

pgtime

Generated by Doxygen 1.8.1.2

Mon Sep 23 2013 21:36:41

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	main.c File Reference	3
2.1.1	Detailed Description	3
2.1.2	Function Documentation	4
2.1.2.1	main	4
2.2	pgtime.c File Reference	4
2.2.1	Detailed Description	5
2.2.2	Function Documentation	5
2.2.2.1	check_utc_timestamp	5
2.2.2.2	get_day_diff	6
2.2.2.3	get_hour_diff	6
2.2.2.4	get_sec_diff	6
2.2.2.5	get_utc_timestamp	7
2.2.2.6	get_utc_timestamp_sec_diff	7
2.2.2.7	is_leap_year	7
2.2.2.8	tm_compare	7
2.2.2.9	tm_decrement_day	8
2.2.2.10	tm_decrement_hour	8
2.2.2.11	tm_decrement_minute	8
2.2.2.12	tm_decrement_second	8
2.2.2.13	tm_increment_day	9
2.2.2.14	tm_increment_hour	9
2.2.2.15	tm_increment_minute	9
2.2.2.16	tm_increment_second	10
2.2.2.17	tm_intraday_secs_diff	10
2.2.2.18	validate_date	10
2.3	pgtime.h File Reference	11
2.3.1	Detailed Description	12

2.3.2	Function Documentation	12
2.3.2.1	check_utc_timestamp	12
2.3.2.2	get_day_diff	13
2.3.2.3	get_hour_diff	13
2.3.2.4	get_sec_diff	13
2.3.2.5	get_utc_timestamp	14
2.3.2.6	get_utc_timestamp_sec_diff	14
2.3.2.7	is_leap_year	14
2.3.2.8	tm_compare	14
2.3.2.9	tm_decrement_day	15
2.3.2.10	tm_decrement_hour	15
2.3.2.11	tm_decrement_minute	15
2.3.2.12	tm_decrement_second	15
2.3.2.13	tm_increment_day	16
2.3.2.14	tm_increment_hour	16
2.3.2.15	tm_increment_minute	16
2.3.2.16	tm_increment_second	17
2.3.2.17	tm_intraday_secs_diff	17
2.3.2.18	validate_date	17

Chapter 1

File Index

1.1 File List

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

main.c	Main function for pgtime	3
pgtime.c	Implementation of miscellaneous time functions	4
pgtime.h	Interface to miscellaneous time functions	11

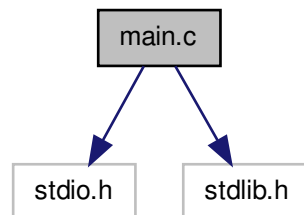
Chapter 2

File Documentation

2.1 main.c File Reference

Main function for pgtime.

```
#include <stdio.h>
#include <stdlib.h>
Include dependency graph for main.c:
```



Functions

- int `main` (void)
Main function.

2.1.1 Detailed Description

Main function for pgtime. Main function for pgtime.

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

2.1.2 Function Documentation

2.1.2.1 int main (void)

Main function.

Main function.

Returns

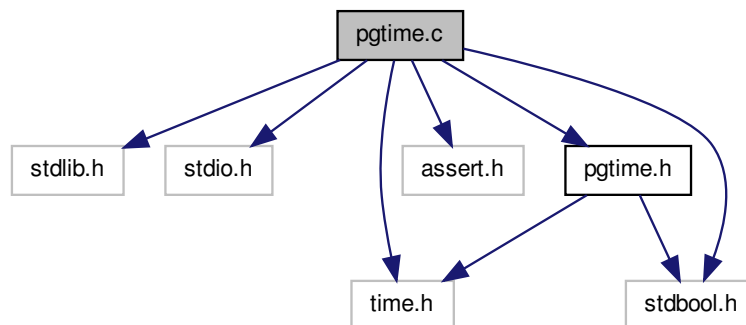
Exit status.

2.2 pgtime.c File Reference

Implementation of miscellaneous time functions.

