

rfc3339.scm

RFC3339 Date and Time Format in Scheme

Version 0.2
2005-12-05

Neil W. Van Dyke
neil@neilvandyke.org

<http://www.neilvandyke.org/rfc3339-scm/>

Copyright © 2005 Neil W. Van Dyke. This program is Free Software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This program 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. See <http://www.gnu.org/copyleft/lesser.html> for details. For other license options and consulting, contact the author.

1 Introduction

The `rfc3339.scm` package implements parsing, formatting, and simple validation of **RFC3339** date and time format, which is a subset of **ISO 8601**, intended for use in Internet protocols.

Note that full Scheme support of ISO 8601 is a very different project of the author, and not at all the intention of `rfc3339.scm`.

`rfc3339.scm` requires R5RS, SRFI-6, SRFI-9, and two particular regular expression functions. Note that the regular expression functions in **Pregexp 1e9** will not work, but are expected to work in subsequent versions of Pregexp. Thus far, `rfc3339.scm` has only been tested under PLT MzScheme.

2 Record Type

`rfc3339-record` is an abstract data type for the information in an RFC3339 format time and date. (“`rfc3339-string`” is used in identifiers to denote the RFC3339 format as a Scheme string.)

`make-rfc3339-record` *year month mday hour minute second secfrac* [Procedure]
offsetmin

Construct an `rfc3339-record` with the given field values. Each of *year*, *month*, *mday*, *hour*, *minute*, and *second* is `#f` or a nonnegative integer. *secfrac* is `#f` or a real number that is greater than or equal to 0.0 and less than 1.0. *offsetmin* is `#f` or a nonnegative integer. Note that *offsetmin* represents both the hour and minute components of an RFC3339 string.

`rfc3339-record? x` [Procedure]
Predicate for `rfc3339-record`.

`rfc3339-record:year rec` [Procedure]
`rfc3339-record:month rec` [Procedure]
`rfc3339-record:mday rec` [Procedure]
`rfc3339-record:hour rec` [Procedure]
`rfc3339-record:minute rec` [Procedure]
`rfc3339-record:second rec` [Procedure]
`rfc3339-record:secfrac rec` [Procedure]
`rfc3339-record:offsetmin rec` [Procedure]
Get the respective field value of `rfc3339-record` *rec*.

`rfc3339-record:set-year! rec val` [Procedure]
`rfc3339-record:set-month! rec val` [Procedure]
`rfc3339-record:set-mday! rec val` [Procedure]
`rfc3339-record:set-hour! rec val` [Procedure]
`rfc3339-record:set-minute! rec val` [Procedure]
`rfc3339-record:set-second! rec val` [Procedure]
`rfc3339-record:set-secfrac! rec val` [Procedure]
`rfc3339-record:set-offsetmin! rec val` [Procedure]
Set the respective field value of `rfc3339-record` *rec* to *val*.

`rfc3339-record->list rec` [Procedure]
Yields a list of the `rfc3339-record` *rec* fields, corresponding to the arguments of the `make-rfc3339-record` procedure.
`(rfc3339-record->list`
 `(make-rfc3339-record 1985 4 12 23 20 50 0.52 0))`
`⇒ (1985 4 12 23 20 50 0.52 0)`

3 Parsing

The parsing procedures are for constructing a `rfc3339-records`, lists, and vectors from RFC3339 strings. The underlying parser can also apply a user-supplied closure directly.

`parse-rfc3339-string str constructor` [Procedure]
Parses RFC3339 string *str* and applies procedure *constructor* with the parsed values. The arguments of *constructor* are the same as those of `make-rfc3339-record`.

`string->rfc3339-record str` [Procedure]
Yields an `rfc3339-record` from RFC3339 string *str*.

`rfc3339-string->list str` [Procedure]
`rfc3339-string->vector str` [Procedure]
Yields a list or vector (respectively) from the parsed values of RFC3339 string *str*. The list and vector elements correspond to the arguments of `make-rfc3339-record`.

```
(rfc3339-string->list "1985-04-12T23:20:69.52+5:0")
⇒ (1985 4 12 23 20 69 0.52 300)
(rfc3339-string->vector "1985-04-12T23:20:69.52+5:0")
⇒ #(1985 4 12 23 20 69 0.52 300)
```

4 Formatting

An RFC3339 string format can be obtained from an `rfc3339-record`.

`write-rfc3339 rec port` [Procedure]
Write an RFC3339 string format of `rfc3339-record rec` to output port `port`.

`rfc3339-record->string rec` [Procedure]
Yield an RFC3339 string format of `rfc3339-record rec` as a Scheme string.

5 Validation

A few procedures are provided for validating `rfc3339-records`.

`check-rfc3339-record-date rec explain?` [Procedure]

`check-rfc3339-record-time rec explain?` [Procedure]

`check-rfc3339-record-offset rec explain?` [Procedure]

Check the respective component of `rfc3339-record rec` for completeness and correctness, yielding `#f` iff no problems were detected. If `explain?` is true, then true values of these procedures are lists that “explain” the error detected. For example:

```
(check-rfc3339-record-date
 (string->rfc3339-record "1999-02") #t)
⇒ (missing mday)
```

```
(check-rfc3339-record-date
 (string->rfc3339-record "1999-02-29") #t)
⇒
(invalid mday 29 (and (integer? mday)
 (<= 1 mday (month-days year month))))
```

```
(check-rfc3339-record-date
 (string->rfc3339-record "2000-02-29") #t)
⇒ #f
```

Leap years are calculated correctly. Leap seconds (61st seconds in minutes) are tolerated in any date and time.

`check-rfc3339-record-full rec explain?` [Procedure]
Checks all three components. See `check-rfc3339-record-date` et al.

`valid-full-rfc3339-record? rec` [Procedure]
Yields a true value iff `check-rfc3339-record-full` yields a false value.

6 SRFI-19 Interoperability

`rfc3339.scm` has no dependency on SRFI-19, but a procedure is provided for constructing a SRFI-19 date.

`rfc3339-string->srfi19-date/constructor` *str* *make-date* [Procedure]

Construct a SRFI-19 date from an RFC3339 string, where *str* is the string, and *make-date* is the SRFI-19 date constructor. Applications using SRFI-19 may wish to define an `rfc3339-string->date` procedure:

```
(define (rfc3339-string->date str)
  (rfc3339-string->srfi19-date/constructor str make-date))
```

7 Tests

The `rfc3339.scm` test suite can be enabled by editing the source code file and loading [Testeez](#).

History

Version 0.2 — 2005-12-05

Release for PLT 299/3xx. Changed portability note in light of Pregexp post-1e9 bug fix. Minor documentation changes.

Version 0.1 — 2005-01-30

Initial release.