

Internet Engineering Task Force (IETF)  
Request for Comments: 7529  
Updates: 5545, 6321, 7265  
Category: Standards Track  
ISSN: 2070-1721

C. Daboo  
Apple Inc.  
G. Yakushev  
Google Inc.  
May 2015

Non-Gregorian Recurrence Rules in the Internet Calendaring and  
Scheduling Core Object Specification (iCalendar)

Abstract

This document defines extensions to the Internet Calendaring and Scheduling Core Object Specification (iCalendar) (RFC 5545) to support use of non-Gregorian recurrence rules. It also defines how Calendaring Extensions to WebDAV (CalDAV) (RFC 4791) servers and clients can be extended to support these new recurrence rules.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7529>.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1. Introduction . . . . . 2
- 2. Conventions Used in This Document . . . . . 3
- 3. Overview . . . . . 4
- 4. Extended RRULE Property . . . . . 6
  - 4.1. Skipping Invalid Dates . . . . . 6
  - 4.2. Handling Leap Months . . . . . 9
  - 4.3. Examples . . . . . 10
- 5. Registering Calendar Systems . . . . . 13
- 6. Compatibility . . . . . 13
- 7. Use with iTIP . . . . . 14
- 8. Use with xCal . . . . . 15
- 9. Use with jCal . . . . . 15
- 10. Use with CalDAV . . . . . 16
  - 10.1. CALDAV:supported-rscale-set Property . . . . . 17
- 11. Security Considerations . . . . . 18
- 12. References . . . . . 18
  - 12.1. Normative References . . . . . 18
  - 12.2. Informative References . . . . . 19
- Appendix A. xCal RELAX NG Schema Update . . . . . 20
- Acknowledgments . . . . . 21
- Authors' Addresses . . . . . 21

1. Introduction

The iCalendar [RFC5545] data format is in widespread use to represent calendar data. iCalendar represents dates and times using the Gregorian calendar system only. It does provide a way to use non-Gregorian calendar systems via a "CALSCALE" property, but this has never been used. However, there is a need to support at least non-Gregorian recurrence patterns to cover anniversaries, and many local, religious, or civil holidays based on non-Gregorian dates.

There are several disadvantages to using the existing "CALSCALE" property in iCalendar for implementing non-Gregorian calendars:

- 1. The "CALSCALE" property exists in the top-level "VCALENDAR" objects and thus applies to all components within that object. In today's multi-cultural society, that restricts the ability to mix events from different calendar systems within the same iCalendar object, e.g., it would prevent having both the Gregorian New Year and Chinese New Year in the same iCalendar object.
- 2. Time zone and daylight saving time rules are typically published using Gregorian calendar dates and rules (e.g., "the 3rd Sunday in March") and are thus converted to iCalendar "VTIMEZONE"

components using Gregorian dates and recurrence rules. This results in the problem whereby one component (the "VTIMEZONE") is fixed to the Gregorian calendar system, and another (a "VEVENT") wants to use a different non-Gregorian calendar scale; thus, the single top-level "CALSCALE" property is again inadequate.

This specification solves these issues by allowing the "CALSCALE" to remain set to Gregorian but redefining the "RRULE" recurrence rule property to accept new items, including one that allows non-Gregorian calendar systems to be used. With this, all the "DATE", "DATE-TIME", and "PERIOD" values in the iCalendar object would remain specified using the Gregorian calendar system, but repeating patterns in other calendar systems could be defined. It is then up to calendar user agents and servers to map between Gregorian and non-Gregorian calendar systems in order to expand out recurrence instances. The non-Gregorian recurrence rules can be used in any iCalendar component that allows the "RRULE" property to be specified, including "VTIMEZONE" components (to allow for possible future use of non-Gregorian rules in published daylight saving time data).