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <assert.h>
#include <stdbool.h>
#include "pgtime.h"
```

Include dependency graph for pgtime.c:



Functions

- bool [validate_date](#) (const struct tm *check_tm)
Checks whether a supplied date is valid.
- bool [check_utc_timestamp](#) (const time_t check_time, int *secs_diff, const struct tm *check_tm)
Checks if a UTC timestamp is accurate.
- time_t [get_day_diff](#) (void)
Returns a time_t interval representing one day.
- time_t [get_hour_diff](#) (void)
Returns a time_t interval representing one hour.
- time_t [get_sec_diff](#) (void)
Returns a time_t interval representing one second.
- int [tm_compare](#) (const struct tm *first, const struct tm *second)
Compares two struct tm structs.

- int `tm_intraday_secs_diff` (const struct tm *first, const struct tm *second)
Returns the difference between two struct tm structs.
- bool `is_leap_year` (const int year)
Checks if the supplied year is a leap year.
- struct tm * `tm_increment_day` (struct tm *changing_tm, const int quantity)
Adds one or more days to a struct tm, incrementing the month and/or the year as necessary.
- struct tm * `tm_increment_hour` (struct tm *changing_tm, const int quantity)
Adds one or more hours to a struct tm, incrementing the day, month and/or the year as necessary.
- struct tm * `tm_increment_minute` (struct tm *changing_tm, const int quantity)
Adds one or more minutes to a struct tm, incrementing the hour, day, month and/or the year as necessary.
- struct tm * `tm_increment_second` (struct tm *changing_tm, const int quantity)
Adds one or more seconds to a struct tm, incrementing the minute, hour, day, month and/or the year as necessary.
- struct tm * `tm_decrement_day` (struct tm *changing_tm, const int quantity)
Deducts one or more days from a struct tm, decrementing the month and/or the year as necessary.
- struct tm * `tm_decrement_hour` (struct tm *changing_tm, const int quantity)
Deducts one or more hours from a struct tm, decrementing the day, month and/or the year as necessary.
- struct tm * `tm_decrement_minute` (struct tm *changing_tm, const int quantity)
Deducts one or more minutes from a struct tm, decrementing the hour, day, month and/or the year as necessary.
- struct tm * `tm_decrement_second` (struct tm *changing_tm, const int quantity)
Deducts one or more seconds from a struct tm, decrementing the minute, hour, day, month and/or the year as necessary.
- time_t `get_utc_timestamp` (const struct tm *utc_tm)
Gets a time_t timestamp for a requested UTC time.
- int `get_utc_timestamp_sec_diff` (const time_t check_time, const struct tm *utc_tm)
Checks a time_t timestamp against a UTC time, and returns the difference in seconds.

2.2.1 Detailed Description

Implementation of miscellaneous time functions.

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

2.2.2 Function Documentation

2.2.2.1 bool check_utc_timestamp (const time_t check_time, int * secs_diff, const struct tm * check_tm)

Checks if a UTC timestamp is accurate.

Checks if a UTC timestamp is accurate. A time_t timestamp is computed from the supplied datetime, and compared to the supplied time_t timestamp. The difference between the two, in seconds, is stored in the supplied secs_diff argument. This function is needed because the methodology used to calculate a timestamp by this library can sometimes be inaccurate when leap seconds or other unpredictable calendar changes occur. We therefore need a method to check if the returned timestamp is accurate. Other functions provided in this library call this function, so the user should not normally need to call it.

Parameters

<i>check_time</i>	The time_t timestamp to check
<i>secs_diff</i>	Modified to contain the difference, in seconds
<i>check_tm</i>	A pointer to a struct tm containing the date to check.

Returns

true if the supplied timestamp is accurate, false otherwise

2.2.2.2 time_t get_day_diff (void)

Returns a time_t interval representing one day.

Returns a time_t interval representing one day. The C standard does not define the units in which a time_t value is measured. On POSIX-compliant systems it is measured in seconds, but we cannot assume this for full portability.

Returns

A time_t interval representing one day.

Exceptions

<i>bad_time</i>	if the current time cannot be obtained.
-----------------	---

2.2.2.3 time_t get_hour_diff (void)

Returns a time_t interval representing one hour.

Returns a time_t interval representing one hour. The C standard does not define the units in which a time_t value is measured. On POSIX-compliant systems it is measured in seconds, but we cannot assume this for full portability.

Returns

