

RdateTZ.man

COLLABORATORS

	TITLE : RdateTZ.man		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	RdateTZ.man	1
1.1	RdateTZ	1
1.2	RdateTZ DESCRIPTION	1
1.3	RdateTZ OPTIONS	2
1.4	RdateTZ PREREQUISITES	3
1.5	RdateTZ CONFIGURATION	3
1.6	RdateTZ CONFIGURATION - time zone information	5
1.7	RdateTZ EXAMPLES	6
1.8	RdateTZ DIAGNOSTICS	7
1.9	RdateTZ BUGS/LIMITATIONS	8
1.10	RdateTZ AUTHOR	9
1.11	RdateTZ LICENCE	9

Chapter 1

RdateTZ.man

1.1 RdateTZ

Use the time returned by one of a number of remote hosts to set the local clock and timezone

```
SYNOPSIS RdateTZ [[ SERVER ] <hostname> ... ]
          [ WITHIN <limit> [ SEC | MIN ]] [ ZONE <zone> ]
          [ SAVE ] [ VERBOSE ] [ TRIAL ]
          SetTZ [ ZONE ] <zone> [ SAVE ] [ VERBOSE ] [ TRIAL ]
```

DESCRIPTION

OPTIONS

PREREQUISITES

CONFIGURATION

EXAMPLES

DIAGNOSTICS

BUGS/LIMITATIONS

AUTHOR

LICENCE

1.2 RdateTZ DESCRIPTION

RdateTZ is an alternative to utilities such as SynClock, which set the local clock in accordance with the time obtained by querying the 'daytime' port (13) of a remote host. RdateTZ improves on SynClock by:

- 1 Using the 'time' port (37) of the remote host, which returns the GMT value of the remote host's clock. (The 'daytime' port (13) normally returns the local time at the remote host's site, but could return GMT and is not required to identify the timezone - see RFC867.)

- 2 (Optionally) ignoring a remote host whose clock is too far out of step with the local clock (reducing the risk of synchronising with a clock which is running fast or slow).

- 3 Using datagrams rather than a stream, which allows several remote hosts to be queried simultaneously. The first

acceptable (see 2 above) reply is used.

4 Using a file of timezone specifications to set the local clock to a specified timezone, allowing for daylight saving time if appropriate. A subset of the XPG3 standard for the TZ environment variable is used, which allows the offsets and daylight saving time start and end dates of most timezones to be encoded.

5 Setting a value identifying the selected timezone in the TZ environment variable, and recording an RFC822-compliant timezone identifier in the comment associated with the file holding this environment variable.

SetTZ provides a subset of the RdateTZ interface, which allows the local clock to be set to a specified timezone without changing the GMT value of the clock. This can be called each time the machine is rebooted, or in the early hours of every morning, to switch to or from daylight saving time on the appropriate dates. No attempt is made to access any external host.

1.3 RdateTZ OPTIONS

RdateTZ supports the following command line options:

SERVER <hostname> ...

A list of host names (or IP addresses in dot notation) specifying the remote hosts to be queried. Queries are sent to the hosts in turn until an acceptable response is received or the list is exhausted. If necessary there will be a short pause after the last query is sent.

The fastest acceptable reply is used - once this has been received RdateTZ will not wait for any other reply.

The SERVER keyword is optional. If no <hostname> is given the current system clock time is used. If at least one <hostname> is given but no acceptable reply is received, RdateTZ exits without changing the clock or timezone.

WITHIN <limit> [SEC | SECS | MIN | MINS]

The maximum acceptable change to the current system clock time. This can be used to prevent a slightly inaccurate system clock time being changed to accord with a wildly inaccurate remote clock time.

Both RdateTZ and SetTZ support the following command line options:

ZONE <zone> A timezone name identifying the timezone specification (from the @{ " TZ.dat " LINK CONFIGURATION} file) which is to be used to set the system clock and the TZ environment variable. ↵

The ZONE parameter may be omitted when RdateTZ is called. The system clock will then be adjusted to accord with a remote host, without changing the current offset from GMT – in particular, the offset of the clock will not be changed when daylight saving time has just started or ended. This avoids a potentially dangerous backward adjustment of the system clock while the system is active.

The ZONE parameter must be specified when SetTZ is called. Be aware that if you use (a clone of) the 'make' utility a large backward change to the system clock can be disastrous.

SAVE If SAVE is specified, the new value of the system clock is saved to the battery-backed clock, and the new value of TZ is saved to envarc:TZ.