This specification does not itself define calendar systems; rather, it utilizes the calendar system registry defined by the Unicode Consortium in their Common Locale Data Repository (CLDR) project [UNICODE.CLDR], as implemented in the Unicode (International Components for Unicode (ICU)) Library [UNICODE.ICU].

This specification makes the following updates:

It updates iCalendar [RFC5545], the XML format for iCalendar (xCal) [RFC6321], and the JSON format for iCalendar (jCal) [RFC7265], to extend the "RRULE" property definition.

It clarifies use of the iCalendar Transport-Independent Interoperability Protocol (iTIP) [RFC5546] to specify how the extended "RRULE" property should be handled in iTIP messages.

It extends CalDAV [RFC4791] to specify how the extended "RRULE" property can be supported by CalDAV servers and clients.

## 2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The notation used in this memo is the ABNF notation of [RFC5234] as used by iCalendar [RFC5545]. Any syntax elements shown below that are not explicitly defined in this specification come from iCalendar [RFC5545], iTIP [RFC5546], and CalDAV [RFC4791].

When XML element types in the namespaces "DAV:" and "urn:ietf:params:xml:ns:caldav" are referenced in this document outside of the context of an XML fragment, the string "DAV:" and "CALDAV:" will be prefixed to the element type names, respectively.

When a Gregorian calendar date is shown in text, it will use the format "YYYYMMDD", where "YYYY" is the 4-digit year, "MM" the 2-digit month, and "DD" the 2-digit day (this is the same format used in iCalendar [RFC5545]). The Chinese calendar will be used as an example of a non-Gregorian calendar for illustrative purposes. When a Chinese calendar date is shown in text, it will use the format "{C}YYYYMM[L]DD" -- i.e., the same format as Gregorian but with a "{C}" prefix, and an optional "L" character after the month element to indicate a leap month. Similarly, {E} and {H} are used in other examples as prefixes for Ethiopic (Amete Mihret) and Hebrew dates, respectively. The "{}" prefix is used for purely illustrative purposes and never appears in actual dates used in iCalendar or related specifications. Note that the Chinese calendar years shown in the examples are based on the Unicode (ICU) [UNICODE.ICU] library's Chinese calendar epoch. While there are several different Chinese calendar epochs in common use, the choice of one over another does not impact the actual calculation of the Gregorian equivalent dates, provided conversion is always done using the same epoch.

### 3. Overview

In the Gregorian calendar system, each year is composed of a fixed number of months (12), with each month having a fixed number of days (between 30 and 31), except for the second month (February), which contains either 28 or 29 days (the latter in a leap year). Weeks are composed of 7 days, with day names Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Years can have either 365 or 366 days (the latter in a leap year). The number of whole weeks in a year is 52 (though the [ISO.8601.2004] week numbering scheme used by iCalendar [RFC5545] can have a numeric count up to 53).

In iCalendar, the "RECUR" value type defines various fields used to express a recurrence pattern, and those fields are given limits based on those of the Gregorian calendar system. Since other calendar systems can have different limits and other behaviors that need to be accounted for, the maximum values for the elements in the "RECUR" value are not covered by this specification.

To generate a set of recurring instances in a non-Gregorian calendar system, the following principles are used:

1. iCalendar data continues to use the "GREGORIAN" calendar system, so all "DATE", "DATE-TIME", and "PERIOD" values continue to use the Gregorian format and limits.
2. The "RRULE" property is extended to include an "RSCALE" element in its value that specifies the calendar system to use for the recurrence pattern. The existing elements of the "RRULE" value type are used, but modified to support different upper limits, based on the "RSCALE" value, as well as a modification to month numbers to allow a leap month to be specified. Existing requirements for the use of "RRULE" all still apply (e.g., the "RRULE" has to match the "DTSTART" value of the master instance). Other recurrence properties such as "RECURRENCE-ID", "RDATE", and "EXDATE" continue to use the Gregorian date format as "CALSCALE" is unchanged.