A time_t interval representing one hour.

Exceptions

<i>bad_time</i>	if the current time cannot be obtained.
-----------------	---

2.2.2.4 time_t get_sec_diff (void)

Returns a time_t interval representing one second.

Returns a time_t interval representing one second. The C standard does not define the units in which a time_t value is measured. On POSIX-compliant systems it is measured in seconds, but we cannot assume this for full portability.

Returns

A time_t interval representing one second.

Exceptions

<i>bad_time</i>	if the current time cannot be obtained.
-----------------	---

2.2.2.5 time_t get_utc_timestamp (const struct tm * *utc_tm*)

Gets a time_t timestamp for a requested UTC time.

Parameters

<i>utc_tm</i>	A pointer to a struct tm containing the UTC time.
---------------	---

Returns

A time_t timestamp for the requested UTC time.

2.2.2.6 int get_utc_timestamp_sec_diff (const time_t *check_time*, const struct tm * *utc_tm*)

Checks a time_t timestamp against a UTC time, and returns the difference in seconds.

This function only returns a good value if the timestamp is less than 24 hours away from the desired time, so the caller is responsible for making sure that it is. This function may also return a bad value if a leap second or other unpredictable calendar change falls between the desired UTC time and the provided time stamp. The result should therefore always be checked with [check_utc_timestamp\(\)](#), or by calling this function again.

Parameters

<i>check_time</i>	The time_t timestamp to check
<i>utc_tm</i>	A pointer to a struct tm against which to check.

Returns

The difference, if any, represented in seconds.

2.2.2.7 bool is_leap_year (const int *year*)

Checks if the supplied year is a leap year.

Checks if the supplied year is a leap year.

Parameters

<i>year</i>	A year
-------------	--------

Returns

true if *year* is a leap year, false otherwise.

2.2.2.8 int tm_compare (const struct tm * *first*, const struct tm * *second*)

Compares two struct tm structs.

Compares two struct tm structs. Only the year, month, day, hour, minute and second are compared. Any timezone or DST information is ignored.

Parameters

<i>first</i>	The first struct tm struct.
<i>second</i>	The second struct tm struct.

Returns

-1 if *first* is earlier than *second*, 1 if *first* is later than *second*, and 0 if *first* is equal to *second*.

2.2.2.9 struct tm* tm.decrement.day (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more days from a struct tm, decrementing the month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of days to deduct

Returns

A pointer to the same struct tm.

2.2.2.10 struct tm* tm.decrement.hour (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more hours from a struct tm, decrementing the day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of hours to deduct

Returns

A pointer to the same struct tm.

2.2.2.11 struct tm* tm.decrement.minute (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more minutes from a struct tm, decrementing the hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of minutes to deduct

Returns

A pointer to the same struct tm.

2.2.2.12 struct tm* tm.decrement.second (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more seconds from a struct tm, decrementing the minute, hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm struct to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of seconds to deduct

Returns

A pointer to the same struct tm.

2.2.2.13 `struct tm* tm_increment_day (struct tm * changing_tm, const int quantity)` [read]

Adds one or more days to a struct tm, incrementing the month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of days to add

Returns

A pointer to the same struct tm.

2.2.2.14 `struct tm* tm_increment_hour (struct tm * changing_tm, const int quantity)` [read]

Adds one or more hours to a struct tm, incrementing the day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of hours to add

Returns

A pointer to the same struct tm.

2.2.2.15 `struct tm* tm_increment_minute (struct tm * changing_tm, const int quantity)` [read]

Adds one or more minutes to a struct tm, incrementing the hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of minutes to add

Returns

A pointer to the same struct tm.

2.2.2.16 `struct tm* tm_increment_second (struct tm * changing_tm, const int quantity)` [read]

Adds one or more seconds to a struct tm, incrementing the minute, hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of seconds to add

Returns

A pointer to the same struct tm.

2.2.2.17 `int tm_intraday_secs_diff (const struct tm * first, const struct tm * second)`

Returns the difference between two struct tm structs.