VERBOSE If VERBOSE is specified, detailed diagnostic messages are output.

TRIAL If TRIAL is specified, detailed diagnostic messages are output and no change is made to the system or battery-backed clock, nor to TZ or envarc:TZ. This makes it possible to safely find out what effect a call to RdateTZ would have.

1.4 RdateTZ PREREQUISITES

RdateTZ needs AmiTCP (bsdsocket.library version 2 or later) and WorkBench 2.04 or later.

The TZ environment variable must be set before RdateTZ or SetTZ is called, so that the current offset from GMT of the system clock can be established using @{ " TZ.dat " LINK CONFIGURATION}. Set envarc:TZ to reflect the offset from GMT of the battery-backed clock, so that TZ is set when the system is rebooted.

(It would be possible to run with the battery-backed clock set permanently to GMT and envarc:TZ set to "GMT0", and use SetTZ to set the system clock whenever the machine is rebooted. If you do this, be aware that SetClock cannot safely be used after SetTZ.)

1.5 RdateTZ CONFIGURATION

The file s:TZ.dat is used to hold specifications of a number of timezones. As issued, the file contains details of the major European timezones, and several relatively obscure timezones used for testing.

The environment variable TZ is used in conjunction with s:TZ.dat to record the offset of the system clock from GMT. The issued version of s:TZ.dat

assumes that the value in TZ is the value needed by the SAS/C library function `__tzset` - i.e a three-letter abbreviation for the standard timezone, followed by a signed number of hours which must be added to the local time to give GMT, followed (if daylight saving time is in effect) by a three-letter abbreviation for the daylight saving timezone.

It will be necessary to change `s:TZ.dat` if:

the timezone you wish to use is not in the file

or the value you need in TZ is different from the value needed by the `__tzset` function - the value may take any form you need, provided it does not contain a comma or colon.

Any line in `s:TZ.dat` which begins with a space, tab or newline character will be ignored (there is no special comment delimiter). The other lines each contain a comma-separated list of keys followed by a colon, followed by the time zone information. For illustration, here are some lines from the file:

```
GMT0,GMT-1BST, UK, Eire: GMT0BST,M3.5.0/1,M10.5.0/2
GMT0,, GMT, Greenwich: GMT0
THT10,THT9THD, THT, Rarotonga: THT10THDT9:30,M10.5.0/0,M3.1.0/0:30
AST4,AST3ADT, AST, Asuncion: AST4ADT,J274/0,J91/0
AST4,AST3ADT, AST, Stanley: AST4ADT,M9.2.6/24,M4.3.6/24
```

Comparison with the keys is not case-sensitive. Any of the keys may be followed by spaces or tabs, and any but the first may be preceded by spaces or tabs.

The first key must be the value of the TZ environment variable when daylight saving time is not in effect, and the second key must be the TZ value when daylight saving time is in effect (or if there is no daylight saving time must be blank). The remaining keys are names for the timezone. If a non-unique key is given, the first matching specification is used. You may wish to copy your timezone specification to the start of the file (replacing the copy of the UK timezone specification) to avoid any problems caused by duplicate keys.

The current value of TZ is used to give the offset from GMT of the system clock, by finding a line in `s:TZ.dat` where TZ matches one of the first two keys. If the first key is matched, standard time is assumed, otherwise daylight saving time. The offset is NOT derived from the value of TZ, which can only give a whole number of hours - compare the daylight saving time offset for Rarotonga with the corresponding value for TZ. Note that although a TZ value such as "AST4" may not identify a unique line in `s:TZ.dat`, it is likely that each of the alternative lines will convert the TZ value to the same offset.

The value given for the ZONE parameter should be one of the remaining keys. Note that in the lines quoted for illustration, the TZ values "AST4" and "AST3ADT", and the key 'AST', will all select the 'Asuncion' timezone (Paraguay) so the key 'Stanley' must be used to get the right start and end dates for daylight saving time in the Falklands.

A good source of information on timezone offsets and daylight saving time start and end dates is `'ftp://elsie.nci.nih.gov/pub/'`. The file

'tzdata95i.tar.gz' (this name will change when the data is updated) contains hundreds of kilobytes of rules and comments in the data files 'europe', 'asia', 'africa' and so on. These compilations are of course copyright, even though the individual contributions may not be.

