unique.timestamp 0.9.0 API

A `unique.timestamp` provides a universally unique identifier which is also a human-readable timestamp. It represents **UTC** time, and it returns the time at the moment it was invoked.

The format of a `unique.timestamp` is

`YYYYmmdd_HHMM_SS_xxxxxx.zzzzzz.cccc`

Where:

`YYYY` is the year

`mm` is the month

`dd` is the day of the month

`HH` is the hour

`MM` is the minute

`SS` is the second

`xxxxxx` is the milliseconds

`zzzzzz` is a process pid

`cccc` is a count for that pid

For example,

```
20090612_0608_56_510702.002621.0000
```

Provided a host machine's system clock continues to move forward only, this timestamp is guaranteed to be unique for that host.

If you have a collection of different hosts all generating `unique.timestamp`s, then there is a statistical probability that some will end up being the same. This probability is dependent on the number of hosts, and the frequency with which the timestamps are being generatied. It is possible to extend `unique.timestamp` in a straight forward fashion so that this probability is as close to zero as is required.

Command line API

The following executables are available from the UNIX command line (shell):

`unique.timestamp`

syntax:

```
unique.timestamp
```

returns a `unique.timestamp` as a string to stdout.

ts2secs

syntax:

```
ts2secs unique.timestamp
```

Converts the given unique timestamp to the number of seconds since epoch.

tsdiff

syntax:

```
tsdiff unique.timestamp1 unique.timestamp2
```

Returns the number of seconds between the two unique timestamps

C library API

The following C functions are available in source code form:

uniquetimestamp_long

```
int uniquetimestamp_long(char *ts)
```

Generates a unique timestamp as a string, and stores it at the specified address *ts*

uniquetimestamp_long_pidcount

```
int uniquetimestamp_long_pidcount(char *ts, int pid, int *count)
```

Produces a `unique.timestamp`, and stores it in *ts*. The value for `pid` is specified explicitly in *pid*, and the count is taken from a value pointed to by *count*.

Use this function when you want to specify the `pid` explicitly, and maintain the `count` yourself.

uniquetimestamp_long_pidcount2time_t

```
time_t uniquetimestamp_long_pidcount2time_t(char *ts)
```

Scans a `unique.timestamp`, given as a string, and converts it to a `time_t` value. the *microseconds*, *pid* and *count* parts are ignored.