Returns the difference between two struct tm structs. The structs are assumed to be within 24 hours of each other, and if they are not, the returned result is computed as if they were. For instance, comparing 10:00 on one day to 14:00 on the next day will yield a difference equivalent to 4 hours, not to 28 hours.

Parameters

<i>first</i>	The first struct tm struct
<i>second</i>	The second struct tm struct

Returns

The difference, in seconds, between the two struct tm structs. The difference is positive if *first* is earlier than *second*, and negative if *second* is earlier than *first*.

2.2.2.18 `bool validate_date (const struct tm * check_tm)`

Checks whether a supplied date is valid.

This function does not support leap seconds, and will return false if `check_tm->tm_sec == 60`.

Parameters

<i>check_tm</i>	A pointer to a struct tm containing the date to check.
-----------------	--

Returns

true if the date is valid, false otherwise.

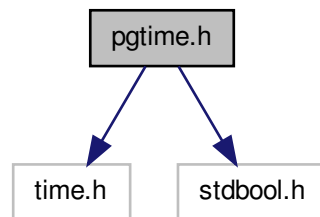
2.3 pgtime.h File Reference

Interface to miscellaneous time functions.

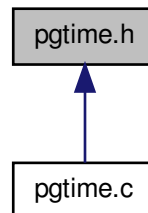
```
#include <time.h>
```

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

Include dependency graph for pgtime.h:



This graph shows which files directly or indirectly include this file:



Functions

- `time_t` [get_day_diff](#) (void)
Returns a time_t interval representing one day.
- `time_t` [get_hour_diff](#) (void)
Returns a time_t interval representing one hour.
- `time_t` [get_sec_diff](#) (void)
Returns a time_t interval representing one second.
- `bool` [validate_date](#) (const struct tm *check_tm)
Checks whether a supplied date is valid.
- `int` [tm_compare](#) (const struct tm *first, const struct tm *second)

- Compares two struct tm structs.*
- int `tm_intraday_secs_diff` (const struct tm *first, const struct tm *second)
Returns the difference between two struct tm structs.
- bool `is_leap_year` (const int year)
Checks if the supplied year is a leap year.
- struct tm * `tm_increment_day` (struct tm *changing_tm, const int quantity)
Adds one or more days to a struct tm, incrementing the month and/or the year as necessary.
- struct tm * `tm_increment_hour` (struct tm *changing_tm, const int quantity)
Adds one or more hours to a struct tm, incrementing the day, month and/or the year as necessary.
- struct tm * `tm_increment_minute` (struct tm *changing_tm, const int quantity)
Adds one or more minutes to a struct tm, incrementing the hour, day, month and/or the year as necessary.
- struct tm * `tm_increment_second` (struct tm *changing_tm, const int quantity)
Adds one or more seconds to a struct tm, incrementing the minute, hour, day, month and/or the year as necessary.
- struct tm * `tm_decrement_day` (struct tm *changing_tm, const int quantity)
Deducts one or more days from a struct tm, decrementing the month and/or the year as necessary.
- struct tm * `tm_decrement_hour` (struct tm *changing_tm, const int quantity)
Deducts one or more hours from a struct tm, decrementing the day, month and/or the year as necessary.
- struct tm * `tm_decrement_minute` (struct tm *changing_tm, const int quantity)
Deducts one or more minutes from a struct tm, decrementing the hour, day, month and/or the year as necessary.
- struct tm * `tm_decrement_second` (struct tm *changing_tm, const int quantity)
Deducts one or more seconds from a struct tm, decrementing the minute, hour, day, month and/or the year as necessary.
- bool `check_utc_timestamp` (const time_t check_time, int *secs_diff, const struct tm *check_tm)
Checks if a UTC timestamp is accurate.
- time_t `get_utc_timestamp` (const struct tm *utc_tm)
Gets a time_t timestamp for a requested UTC time.
- int `get_utc_timestamp_sec_diff` (const time_t check_time, const struct tm *check_tm)
Checks a time_t timestamp against a UTC time, and returns the difference in seconds.

2.3.1 Detailed Description

Interface to miscellaneous time functions.

Author

Paul Griffiths

Copyright

Copyright 2013 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

2.3.2 Function Documentation

2.3.2.1 bool check_utc_timestamp (const time_t check_time, int * secs_diff, const struct tm * check_tm)

Checks if a UTC timestamp is accurate.

Checks if a UTC timestamp is accurate. A time_t timestamp is computed from the supplied datetime, and compared to the supplied time_t timestamp. The difference between the two, in seconds, is stored in the supplied secs_diff argument. This function is needed because the methodology used to calculate a timestamp by this library can sometimes be inaccurate when leap seconds or other unpredictable calendar changes occur. We therefore need a method to check if the returned timestamp is accurate. Other functions provided in this library call this function, so the user should not normally need to call it.