If you're in Europe and you've noticed that the rule in the issued s:TZ.dat file is different from the rule on your UNIX system, get the UNIX system fixed. The UK rule changed in about 1990, affecting 1995 but not 1990-1994, and the EEC rules change for 1996 onwards, affecting all EEC countries including the UK. Some UNIX systems are still applying the 1989 rule.

1.6 RdateTZ CONFIGURATION - time zone information

The form of the time zone information is based on the XPG3 specification of the TZ environment variable. Spaces are allowed only in timezone designations, where they are significant. The following description closely follows the XPG3 specification, except for the paragraphs starting ****CLARIFICATION****.

```
<std><offset>[<dst>[<offset>],<start>[/<time>],<end>[/<time>]]
```

Where:

<std> and <dst>

Are each three or more bytes that are the designation for the standard (<std>) and daylight savings time (<dst>) timezones. Only <std> is required - if <dst> is missing, then daylight savings time does not apply in this locale. Upper- and lower-case letters are allowed. Any characters except a leading colon (:), digits, a comma (,), a minus (-) or a plus (+) are allowed.

****CLARIFICATION**** The two-byte designation 'UT' is permitted.

<offset>

Indicates the value one must add to the local time to arrive at Coordinated Universal Time. The offset has the form:
<hh>[:<mm>[:<ss>]]

The minutes <mm> and seconds <ss> are optional. The hour <hh> is required and may be a single digit. The offset following <std> is required. If no offset follows <dst>, daylight savings time is assumed to be one hour ahead of standard time. One or more digits may be used; the value is always interpreted as a decimal number. The hour must be between 0 and 24, and the minutes (and seconds) if present between 0 and 59. Out of range values may cause unpredictable behavior. If preceded by a '-', the timezone is east of the Prime Meridian; otherwise it is west (which may be indicated by an optional preceding '+' sign).

****CLARIFICATION**** No more than two digits are allowed in any of <hh>, <mm> or <ss>. Leading zeros are permitted.

<start>/<time> and <end>/<time>

Indicates when to change to and back from daylight savings time, where <start>/<time> describes when the change from standard time to daylight savings time occurs, and <end>/<time> describes when the change back happens. Each <time> field describes when, in current local time, the change is made.

****CLARIFICATION**** It is recognized that in the Southern hemisphere <start> will specify a date later than <end>.

The formats of <start> and <end> are one of the following:

J<n> The Julian day <n> (1 < <n> < 365). Leap days are not counted. That is, in all years, February 28 is day 59 and March 1 is day 60. It is impossible to refer to the occasional February 29.

<n> The zero-based Julian day (0 < <n> < 365). Leap days are counted, and it is possible to refer to February 29.

M<m>.<n>.<d>

The <d>th day, (0 < <d> < 6) of week <n> of month <m> of the year (1 < <n> < 5, 1 < <m> < 12), where week 5 means 'the last <d>-day in month <m>' (which may occur in either the fourth or the fifth week). Week 1 is the first week in which the <d>th day occurs. Day zero is Sunday.

****CLARIFICATION**** Neither <n> nor <m> may have a leading zero. <d> must be a single digit.

****CLARIFICATION**** If <dst> is given then <start> and <end> are required - there is no 'implementation specific default'.

<time> has the same format as <offset> except that no leading sign ('-' or '+') is allowed. The default, if <time> is not given is 02:00:00.

****CLARIFICATION**** The number of hours in <time> may be up to 167, to allow encoding of rules such as '00:00hrs on the Sunday after the second Friday in September'

1.7 RdateTZ EXAMPLES

Assuming that envarc:TZ and the battery-backed clock are consistent but that there may have been a change to or from daylight saving time, the following command (in sys:S/User-Startup or in a 'cron' script) will change the system clock and TZ to the UK timezone. Diagnostic messages are logged.

```
SetTZ UK VERBOSE >>t:TZ.log
```

The following commands will change the system clock and TZ to the UK timezone, and also copy the RFC822-compliant timezone designation to the TimeZone environment variable. (Note that on workBench 2.04 and possibly on later versions the file comment is NOT copied when env:TZ is set from envarc:TZ when Startup-Sequence is executed during a reboot.)

```
SetTZ UK
SetEnv TimeZone `List env:TZ LFORMAT %C`
```

The following command will set both the system clock and the battery-backed clock to the Central Europe timezone.

```
SetTZ MET SAVE
```

The following command will (provided the machine is on-line via AmiTCP) synchronize the system clock to the clock in the remote host 'gate.demon.co.uk', without changing the timezone in use locally - the system clock will not jump back an hour if daylight saving time has just ended. If the local and remote clocks do not agree to within 30 seconds the local clock is unchanged. Diagnostic messages are logged.

```
RdateTZ gate.demon.co.uk WITHIN 30 SEC VERBOSE >>t:TZ.log
```

Annotated examples of actual use may be found in the description of the diagnostic output .

