Time for Julia

Jeffrey A. Sarnoff

2012-Aug-30

Preface

These page provide an admixture of user and technical documentation to introduce Julia's evolving sense of time. The current pre-release version is intended to offer a structurally sound interface. The underpinnings are subject to refinement and, where appropriate, less delicate adjustment.

Thanks to all Julia dev-list members who have responded to my many questions.

Julia is

Julia is a new programming language and environment embodying the best of applicable current practice and advancing the art of enjoyable effectiveness in software development. Or, we'll get there.

The Right Place

At present, the most current pre-release version of tm4julia is a subdirectory tree available from GitHub: < https://github.com/JeffreySarnoff/jtm >. Clone or otherwise copy it under your julia directory. Before starting, edit the file tm4julia/config/tz/MyTimezone.txt to hold your IANA standard timezone name (in a future release, this will not be needed iff your "TZ" environment variable is properly set). It is a good idea to confirm the standard timezone name as they are not self-evident, < http://www.pageloom.com/automatic-timezone-detection-with-javascript > may be helpful (with javascript enabled, the timezone name appears in a black stripe).

That done, start Julia and require("tm4julia/tm4julia.jl"). The datetime facility should initialize or, if not, it should tell you why.

Typographic Conventions

User input and exemplar source code uses this typeface.
`Responses and items print()ed or show()n on screen use this typeface.`

Enter, when shown explicitly: ↵

Table of Contents

Enter dates, times and timezones

Start Julia. Here are examples with America/New_York (EST,EDT) as the local timezone:

```
julia> require("jtm/tm4julia.jl")
julia> # enter dates with date(y,m,d, tmzone_name)
       # enter times with time(h,m,s, tmzone_name)
       # enter dates and times-of-day with date(y,m,d,h,m,s, tmzone_name)
julia> # dates are given with respect to an IANA standard timezone
       # or given as UTC (respecting leap seconds), or as UT (no leap seconds),
       # and the local timezone is used when no timezone is specified
julia> local_date = date(2002,11,18)
d"2002-11-18 EST"
julia> other_date = date(2002,11,18,"America/Los_Angeles")
d"2002-11-18 PST"
julia> utc_date = date(2002,11,18,"UTC")
d"2002-11-18 UTC"
julia> local_datetime = date(2002,7,18,5,15,55)
d"2002-07-18 05:15:55 EDT"
julia> other_datetime = date(2002,7,18,5,15,55,"America/Los_Angeles")
d"2002-07-18 05:15:55 PDT"
julia> utc_datetime = date(2002,7,18,5,15,55,"UTC")          # uses leap seconds
d"2002-07-18 05:15:55 UTC"
julia> ut_datetime = date(2002,7,18,5,15,55,"UT")            # ignores leap seconds
d"2002-07-18 05:15:55 UT"
julia> # times relate to a standard timezone (shown in parenthesis)
       # until associated with a date, unknown whether Standard Time or Saving Time
julia> local_time = time(2,11,18)
t"02:11:18 (New_York)"
julia> other_time = time(2,11,18,"America/Los_Angeles")
t"02:11:18 (Los_Angeles)"
julia> utc_time = time(2,11,18,"UTC")
t"02:11:18 (UTC)"
julia> date(2002,7,18) + time(2,11,18)
d"2002-07-18 02:11:18 EDT"
julia> # if the timezones differ, the time's timezone will be shifted into the date's timezone
julia> date(2002,7,18,"America/Los_Angeles") + time(3,11,18,"America/New_York")
d"2002-07-18 00:11:18 PDT"
julia> #  end examples
```