Parameters

<i>check_time</i>	The time_t timestamp to check
<i>secs_diff</i>	Modified to contain the difference, in seconds
<i>check_tm</i>	A pointer to a struct tm containing the date to check.

Returns

true if the supplied timestamp is accurate, false otherwise

2.3.2.2 time_t get_day_diff (void)

Returns a time_t interval representing one day.

Returns a time_t interval representing one day. The C standard does not define the units in which a time_t value is measured. On POSIX-compliant systems it is measured in seconds, but we cannot assume this for full portability.

Returns

A time_t interval representing one day.

Exceptions

<i>bad_time</i>	if the current time cannot be obtained.
-----------------	---

2.3.2.3 time_t get_hour_diff (void)

Returns a time_t interval representing one hour.

Returns a time_t interval representing one hour. The C standard does not define the units in which a time_t value is measured. On POSIX-compliant systems it is measured in seconds, but we cannot assume this for full portability.

Returns

A time_t interval representing one hour.

Exceptions

<i>bad_time</i>	if the current time cannot be obtained.
-----------------	---

2.3.2.4 time_t get_sec_diff (void)

Returns a time_t interval representing one second.

Returns a time_t interval representing one second. The C standard does not define the units in which a time_t value is measured. On POSIX-compliant systems it is measured in seconds, but we cannot assume this for full portability.

Returns

A time_t interval representing one second.

Exceptions

<i>bad_time</i>	if the current time cannot be obtained.
-----------------	---

2.3.2.5 `time_t get_utc_timestamp (const struct tm * utc_tm)`

Gets a `time_t` timestamp for a requested UTC time.

Parameters

<i>utc_tm</i>	A pointer to a struct tm containing the UTC time.
---------------	---

Returns

A `time_t` timestamp for the requested UTC time.

2.3.2.6 `int get_utc_timestamp_sec_diff (const time_t check_time, const struct tm * utc_tm)`

Checks a `time_t` timestamp against a UTC time, and returns the difference in seconds.

This function only returns a good value if the timestamp is less than 24 hours away from the desired time, so the caller is responsible for making sure that it is. This function may also return a bad value if a leap second or other unpredictable calendar change falls between the desired UTC time and the provided time stamp. The result should therefore always be checked with [check_utc_timestamp\(\)](#), or by calling this function again.

Parameters

<i>check_time</i>	The <code>time_t</code> timestamp to check
<i>utc_tm</i>	A pointer to a struct tm against which to check.

Returns

The difference, if any, represented in seconds.

2.3.2.7 `bool is_leap_year (const int year)`

Checks if the supplied year is a leap year.

Checks if the supplied year is a leap year.

Parameters

<i>year</i>	A year
-------------	--------

Returns

`true` if *year* is a leap year, `false` otherwise.

2.3.2.8 `int tm_compare (const struct tm * first, const struct tm * second)`

Compares two struct tm structs.

Compares two struct tm structs. Only the year, month, day, hour, minute and second are compared. Any timezone or DST information is ignored.

Parameters

<i>first</i>	The first struct tm struct.
<i>second</i>	The second struct tm struct.

Returns

-1 if *first* is earlier than *second*, 1 if *first* is later than *second*, and 0 if *first* is equal to *second*.

2.3.2.9 struct tm* tm.decrement.day (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more days from a struct tm, decrementing the month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of days to deduct

Returns

A pointer to the same struct tm.

2.3.2.10 struct tm* tm.decrement.hour (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more hours from a struct tm, decrementing the day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of hours to deduct

Returns

A pointer to the same struct tm.

2.3.2.11 struct tm* tm.decrement.minute (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more minutes from a struct tm, decrementing the hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of minutes to deduct

Returns

A pointer to the same struct tm.

2.3.2.12 struct tm* tm.decrement.second (struct tm * *changing_tm*, const int *quantity*) [read]

Deducts one or more seconds from a struct tm, decrementing the minute, hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm struct to decrement. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of seconds to deduct