When generating instances, the following procedure might be used:

1. Convert the "DTSTART" property value of the master recurring component into the date and time components for the calendar system specified by the "RSCALE" element in the "RRULE" value. This provides the "seed" value for generating subsequent recurrence instances.
2. Iteratively generate instances using the "RRULE" value applied to the year, month, and day components of the date in the new calendar system.
3. For each generated instance, convert the dates and times back from the non-Gregorian form into Gregorian and use those values for other properties such as "RECURRENCE-ID".

Consider the following example for an event representing the Chinese New Year:

```
DTSTART;VALUE=DATE:20130210
RRULE:RSCALE=CHINESE;FREQ=YEARLY
SUMMARY:Chinese New Year
```

To generate instances, first the "DTSTART" value "20130210" is converted into the Chinese calendar system giving "{C}46500101". Next, the year component is incremented by one to give "{C}46510101", and that is then converted back into Gregorian as "20140131". Additional instances are generated by iteratively increasing the year component in the Chinese date and converting back to Gregorian.

#### 4. Extended RRULE Property

This specification extends the existing "RRULE" iCalendar property value to include a new "RSCALE" element that can be used to indicate the calendar system used for generating the recurrence pattern.

When "RSCALE" is present, the other changes to "RRULE" are:

1. Elements that include numeric values (e.g., "BYYEARDAY") have numeric ranges defined by the "RSCALE" value (i.e., in some calendar systems there might be more than 366 days in a year).
2. Month numbers can include an "L" suffix to indicate that the specified month is a leap month in the corresponding calendar system (see Section 4.2).
3. A "SKIP" element is added to define how "missing" instances are handled (see Section 4.1).

The syntax for the "RECUR" value is modified in the following fashion:

```

; recur-rule-part is extended from RFC 5545
recur-rule-part =/ ("RSCALE" "=" rscale)
                  / ("SKIP" "=" skip)

rscale           = (iana-token ; A CLDR-registered calendar system
                   ; name.
                   / x-name) ; A non-standard, experimental
                               ; calendar system name.
                               ; Names are case insensitive,
                               ; but uppercase values are preferred.

skip             = ("OMIT" / "BACKWARD" / "FORWARD")
                   ; Optional, with default value "OMIT", and
                   ; MUST NOT be present unless "RSCALE" is present.

monthnum        = 1*2DIGIT ["L"]
                   ; Existing element modified to include a leap
                   ; month indicator suffix.

```

##### 4.1. Skipping Invalid Dates

In every calendar system, only certain combinations of day-of-month and month values are valid for a given year, e.g., in the Gregorian calendar system, January 31st is valid but February 31st is not. Similarly, February 29th is valid in a leap year but invalid in a non-leap year. Other calendar systems can also include leap months

(see Section 4.2), which vary from year to year. This poses a problem for recurring events where the frequency of recurrence might give rise to an invalid date. For example, a recurring event that starts on January 31st and is set to repeat monthly will generate invalid dates for months with fewer than 31 days. The iCalendar [RFC5545] specification requires recurrence rule generators to ignore any invalid dates generated when iterating the rule. However, that behavior might be surprising to a calendar user born on a leap day and whose birthday event only appears on their calendar every four years. There are common conventions used by humans to determine what to do in such cases, but those conventions can differ from calendar system to calendar system, as well as within the same calendar system, depending on the nature of the event. Typically, humans will expect the "missing" events to be moved to an earlier or later (valid) date.

This specification introduces a new "RRULE" element, "SKIP", for use only when the "RSCALE" element is present. The "SKIP" element allows the calendar user agent to specify new options for handling invalid dates.

"SKIP=OMIT": this is the default option and corresponds to the existing iCalendar behavior of simply ignoring the invalid date.

"SKIP=BACKWARD": when this option is set, a date with an invalid month is changed to the previous (valid) month. A date with an invalid day-of-month is changed to the previous (valid) day-of-month.

