

---+ Package TWiki::Time

Time handling functions.

## **StaticMethod parseTime (\$szDate, \$defaultLocal) -> \$iSecs**

Convert string date/time string to seconds since epoch (1970-01-01T00:00:00Z).

- \$sDate - date/time string

Handles the following formats:

Default TWiki format

- 31 Dec 2001 - 23:59

TWiki format without time (defaults to 00:00)

- 31 Dec 2001

Date separated by '/', '.' or '-', time with ':' or '.' Date and time separated by ' ', '.' and/or '-'

- 2001/12/31 23:59:59
- 2001.12.31.23.59.59
- 2001/12/31 23:59
- 2001.12.31.23.59
- 2001-12-31 23:59
- 2001-12-31 - 23:59

ISO format

- 2001-12-31T23:59:59

ISO dates may have a timezone specifier, either Z or a signed difference in hh:mm format. For example:

- 2001-12-31T23:59:59+01:00
- 2001-12-31T23:59Z

The default timezone is Z, unless \$defaultLocal is true in which case the local timezone will be assumed.

If the date format was not recognised, will return 0.

## **StaticMethod formatTime (\$epochSeconds, \$formatString, \$outputTimeZone) -> \$value**

- \$epochSeconds epochSecs GMT
- \$formatString twiki time date format, default \$day \$month \$year - \$hour:\$min
- \$outputTimeZone timezone to display, gmtime or servertime, default is whatever is set in \$TWiki::cfg{DisplayTimeValues}

\$formatString supports:

\$seconds	secs
\$minutes	mins
\$hours	hours
\$day	date
\$wday	weekday name
\$dow	day number (0 = Sunday)
\$week	week number
\$month	month name
\$mo	month number
\$year	4-digit year
\$ye	2-digit year
\$http	ful HTTP header format date/time
\$email	full email format date/time
\$res	full RCS format date/time
\$epoch	seconds since 1st January 1970

## StaticMethod formatDelta (\$s) -> \$string

Format a time in seconds as a string. For example, "1 day, 3 hours, 2 minutes, 6 seconds"

## StaticMethod parseInterval (\$szInterval) -> [\$iSecs, \$iSecs]

Convert string representing a time interval to a pair of integers representing the amount of seconds since epoch for the start and end extremes of the time interval.

- \$szInterval - time interval string

in yacc syntax, grammar and actions:

```
interval ::= date { $$start = fillStart($1); $$end = fillEnd($1); }
          | date '/' date { $$start = fillStart($1); $$end = fillEnd($3); }
          | 'P' duration '/' date { $$start = fillEnd($4)-$2; $$end = fillEnd($4); }
          | date '/' 'P' duration { $$start = fillStart($1); $$end = fillStart($1)+$4; }
          ;
```

an interval may be followed by a timezone specification string (this is not supported yet).

duration has the form (regular expression):

```
P(<number><nameOfDuration>)+
```

nameOfDuration may be one of:

- y(year), m(month), w(week), d(day), h(hour), M(minute), S(second)

date follows ISO8601 and must include hypens. (any amount of trailing elements may be omitted and will be filled in differently on the differents ends of the interval as to include the longest possible interval):

- 2001-01-01T00:00:00
- 2001-12-31T23:59:59

StaticMethod formatTime(\$epochSeconds,\$formatString,\$outputTimeZone)-> \$value

timezone is optional. Default is local time.

If the format is not recognised, will return empty interval [0,0].

TODO: timezone testing, especially on non valid strings

---

This topic: TWiki > TWikiTimeDotPm

Topic revision: r3 - 2008-01-22 - TWikiContributor



Copyright © 1999-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback

**Note:** Please contribute updates to this topic on TWiki.org at TWiki:TWiki.TWikiTimeDotPm.