Returns

A pointer to the same struct tm.

2.3.2.13 `struct tm* tm_increment_day (struct tm * changing_tm, const int quantity)` [read]

Adds one or more days to a struct tm, incrementing the month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of days to add

Returns

A pointer to the same struct tm.

2.3.2.14 `struct tm* tm_increment_hour (struct tm * changing_tm, const int quantity)` [read]

Adds one or more hours to a struct tm, incrementing the day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of hours to add

Returns

A pointer to the same struct tm.

2.3.2.15 `struct tm* tm_increment_minute (struct tm * changing_tm, const int quantity)` [read]

Adds one or more minutes to a struct tm, incrementing the hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of minutes to add

Returns

A pointer to the same struct tm.

2.3.2.16 `struct tm* tm_increment_second (struct tm * changing_tm, const int quantity)` [read]

Adds one or more seconds to a struct tm, incrementing the minute, hour, day, month and/or the year as necessary.

Parameters

<i>changing_tm</i>	A pointer to the struct tm to increment. The struct referred to by the pointer is modified by the function.
<i>quantity</i>	The number of seconds to add

Returns

A pointer to the same struct tm.

2.3.2.17 `int tm_intraday_secs_diff (const struct tm * first, const struct tm * second)`

Returns the difference between two struct tm structs.

Returns the difference between two struct tm structs. The structs are assumed to be within 24 hours of each other, and if they are not, the returned result is computed as if they were. For instance, comparing 10:00 on one day to 14:00 on the next day will yield a difference equivalent to 4 hours, not to 28 hours.

Parameters

<i>first</i>	The first struct tm struct
<i>second</i>	The second struct tm struct

Returns

The difference, in seconds, between the two struct tm structs. The difference is positive if *first* is earlier than *second*, and negative if *second* is earlier than *first*.

2.3.2.18 `bool validate_date (const struct tm * check_tm)`

Checks whether a supplied date is valid.

This function does not support leap seconds, and will return false if `check_tm->tm_sec == 60`.

Parameters

<i>check_tm</i>	A pointer to a struct tm containing the date to check.
-----------------	--

Returns

true if the date is valid, false otherwise.

Index

- check_utc_timestamp
 - pgtime.c, [5](#)
 - pgtime.h, [12](#)
- get_day_diff
 - pgtime.c, [6](#)
 - pgtime.h, [13](#)
- get_hour_diff
 - pgtime.c, [6](#)
 - pgtime.h, [13](#)
- get_sec_diff
 - pgtime.c, [6](#)
 - pgtime.h, [13](#)
- get_utc_timestamp
 - pgtime.c, [6](#)
 - pgtime.h, [13](#)
- get_utc_timestamp_sec_diff
 - pgtime.c, [7](#)
 - pgtime.h, [14](#)
- is_leap_year
 - pgtime.c, [7](#)
 - pgtime.h, [14](#)
- main
 - main.c, [4](#)
- main.c, [3](#)
 - main, [4](#)
- pgtime.c, [4](#)
 - check_utc_timestamp, [5](#)
 - get_day_diff, [6](#)
 - get_hour_diff, [6](#)
 - get_sec_diff, [6](#)
 - get_utc_timestamp, [6](#)
 - get_utc_timestamp_sec_diff, [7](#)
 - is_leap_year, [7](#)
 - tm_compare, [7](#)
 - tm_decrement_day, [8](#)
 - tm_decrement_hour, [8](#)
 - tm_decrement_minute, [8](#)
 - tm_decrement_second, [8](#)
 - tm_increment_day, [9](#)
 - tm_increment_hour, [9](#)
 - tm_increment_minute, [9](#)
 - tm_increment_second, [9](#)
 - tm_intraday_secs_diff, [10](#)
 - validate_date, [10](#)
- pgtime.h, [11](#)
 - check_utc_timestamp, [12](#)
 - get_day_diff, [13](#)
 - get_hour_diff, [13](#)
 - get_sec_diff, [13](#)
 - get_utc_timestamp, [13](#)
 - get_utc_timestamp_sec_diff, [14](#)
 - is_leap_year, [14](#)
 - tm_compare, [14](#)
 - tm_decrement_day, [15](#)
 - tm_decrement_hour, [15](#)
 - tm_decrement_minute, [15](#)
 - tm_decrement_second, [15](#)
 - tm_increment_day, [16](#)
 - tm_increment_hour, [16](#)
 - tm_increment_minute, [16](#)
 - tm_increment_second, [16](#)
 - tm_intraday_secs_diff, [17](#)
 - validate_date, [17](#)
- tm_compare
 - pgtime.c, [7](#)
 - pgtime.h, [14](#)
- tm_decrement_day
 - pgtime.c, [8](#)
 - pgtime.h, [15](#)
- tm_decrement_hour
 - pgtime.c, [8](#)
 - pgtime.h, [15](#)
- tm_decrement_minute
 - pgtime.c, [8](#)
 - pgtime.h, [15](#)
- tm_decrement_second
 - pgtime.c, [8](#)
 - pgtime.h, [15](#)
- tm_increment_day
 - pgtime.c, [9](#)
 - pgtime.h, [16](#)
- tm_increment_hour
 - pgtime.c, [9](#)
 - pgtime.h, [16](#)
- tm_increment_minute
 - pgtime.c, [9](#)
 - pgtime.h, [16](#)
- tm_increment_second
 - pgtime.c, [9](#)
 - pgtime.h, [16](#)
- tm_intraday_secs_diff
 - pgtime.c, [10](#)
 - pgtime.h, [17](#)
- validate_date
 - pgtime.c, [10](#)
 - pgtime.h, [17](#)

pgtime.c, [10](#)

pgtime.h, [17](#)