"SKIP=FORWARD": when this option is set, a date with an invalid month is changed to the next (valid) month. A date with an invalid day-of-month is changed to the next (valid) day-of-month.

Note that for both "BACKWARD" and "FORWARD", if the month is changed and results in an invalid day-of-month, then the skip behavior will be reapplied as per the day-of-month rules, according to the processing order defined below.

The month and day-of-month skip behavior is only applied at specific points during the processing of an "RRULE" as determined by the order in which any "BYxxx" elements are processed. The order is as follows (based on the "RRULE" element processing order defined in Section 3.3.10 of [RFC5545]):

- o BYMONTH
- o SKIP (for invalid month only)

- o BYWEEKNO
- o BYYEARDAY
- o BYMONTHDAY
- o SKIP (for invalid day)
- o BYDAY
- o BYHOUR
- o BYMINUTE
- o BYSECOND
- o BYSETPOS
- o COUNT
- o UNTIL

It is often possible to avoid having to deal with invalid dates by determining the real intent of a human user, e.g., a human creating a monthly recurring event that starts on January 31st likely intends the event to occur on the last day of every month, in which case that could be encoded into an "RRULE" by using the "BYMONTHDAY=-1" element.

Only a few types of recurrence patterns are likely to need the use of "SKIP". The following is a list of some situations where it might be needed:

1. The start date of the recurrence is a leap day in the specified calendar system.
2. The start date of the recurrence is in a leap month in the specified calendar system.
3. The start date of the recurrence has a day-of-month value greater than the smallest day-of-month value for any month in any year in the specified calendar system.
4. A "BYMONTHDAY" element in an "RRULE" has a day-of-month value greater than the smallest day-of-month value for any month in any year in the specified calendar system.

5. A "BYMONTH" element in an "RRULE" has a value corresponding to a leap month in the specified calendar system.
6. A combination of a "BYMONTHDAY" element and a "BYMONTH" element in an "RRULE" has a value corresponding to a leap day in the specified calendar system.
7. A "BYYEARDAY" element in an "RRULE" has an absolute value greater than the smallest number of days in any year in the specified calendar system.
8. A "BYWEEKNO" element in an "RRULE" has an absolute value greater than the smallest number of weeks in any year in the specified calendar system.

Examples of using "SKIP" for some common use cases appear in Section 4.3.

#### 4.2. Handling Leap Months

Leap months can occur in different calendar systems. For such calendar systems, the following rules are applied for "identifying" months:

1. Numeric values 1 through N are used to identify regular, non-leap, months (where N is the number of months in a regular, non-leap, year).
2. The suffix "L" is added to the regular month number to indicate a leap month that follows the regular month, e.g., "5L" is a leap month that follows the 5th regular month in the year.

Care has to be taken when mapping the month identifiers used here with those of any underlying calendar system library being used. In particular, the Hebrew calendar system used by Unicode (ICU) [UNICODE.ICU] uses a month number scheme of 1 through 13, with month 6 being the leap month, and in non-leap years, month 6 is skipped. Thus, ICU months 1 through 5 map to iCalendar months 1 through 5, ICU month 6 maps to iCalendar month "5L", and ICU months 7 through 13 map to iCalendar months 6 through 12.

### 4.3. Examples

#### 4.3.1. Chinese New Year

Consider the following set of iCalendar properties (from the example above):

```
DTSTART;VALUE=DATE:20130210
RRULE:RSCALE=CHINESE;FREQ=YEARLY
SUMMARY:Chinese New Year
```

These define a recurring event for the Chinese New Year, with the first instance being the one in Gregorian year 2013.

