



# LuaTask addon library 1.6

## Reference guide

---

### Table of Contents

- [Task library initialization](#)
  - [create - Task creation](#)
  - [id - Task id](#)
  - [register - Task name registration](#)
  - [find - Task finding](#)
  - [unregister - Task name un-registration](#)
  - [isrunning - Task status checking](#)
  - [cancel - Task termination](#)
  - [list - Task list retrieval](#)
  - [post - Message posting](#)
  - [receive - Message receiving](#)
  - [getqhandle - Task queue handle retrieval](#)
  - [sleep - Task execution suspension](#)
- 

**Name:**           **Task library initialization**

**Synopsis:**  
**(function returned by loadlib)( [libraries-to-load])**

**Description:**  
The initialization function only exists in the dynamic version of LuaTask.  
It must be called for the "main" task.

**Arguments:**  
libraries-to-load (table) : libraries to be loaded automatically. (members in table with value "true" will be loaded)

base	Base lib (always loaded)
table	Table manipulation
io	Input/output
string	String manipulation
math	Mathematical functions
debug	Debugging
loadlib	Dynamic loading

**Notes:**

If libraries-to-load is nil, all libraries will be loaded.

---

**Name:**

**create - Task creation**

**Synopsis:**

`rc = task.create( s [, args]`

**Description:**

The **task.create** function creates a new task and tries to execute the script "s". Where "s" is a script file name or the "=" character followed for a text or binary chunk.

**Arguments:**

s (string)	Lua chunk/script file name.
args (table)	The new task argument list.

**Returns:**

rc (number):

> 0	New task id.
-1	Can't expand task list.
-2	Can't strdup file name.
-3	Can't create message queue.
-4	Can't create os thread.
-11	The library seems corrupt.

**Notes:**

The args table must be number indexed, and only member of string and number type are supported.

---

**Name:****id - Task id****Synopsis:**

```
rc = task.id( )
```

**Description:**

The **task.id** function returns the id for the current task.

**Returns:**

rc (number):

> 0	Current task id.
-1	The library seems corrupt.

---

**Name:****register - Task name registration****Synopsis:**

```
rc = task.register( name)
```

**Description:**

The **task.register** function sets a user defined name for the current task.

**Arguments:**

name (string)	String to "name" the current task.
---------------	------------------------------------

**Returns:**

rc (number):

> 0	Ok.
-1	The library seems corrupt.

---

**Name:** **find - Task finding**

**Synopsis:**

rc = **task.find**( name)

**Description:**

The **task.find** function looks for the id of task registered with the given name.

**Arguments:**

name (string)	Task register name to find.
---------------	-----------------------------

**Returns:**

rc (number):

> 0	Task id.
-1	Task not found.

---

**Name:** **unregister - Task name un-registration**

**Synopsis:**

rc = **task.unregister**( )

**Description:**

The **task.unregister** function clears the name registered for the current task.

**Returns:**

rc (number):

> 0	Task id.
-1	The library seems corrupt.

---

**Name:** **isrunning - Task status checking**

**Synopsis:**

```
running = task.isrunning( id)
```

**Description:**

The **task.isrunning** function returns if the task identified by id is running or not.

**Arguments:**

id (number)	Task to check.
-------------	----------------

**Returns:**

running (boolean) : Running or not

---

**Name:** **cancel - Task termination**

**Synopsis:**

```
rc = task.cancel( id)
```

**Description:**

The **task.cancel** function interrupts execution of the task identified by id.

**Arguments:**

id (number)	Task to terminate.
-------------	--------------------

**Returns:**

rc (number): pthread\_cancel return code.

**Notes for Win32:**

Only works with the Pthreads-Win32 version.  
Thread cancellation is guaranteed only if the QueueUserAPCEx service is running.

---

**Name:** **list - Task list retrieval**

**Synopsis:**

```
list = task.list()
```

**Description:**

The **task.list** function gets a representation of the global tasks list.

**Returns:**

list (table): Task list indexed by task id.

Members:	
id	Registered name ( or nil).
script	Script file name.
msgcount	Current message in queue count.

---

**Name:** **post - Message posting**

**Synopsis:**

```
rc = task.post( id, msg, flags)
```

**Description:**

The **task.post** function appends msg to the message queue of the task identified by id.

**Arguments:**

id (number)	Task to post.
msg (string)	Data to post.
flags (number)	General purpose 32 bit number.

**Returns:**

rc (number):

0	Message posted ok.
-1	Task to post not running.
-2	Can't malloc message entry.

---

**Name:**            **receive - Message receiving**

**Synopsis:**  
msg, flags, rc = **task.receive**( [timeout])

**Description:**  
The **task.receive** function gets the first entry from the message queue of the current task.  
If timeout exists and it is not equal to -1, it specifies the maximum interval to wait for message arrival if there are none in the queue .

**Arguments:**  

timeout (number)	Receive timeout in milliseconds, -1 or nil waits forever.
------------------	---

**Returns:**  
msg (string) : Data received  
flags (number): General purpose 32 bit number.  
rc (number):

0	Received ok.
-1	The library seems corrupt.
-2	Timed out.

---

**Name:**            **getqhandle - Task queue handle retrieval**

**Synopsis:**  
rc = **task.getqhandle**( )

**Description:**

The **task.getqhandle** function returns the message queue handle for the current task.

**Returns:**

rc (number):

> 0	Current task internal message queue handle.
0	The library seems corrupt.

**Notes:**

The handle returned by this function is dependent of the platform.

The only purpose of this function is to make available a handle to use with things like select or WaitFor\* functions.

---

**Name:**

**sleep – Task execution suspension**

**Synopsis:**

**task.sleep**( ms)

**Description:**

The **task.sleep** function suspends execution of the current task.

**Arguments:**

ms (number)	Time to suspend execution.
-------------	----------------------------

---