8
- LWL_PRI_INFO, 8
- LWL_PRI_NOTICE, 8
- LWL_PRI_WARNING, 8
- lwl_priority_t, 7
- lwl_set_attributes, 10
- LWL_TAG_DONE, 8
- LWL_TAG_FILE, 8
- LWL_TAG_FILE_ALERT, 8
- LWL_TAG_FILE_CRIT, 8
- LWL_TAG_FILE_DEBUG, 9
- LWL_TAG_FILE_EMERG, 8
- LWL_TAG_FILE_ERR, 8
- LWL_TAG_FILE_INFO, 9

LWL_TAG_FILE_NOTICE, 8

LWL_TAG_FILE_WARNING, 8

LWL_TAG_LEVEL, 8

LWL_TAG_OPTIONS, 8

LWL_TAG_PREFIX, 8

lwl_tag_t, 8

lwl_write_log, 11

lwlh_t, 7

lwl.h, 13

lwl_alloc
LWL, 9

lwl_free
LWL, 9

lwl_get_default_log_file
LWL, 9

lwl_get_hostname
LWL, 10

lwl_get_log_file
LWL, 6

lwl_get_log_level
LWL, 10

LWL_OPT_ADR
LWL, 7

LWL_OPT_DATE
LWL, 7

LWL_OPT_IP
LWL, 7

LWL_OPT_NO_FLUSH
LWL, 7

LWL_OPT_NONE
LWL, 7

LWL_OPT_PERROR
LWL, 7

LWL_OPT_PID
LWL, 7

LWL_OPT_PREFIX
LWL, 7

LWL_OPT_PRIORITY
LWL, 7

LWL_OPT_TIME	LWL_TAG_PREFIX
LWL, 7	LWL, 8
LWL_OPT_USE_LOCALE	lwl_tag_t
LWL, 7	LWL, 8
lwl_option_t	lwl_write_log
LWL, 7	LWL, 11
LWL_PRI_ALERT	lwlh_t
LWL, 8	LWL, 7
LWL_PRI_CRIT	
LWL, 8	
LWL_PRI_DEBUG	
LWL, 8	
LWL_PRI_EMERG	
LWL, 8	
LWL_PRI_ERR	
LWL, 8	
LWL_PRI_INFO	
LWL, 8	
LWL_PRI_NOTICE	
LWL, 8	
LWL_PRI_WARNING	
LWL, 8	
lwl_priority_t	
LWL, 7	
lwl_set_attributes	
LWL, 10	
LWL_TAG_DONE	
LWL, 8	
LWL_TAG_FILE	
LWL, 8	
LWL_TAG_FILE_ALERT	
LWL, 8	
LWL_TAG_FILE_CRIT	
LWL, 8	
LWL_TAG_FILE_DEBUG	
LWL, 9	
LWL_TAG_FILE_EMERG	
LWL, 8	
LWL_TAG_FILE_ERR	
LWL, 8	
LWL_TAG_FILE_INFO	
LWL, 9	
LWL_TAG_FILE_NOTICE	
LWL, 8	
LWL_TAG_FILE_WARNING	
LWL, 8	
LWL_TAG_LEVEL	
LWL, 8	
LWL_TAG_OPTIONS	
LWL, 8	