The Chinese date corresponding to the first instance is "{C}46500101". The table below shows the initial instance and the next four, each of which is determined by adding the appropriate amount to the year component of the Chinese date. Also shown is the conversion back to the Gregorian date:

Chinese Date	Gregorian Date
{C}46500101	20130210 - DTSTART value
{C}46510101	20140131
{C}46520101	20150219
{C}46530101	20160208
{C}46540101	20170128

#### 4.3.2. Ethiopic 13th Month

Consider the following set of iCalendar properties:

```
DTSTART;VALUE=DATE:20130906
RRULE:RSCALE=ETHIOPIC;FREQ=MONTHLY;BYMONTH=13
SUMMARY:First day of 13th month
```

These define a recurring event for the first day of the 13th month, with the first instance being the one in Gregorian year 2013. While there are a number of alternative ways of writing the "RRULE" above to achieve the same pattern of recurring dates, the one above was chosen to illustrate a "BYMONTH" value exceeding the limit of 12, previously described in iCalendar (Section 3.3.10 of [RFC5545]).

The Ethiopic date corresponding to the first instance is "{E}20051301". The table below shows the initial instance and the next four, each of which is determined by adding the appropriate amount to the year component of the Ethiopic date. Also shown is the conversion back to the Gregorian date:

Ethiopic Date	Gregorian Date
{E}20051301	20130906 - DTSTART value
{E}20061301	20140906
{E}20071301	20150906
{E}20081301	20160906
{E}20091301	20170906

Note that in this example, the value of the "BYMONTH" component in the "RRULE" matches the Ethiopic month value and not the Gregorian month.

#### 4.3.3. Hebrew Anniversary Starting in a Leap Month

Consider the following set of iCalendar properties:

```
DTSTART;VALUE=DATE:20140208
RRULE:RSCALE=HEBREW;FREQ=YEARLY;BYMONTH=5L;BYMONTHDAY=8;SKIP=FORWARD
SUMMARY:Anniversary
```

These define a recurring event for the 8th day of the Hebrew month of Adar I (the leap month identified by "5L"), with the first instance being the one in Gregorian year 2014.

The Hebrew date corresponding to the first instance is "{H}577405L08", which is a leap month in year 5774. The table below shows the initial instance and the next four, each of which is determined by adding the appropriate amount to the year component of the Hebrew date, taking into account that only year 5776 is a leap year. Thus, in other years the Hebrew month component is adjusted forward to month 6. Also shown is the conversion back to the Gregorian date:

Hebrew Date	Gregorian Date
{H}577405L08	20140208 - DTSTART value
{H}57750608	20150227
{H}577605L08	20160217
{H}57770608	20170306
{H}57780608	20180223

#### 4.3.4. Gregorian Leap Day with SKIP

Consider the following set of iCalendar properties:

```
DTSTART;VALUE=DATE:20120229
RRULE:FREQ=YEARLY
SUMMARY:Anniversary
```

These define a recurring event for the 29th of February, 2012, in the standard iCalendar calendar scale -- Gregorian. The standard iCalendar behavior is that non-existent dates in a recurrence set are ignored. Thus, the properties above would only generate instances in leap years (2016, 2020, etc.), which is likely not what users expect. The new "RSCALE" option defined by this specification provides the "SKIP" element, which can be used to "fill in" the missing instances in an appropriate fashion. The set of iCalendar properties below does that:

```
DTSTART;VALUE=DATE:20120229
RRULE:RSCALE=GREGORIAN;FREQ=YEARLY;SKIP=FORWARD
SUMMARY:Anniversary
```

With these properties, the "missing" instances in non-leap years now appear on the 1st of March in those years:

Instances (with SKIP=FORWARD)	Instances (without RSCALE)
20120229	20120229 - DTSTART value
20130301	
20140301	
20150301	
20160229	20160229
20170301	

## 5. Registering Calendar Systems

This specification uses the Unicode Consortium's registry of calendar systems [UNICODE.CLDR] to define valid values for the "RSCALE" element of an "RRULE". Note that the underscore character "\_" is never used in CLDR-based calendar system names. New values can be added to this registry following Unicode Consortium rules. It is expected that many implementations of non-Gregorian calendars will use software libraries provided by Unicode (ICU) [UNICODE.ICU], and hence it makes sense to reuse their registry rather than creating a new one. "RSCALE" values are case insensitive, but uppercase is preferred.