Enter intervals: datespans, timespans

Start Julia. Here are examples; time intervals are independent of timezone:

```
julia> require("jtm/tm4julia.jl")
julia> # enter spans of multiple days with ymd_span(y,m,d) or tspan(y,m,d,0,0,0)
       # enter spans less than on day with hms_span(h,m,s) or tspan(0,0,0,h,m,s)
       # enter mixed datetime spans with tspan(y,m,d,h,m,s)
julia> span_years = ymd_span(2,0,0)
t"02-00-00"
julia> span_days = ymd_span(0,0,18)
t"00-00-18"
julia> span_minutes = hms_span(0,15,0)
t"00-00-00 00:15:00"
julia> span_datetime = ymd_span(0,7,18,5,15,55)
t"00-07-18 05:15:55"
julia> span_yearhour = ymd_span(1,0,0,6,0,0)
t"01-00-00 06:00:00"
julia> # timespans add/subtract with themselves and with dates, datetimes, times
julia> ymd_span(0,1,0) + hms_span(6,0,0)
t"00-01-00 06:00:00"
julia> date(2002,7,18,5,15,55,"UTC") + ymd_span(0,1,0)
d"2002-08-18 05:15:55 UTC"
julia> date(2002,7,18,5,15,55)  - hms_span(5,15,55)
d"2002-07-18 00:00:00 EDT"
julia> # timespans can be created by subtracting dates or datetimes
julia> date(1994,7,18,"UTC") - date(1993,1,3,"UTC")  # using "UTC" respects leap seconds
t"01-06-15 00:00:01"
julia> date(1994,7,18,"UT")  - date(1993,1,3,"UT")     # using "UT" ignores leap seconds
t"01-06-15"
julia> #  end examples
```

Convert into other timezones

Start Julia. Here are examples with America/New_York (EST,EDT) as the local timezone:

```
julia> require("jtm/tm4julia.jl")
julia> # timezone conversion is done using date(date_given, target_timezone)
       # dates without a  time-of-day, are assigned 00:00:00 before timezone conversion
julia> local_date = date(2002,11,18)
d"2002-11-18 EST"
julia> convert_timezone = date(local_date,"America/Los_Angeles")
d"2002-11-17 21:00:00 PST"
julia> utc_date = date(2002,11,18,"UTC")
d"2002-11-18 UTC"
julia> convert_timezone = date(utc_date)     # default is to use local timezone
d"2002-11-18 05:00:00 EST"
julia> utc_date == date(utc_date,"UTC")      # converting to given timezone does nothing
true
julia> local_datetime = date(2002,7,18,5,15,55)
d"2002-07-18 05:15:55 EDT"
julia> convert_timezone = date(local_datetime,"UTC")
d"2002-07-18 09:15:55 UTC"
julia> local_datetime – convert_timezone     # timezone conversion does not move time
t"00-00-00 00:00:00"
julia> #  end examples
```

Units of Time: combine, apply

Units of time are written with a preceeding underscore. Integral multiples of any time unit may be formed by prefixing the integer: fourteen_days = 14_dy. Units of year, quarter and month may be added or subtracted together to yield an integral number of months. Units of week, day, hour, minute, second, [microsec] may be added or subtracted together to yield an integral number of seconds [microsecs]. Adding or subtracting any units from the first group with any units from the second group is disallowed. These are the time units and postfix forms available with the current pre-release version:

year	quarter	month		week	day	hour	minute	second	microsec
_yr	_qr	_mo		_wk	_dy	_hr	_mi	_sc	_us

Here are a few examples of their use and mixing time units with datetimes or timespans:

```
julia> require("jtm/tm4julia.jl")
julia> span_two_years = ymd_span(2,0,0)
t"02-00-00"
julia> two_years = 2_yr
2_yr
julia> span_four_years = span_two_years + two_years
t"04-00-00"
julia> local_date = date(2010,10,20,9,30,0)
d"210-10-20 09:30:00"
julia>local_date + two_years == local_date + span_two_years
true
julia> 2_yr - 6_mo
18_mo
julia> 3_mo - 1_dy
error:  Cannot subtract incompatible time units.
julia> # highlight distinction between datetime + timespans, datetime + time units
julia> # end examples
```