1.8 RdateTZ DIAGNOSTICS

Using either of the keywords VERBOSE or TRIAL results in the output of self-explanatory messages. The following examples are each followed by additional notes:

```
> RdateTZ ftp.luth.se post gate ftp.funet.fi demon-du ZONE Rarotonga VERBOSE
Current time zone +0000 (GMT)
Changing to standard time zone -1000 (THT)      "THT10"
    daylight saving time -0930 (THDT)      "THT9THD"
DST starts on the last Sunday in October
    ends on the first Sunday in March
Trying server "ftp.luth.se"
Trying server "post.demon.co.uk"
Trying server "gate.demon.co.uk"
Server "ftp.funet.fi" unknown
Accepted reply from "post.demon.co.uk"
DST begins at 00:00:00 on Sunday 29 October 1995
    ends at 00:30:00 on Sunday 5 March 1995
Using daylight saving time
TZ set to "THT9THD" (comment "-0930 (THDT)")
System clock put BACK 9 hours 30 mins 1 secs
```

The offsets from GMT in the form '-hhmm (ddd)' give the time in hours and minutes which must be added to GMT to give local time in the selected timezone, and in brackets its designation. This identification may be used in RFC822-compliant mail headers (the bracketed portion will be treated as a comment).

The offsets in values in TZ give the time in hours which must be added to local time to give GMT. The two descriptions of an offset will have opposite signs (unless the offset is zero). Note that in the example the offset of daylight saving time is not a whole number of hours and cannot be encoded in TZ.

Rarotonga is in the Southern hemisphere, so the start date for daylight saving time is later than the end date. This first example was generated on 4 Mar 1995, just before the end of daylight saving time.

The first host was about twelve hops away. The second host was only two hops away and its reply was received before any reply from the first host - but not quickly enough to prevent a query being sent to the third host and an attempt to resolve the name of the fourth host, which took so long to fail that no query was sent to the fifth host.

Putting the system clock back this far can wreak havoc on your file timestamps.

```
> RdateTZ post.demon.co.uk WITHIN 30 SEC VERBOSE
Current time zone -0930 (THDT)
Trying server "post.demon.co.uk"
Accepted reply from "post.demon.co.uk"
TZ set to "THT9THD" (comment "-0930 (THDT)")
System clock put forward 7 secs
```

This example was generated shortly after the switch back to standard time. Since no ZONE parameter was specified, the system clock was kept on daylight saving time.

```
> RdateTZ 158.152.6.65 ZONE Rarotonga TRIAL
Current time zone -0930 (THDT)
Changing to standard time zone -1000 (THT)           "THT10"
      daylight saving time -0930 (THDT)           "THT9THD"
DST starts on the last Sunday in October
      ends on the first Sunday in March
Trying server "clootie.demon.co.uk"
Accepted reply from "clootie.demon.co.uk"
DST begins at 00:00:00 on Sunday 29 October 1995
      ends at 00:30:00 on Sunday 5 March 1995
Using standard time
TZ set to "THT10" (comment "-1000 (THT)")
System clock put BACK 30 mins 5 secs
```

This time ZONE was specified and the system clock was switched to standard time.

The server was specified by its IP address, but in the diagnostic messages it is identified by name.

1.9 RdateTZ BUGS/LIMITATIONS

December and January

RdateTZ uses the GMT time to determine the year, then calculates the start and end dates for daylight saving time in order to decide which offset from GMT to use. If the GMT time is close to the end of December or the beginning of January the wrong year may be used.

It is also possible to invent rules which result in a change occurring in December in some years and January in others. There will then be one or three changes in a year rather than the expected two.

Either of the cases described above may result in RdateTZ wrongly selecting, or failing to select, daylight saving time.

1.10 RdateTZ AUTHOR

Dave.Sparks@sisyphus.demon.co.uk (as at 1 Nov 1995)

1.11 RdateTZ LICENCE

RdateTZ is free software; you can redistribute it provided that:

- 1 no charge is made for the program
- 2 all the original files are included intact in the distribution
- 3 any additional or alternative files are clearly identified as such

RdateTZ is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.