

Dynamic Load Balancing Library - Timer User API

0.0.1

Generated by Doxygen 1.5.6

Tue Aug 19 10:43:53 2008

Contents

1	Todo List	1
2	File Index	1
2.1	File List	1
3	File Documentation	1
3.1	src/timer.h File Reference	1
3.1.1	Detailed Description	1
3.1.2	Typedef Documentation	2
3.1.3	Function Documentation	2

1 Todo List

File [timer.h](#) Add some error handling

2 File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

src/timer.h	1
-----------------------------	----------

3 File Documentation

3.1 src/timer.h File Reference

3.1.1 Detailed Description

This is the header file that needs to be included if you want the timer functionality of the dynamic load balancing library.

It includes definition of the `DLB_Timer_Handle` and `DLB_Time_Span` types and the functions that can manipulate timers.

Todo

Add some error handling

Typedefs

- typedef int [DLB_Timer_Handle](#)
- typedef double [DLB_Time_Span](#)

Functions

- [DLB_Timer_Handle](#) [DLB_Timer_Create](#) ()
- void [DLB_Timer_Start](#) ([DLB_Timer_Handle](#) handle)
- [DLB_Time_Span](#) [DLB_Timer_Get_Elapsed_Time](#) ([DLB_Timer_Handle](#) handle)
- void [DLB_Timer_Stop](#) ([DLB_Timer_Handle](#) handle)
- void [DLB_Timer_Reset](#) ([DLB_Timer_Handle](#) handle)
- void [DLB_Timer_Destroy](#) ([DLB_Timer_Handle](#) handle)
- void [DLB_Timer_Destroy_All](#) ()

3.1.2 Typedef Documentation

3.1.2.1 typedef double [DLB_Time_Span](#)

A time span. Used to return the elapsed time from a timer, see [DLB_Timer_Get_Elapsed_Time\(\)](#)

3.1.2.2 typedef int [DLB_Timer_Handle](#)

The handle to a timer. Use this to refer to a timer when you want data from it. It is returned when you create a timer, see [DLB_Timer_Create\(\)](#).

3.1.3 Function Documentation

3.1.3.1 [DLB_Timer_Handle](#) [DLB_Timer_Create](#) ()

Create a timer. Returns a handle to the timer. It is necessary to destroy the timer (with [DLB_Timer_Destroy\(\)](#)) when it is no longer required thus freeing the memory used by the timer.

Returns:

A handle to the new timer.

3.1.3.2 void [DLB_Timer_Destroy](#) ([DLB_Timer_Handle](#) *handle*)

Destroy a timer. Frees the memory the timer used.

Parameters:

handle The handle of the timer to be destroyed.

3.1.3.3 void [DLB_Timer_Destroy_All](#) ()

Destroy all timers. Frees the memory the timers used.

3.1.3.4 [DLB_Time_Span](#) [DLB_Timer_Get_Elapsed_Time](#) ([DLB_Timer_Handle](#) *handle*)

Get the elapsed time in seconds for a timer.

Parameters:

handle The timer we are interested in.

Returns:

The elapsed time for this timer.

3.1.3.5 void DLB_Timer_Reset (DLB_Timer_Handle *handle*)

Reset a timer so elapsed time is zero. Has no effect on the running status of the timer.

Parameters:

handle The handle of the timer to be reset.

3.1.3.6 void DLB_Timer_Start (DLB_Timer_Handle *handle*)

Start a timer. The elapsed time for the timer will not be reset.

Parameters:

handle The timer to start.

3.1.3.7 void DLB_Timer_Stop (DLB_Timer_Handle *handle*)

Stop a timer. Stops the timer.

Parameters:

handle The timer to stop.

Index

DLB_Time_Span

timer.h, [2](#)

DLB_Timer_Create

timer.h, [2](#)

DLB_Timer_Destroy

timer.h, [2](#)

DLB_Timer_Destroy_All

timer.h, [2](#)

DLB_Timer_Get_Elapsed_Time

timer.h, [2](#)

DLB_Timer_Handle

timer.h, [2](#)

DLB_Timer_Reset

timer.h, [2](#)

DLB_Timer_Start

timer.h, [2](#)

DLB_Timer_Stop

timer.h, [3](#)

src/timer.h, [1](#)

timer.h

DLB_Time_Span, [2](#)

DLB_Timer_Create, [2](#)

DLB_Timer_Destroy, [2](#)

DLB_Timer_Destroy_All, [2](#)

DLB_Timer_Get_Elapsed_Time, [2](#)

DLB_Timer_Handle, [2](#)

DLB_Timer_Reset, [2](#)

DLB_Timer_Start, [2](#)

DLB_Timer_Stop, [3](#)