CLDR supports the use of "alias" values as alternative names for specific calendar systems. These alias values can be used as "RSCALE" values and are treated the same as the equivalent CLDR calendar system they are an alias for.

When using the CLDR data, calendar agents SHOULD take into account the "deprecated" value and use the alternative "preferred" calendar system. In particular, the "islamicc" calendar system is considered deprecated in favor of the "islamic-civil" calendar system.

## 6. Compatibility

For calendar user agents that do not support the "RSCALE" element, or do not support the calendar system specified by the "RSCALE" element value, the following behaviors are possible when processing iCalendar data:

1. The calendar user agent can reject the entire iCalendar object within which at least one iCalendar component uses the unrecognized "RSCALE" element or element value.
2. The calendar user agent can reject just the iCalendar components containing an unrecognized "RSCALE" element or element value. Note that any overridden components associated with those rejected components MUST also be rejected (i.e., any other components with the same "UID" property value as the one with the unrecognized "RSCALE" element or element value).
3. The calendar user agent can fall back to a non-recurring behavior for the iCalendar component containing the unrecognized "RSCALE" element or element value (effectively ignoring the "RRULE" property). However, any overridden components SHOULD be rejected as they would represent "orphaned" instances that would seem to be out of place.

In general, the best choice for a calendar user agent would be option (2) above, as it would be the least disruptive choice. Note that when processing iTIP [RFC5546] messages, the manner of the rejection is covered as discussed in the next section.

## 7. Use with iTIP

iTIP [RFC5546] defines how iCalendar data can be sent between calendar user agents to schedule calendar components between calendar users. It is often not possible to know the capabilities of a calendar user agent to which an iTIP message is being sent, but iTIP defines fallback behavior in such cases.

For calendar user agents that do not support the "RSCALE" element, the following can occur when iTIP messages containing an "RSCALE" element are received:

The receiving calendar user agent can reject the entire iTIP message and return an iTIP reply with a "REQUEST-STATUS" property set to the "3.1" status code (as per Section 3.6.14 of [RFC5546]).

The receiving calendar user agent can fall back to a non-recurring behavior for the calendar component (effectively ignoring the "RRULE" property) and return an iTIP reply with a "REQUEST-STATUS" property set to the "2.3", "2.5", "2.8", or "2.10" status codes (as per Sections 3.6.4, 3.6.6, 3.6.9, or 3.6.11, respectively, of [RFC5546]).

For calendar user agents that support the "RSCALE" element but do not support the calendar system specified by the "RSCALE" element value, the following can occur:

The iTIP message SHOULD be rejected, returning a "REQUEST-STATUS" property set to the "3.1" status code (as per Section 3.6.14 of [RFC5546]).

If the iTIP message is accepted and the calendar component treated as non-recurring, an iTIP reply with a "REQUEST-STATUS" property set to the "2.8" or "2.10" status codes (as per Sections 3.6.9 or 3.6.11, respectively, of [RFC5546]) SHOULD be returned.

As noted in Section 6, the best choice is to reject the entire iTIP message.

## 8. Use with xCal

xCal [RFC6321] defines how iCalendar data is represented in XML. This specification extends the <recur> XML element in Section 3.6.10 of [RFC6321] in the following manner:

1. A new <rscale> XML element is defined as a child element of <recur>. The content of this element MUST be a string whose value is the "RSCALE" element value of the "RRULE", with case preserved.
2. A new <skip> XML element is defined as a child element of <recur>. The content of this element MUST be a string whose value is the "SKIP" element value of the "RRULE", with case preserved.
3. The <bymonth> XML element is redefined to support either numeric or string values as its content (as per Section 4.2).

Extensions to the RELAX NG schema in Appendix A of [RFC6321] are defined in Appendix A of this document.

Example: the iCalendar "RRULE" property:

```
RRULE:RSCALE=GREGORIAN;FREQ=YEARLY;SKIP=FORWARD
```

would be represented in XML as:

```
<rrule>
  <recur>
    <rscale>GREGORIAN</rscale>
    <freq>YEARLY</freq>
    <skip>FORWARD</skip>
  </recur>
</rrule>
```

## 9. Use with jCal

jCal [RFC7265] defines how iCalendar data is represented in JSON. This specification extends the "recur" JSON object defined in Section 3.6.10 of [RFC7265] in the following manner:

1. A new "rscale" child member is defined. This MUST be a string whose value is the "RSCALE" element value of the "RRULE", with case preserved.

2. A new "skip" child member is defined. This MUST be a string whose value is the "SKIP" element value of the "RRULE", with case preserved.
3. The "bymonth" child member is redefined to support either numeric or string values. If the "BYMONTH" element value is an integer, then a numeric JSON value MUST be used. If the "BYMONTH" element value is an integer with the "L" suffix (as per Section 4.2), then a JSON string value MUST be used.

Example: the iCalendar "RRULE" property:

```
RRULE:RSCALE=GREGORIAN;FREQ=YEARLY;SKIP=FORWARD
```

would be represented in JSON as:

```
[
  "rrule",
  {},
  "recur",
  {
    "rscale": "GREGORIAN",
    "freq": "YEARLY",
    "skip": "FORWARD"
  }
]
```

## 10. Use with CalDAV

The CalDAV [RFC4791] calendar access protocol allows clients and servers to exchange iCalendar data. In addition, CalDAV clients are able to query calendar data stored on the server, including time-based queries. Since an "RSCALE" element value determines the time ranges for recurring instances in a calendar component, CalDAV servers need to support it to interoperate with clients also using the "RSCALE" element.

A CalDAV server advertises a CALDAV:supported-rscale-set Web Distributed Authoring and Versioning (WebDAV) property on calendar home or calendar collections if it supports use of the "RSCALE" element as described in this specification. The server can advertise a specific set of supported calendar systems by including one or more CALDAV:supported-rscale XML elements within the CALDAV:supported-rscale-set XML element. If no CALDAV:supported-rscale XML elements are included in the WebDAV property, then clients can try any calendar system value but need to be prepared for a failure when attempting to store the calendar data.

Clients MUST NOT attempt to store iCalendar data containing "RSCALE" elements if the CALDAV:supported-rscale-set WebDAV property is not advertised by the server.

The server SHOULD return an HTTP 403 response with a DAV:error element containing a CALDAV:supported-rscale XML element, if a client attempts to store iCalendar data with an "RSCALE" element value not supported by the server.

It is possible for an "RSCALE" value to be present in calendar data on the server being accessed by a client that does not support an "RSCALE" element or its specified value. It is expected that existing clients, unaware of "RSCALE", will fail gracefully by ignoring the calendar component, while still processing other calendar data on the server (as per option (2) in Section 6).

#### 10.1. CALDAV:supported-rscale-set Property

Name: supported-rscale-set

Namespace: urn:ietf:params:xml:ns:caldav

Purpose: Enumerates the set of supported iCalendar "RSCALE" element values supported by the server.

Protected: This property MUST be protected and SHOULD NOT be returned by a PROPFIND allprop request (as defined in Section 14.2 of [RFC4918]).

Description: See above.

Definition:

```
<!ELEMENT supported-rscale-set (supported-rscale*)>
<!ELEMENT supported-rscale (#PCDATA)>
<!-- PCDATA value: string - case insensitive but
      uppercase preferred -->
```

Example:

```
<C:supported-rscale-set
  xmlns:C="urn:ietf:params:xml:ns:caldav">
  <C:supported-rscale>GREGORIAN</C:supported-rscale>
  <C:supported-rscale>CHINESE</C:supported-rscale>
  <C:supported-rscale>ISLAMIC-CIVIL</C:supported-rscale>
  <C:supported-rscale>HEBREW</C:supported-rscale>
  <C:supported-rscale>ETHIOPIC</C:supported-rscale>
</C:supported-rscale-set>
```

## 11. Security Considerations

This specification does not introduce any additional security concerns beyond those described in [RFC5545], [RFC5546], and [RFC4791].

## 12. References

### 12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault, "Calendar Extensions to WebDAV (CalDAV)", RFC 4791, March 2007, <<http://www.rfc-editor.org/info/rfc4791>>.
- [RFC4918] Dusseault, L., Ed., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007, <<http://www.rfc-editor.org/info/rfc4918>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <<http://www.rfc-editor.org/info/rfc5234>>.
- [RFC5545] Desruisseaux, B., Ed., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, September 2009, <<http://www.rfc-editor.org/info/rfc5545>>.
- [RFC5546] Daboo, C., Ed., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", RFC 5546, December 2009, <<http://www.rfc-editor.org/info/rfc5546>>.
- [RFC6321] Daboo, C., Douglass, M., and S. Lees, "xCal: The XML Format for iCalendar", RFC 6321, August 2011, <<http://www.rfc-editor.org/info/rfc6321>>.
- [RFC7265] Kewisch, P., Daboo, C., and M. Douglass, "jCal: The JSON Format for iCalendar", RFC 7265, May 2014, <<http://www.rfc-editor.org/info/rfc7265>>.
- [UNICODE.CLDR] The Unicode Consortium, "CLDR calendar.xml Data", Unicode Consortium CLDR, <<http://www.unicode.org/repos/cldr/tags/latest/common/bcp47/calendar.xml>>.

## 12.2. Informative References

[ISO.8601.2004]

International Organization for Standardization, "Data elements and interchange formats - Information interchange - Representation of dates and times", ISO Standard 8601, December 2004.

[UNICODE.ICU]

"International Components for Unicode", April 2014, <<http://site.icu-project.org>>.

## Appendix A. xCal RELAX NG Schema Update

The following changes are made to the RELAX NG schema defined in Appendix A of [RFC6321].

```
# 3.3.10 RECUR
# This extension adds type-rscale and type-skip,
# and modifies type-bymonth
```

```
value-recur = element recur {
  type-rscale?,
  type-freq,
  (type-until | type-count)?,
  element interval {
    xsd:positiveInteger
  }?,
  type-bysecond*,
  type-byminute*,
  type-byhour*,
  type-byday*,
  type-bymonthday*,
  type-byyearday*,
  type-byweekno*,
  type-bymonth*,
  type-bysetpos*,
  element wkst { type-weekday }?,
  type-skip?
}
```

```
type-rscale = element rscale {
  xsd:string
}
```

```
type-bymonth = element bymonth {
  xsd:positiveInteger |
  xsd:string
}
```

```
type-skip = element skip {
  "OMIT" |
  "BACKWARD" |
  "FORWARD"
}
```

## Acknowledgments

Thanks to the following for feedback: Mark Davis, Mike Douglass, Donald Eastlake, Peter Edberg, Marten Gajda, Philipp Kewisch, Barry Leiba, Jonathan Lennox, Ken Murchison, Arnaud Quillaud, Dave Thewlis, and Umaoka Yoshito.

This specification originated from work at the Calendaring and Scheduling Consortium, which has helped with the development and testing of implementations.

## Authors' Addresses

Cyrus Daboo  
Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
United States

EMail: [cyrus@daboo.name](mailto:cyrus@daboo.name)  
URI: <http://www.apple.com/>

Gregory Yakushev  
Google Inc.  
Brandschenkestrasse 100  
8002 Zurich  
Switzerland

EMail: [gyakushev@yahoo.com](mailto:gyakushev@yahoo.com)  
URI: <http://www.google.com/>

