Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2011

C. Daboo
Apple, Inc.
M. Douglass
RPI
S. Lees
Microsoft

July 13, 2010

xCal: The XML format for iCalendar
draft-daboo-et-al-icalendar-in-xml-05

Abstract

This specification defines "xCal", an XML format for iCalendar data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2011.

Copyright Notice

Copyright (c) 2010 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 3
2. Conventions Used in This Document 3
3. Converting from iCalendar to xCal 3
3.1. Pre-processing 4
3.2. iCalendar stream (RFC5545 section 3.4) 4
3.3. Components (RFC5545 section 3.6) 5
3.4. Properties (RFC5545 section 3.7 and 3.8) 6
3.4.1. Special Cases for Properties 7
3.4.1.1. Multi-valued Properties 8
3.4.1.2. GEO Property 8
3.4.1.3. REQUEST-STATUS Property 8
3.5. Parameters (RFC5545 section 3.2) 8
3.5.1. VALUE parameter 10
3.6. Values (RFC5545 section 3.3) 10
3.6.1. Binary (RFC5545 section 3.3.1) 10
3.6.2. Boolean (RFC5545 section 3.3.2) 10
3.6.3. Calendar User Address (RFC5545 section 3.3.3) 10
3.6.4. Date (RFC5545 section 3.3.4) 11
3.6.5. Date-Time (RFC5545 section 3.3.5) 11
3.6.6. Duration (RFC5545 section 3.3.6) 11
3.6.7. Float (RFC5545 section 3.3.7) 11
3.6.8. Integer (RFC5545 section 3.3.8) 11
3.6.9. Period of Time (RFC5545 section 3.3.9) 11
3.6.10. Recurrence Rule (RFC5545 section 3.3.10) 12
3.6.11. Text (RFC5545 section 3.3.11) 12
3.6.12. Time (RFC5545 section 3.3.12) 12
3.6.13. URI (RFC5545 section 3.3.13) 12
3.6.14. UTC Offset (RFC5545 section 3.3.14) 12
3.7. Extensions 13
4. Converting from XML into iCalendar 13
4.1. Converting XML Extensions into iCalendar 13
4.2. The XML property for iCalendar 13
5. Security Considerations 14
6. IANA Considerations 15
6.1. Namespace Registration 15
6.2. Media Type 15
7. Acknowledgments 16
8. References 16
8.1. Normative References 16
8.2. Informative References 17
Appendix A. Relax NG Schema 17
Appendix B. Example 40
B.1. iCalendar Data 40
B.2. XML Data 41
Appendix C. Change History (to be removed prior to publication as an RFC) 41

1. Introduction

The iCalendar data format [RFC5545] is a widely deployed interchange format for calendaring and scheduling data. While many applications and services consume and generate calendar data, iCalendar is a specialized format that requires its own parser/generator. In contrast, XML-based formats are widely used for interoperability between applications, and the many tools that generate, parse, and manipulate XML make it easier to work with than iCalendar.

The purpose of this specification is to define "xCal", an XML format for iCalendar data. xCal is defined so that iCalendar data to be converted to XML, and then back to iCalendar, without losing any semantic meaning in the data. Anyone creating XML calendar data according to this specification will know that their data can be converted to a valid iCalendar representation as well.

Two key design considerations are:

Round-tripping (converting an iCalendar instance to XML and back) will give the same result as the starting point.

Preserve the semantics of the iCalendar data. While a simple consumer can easily browse the calendar data in XML, a full understanding of iCalendar is still required in order to modify and/or fully comprehend the calendar data.

2. Conventions Used in This Document

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

When XML element types in the namespace "urn:ietf:params:xml:ns:icalendar-2.0" are referenced in this document outside of the context of an XML fragment, the string "ICAL:" will be prefixed to the element types.

Some examples in this document contain "partial" XML documents used for illustrative purposes. In these examples, three periods "..." are used to indicate a portion of the document that has been removed for compactness.

3. Converting from iCalendar to xCal

This section describes how iCalendar data is converted to XML using a simple mapping between the iCalendar data model and XML elements.

3.1. Pre-processing

iCalendar uses a line folding mechanism to limit lines of data to a maximum line length (typically 72 characters) to ensure maximum likelihood of preserving data integrity as it is transported via various means (e.g., email) - see Section 3.1 of [RFC5545]. Prior to converting iCalendar data into XML all folded lines MUST be unfolded.

iCalendar data uses an "escape" character sequence for text values and parameter values. When such text elements are converted into XML the escaping MUST be removed.

iCalendar uses a base64 encoding for binary data. The base64 encoding MUST remain when converted to XML.

3.2. iCalendar stream (RFC5545 section 3.4)

At the top level of the iCalendar object model is an "iCalendar stream". This object encompasses multiple "iCalendar objects". In XML, the entire stream is contained in the root ICAL:icalendar XML element.

An iCalendar stream can contain one or more iCalendar objects. Each iCalendar object, delimited by BEGIN:VCALENDAR and END:VCALENDAR, is enclosed by the ICAL:vcalendar XML element.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
  ...
  </vcalendar>
</icalendar>
```

iCalendar objects are comprised of a set of "components", "properties", "parameters" and "values". A "component" can contain other "components" or "properties". A "property" has a value and optionally a set of "parameters".

In xCal, "components" are contained within an ICAL:components XML element. With that element, another ICAL:components element could appear (representing components nested within components) or the ICAL:properties XML element could appear. ICAL:properties is used to encapsulate iCalendar properties.

Each iCalendar property will be mapped to its own XML element as described below. Within each of these elements there is an optional

ICAL:parameters XML element used to encapsulate any iCalendar parameters. Additionally there will be one or more XML elements representing the value of the iCalendar property.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
    ...
    </properties>
    <components>
    ...
    </components>
  </vcalendar>
</icalendar>
```

Item	XML element	XML Definition
iCalendar Stream	ICAL:icalendar	Appendix A # 3.4
VCALENDAR	ICAL:vcalendar	Appendix A # 3.6

3.3. Components (RFC5545 section 3.6)

Each calendar component in the VCALENDAR object, delimited by BEGIN and END, will be converted to an enclosing XML element with the same name, but in lowercase:

Component	XML element	XML Definition
VEVENT	ICAL:vevent	Appendix A # 3.6.1
VTODO	ICAL:vtodo	Appendix A # 3.6.2
VJOURNAL	ICAL:vjournal	Appendix A # 3.6.3
VFREEBUSY	ICAL:vfreebusy	Appendix A # 3.6.4
VTIMEZONE	ICAL:vtimezone	Appendix A # 3.6.5
STANDARD	ICAL:standard	Appendix A # 3.6.5
DAYLIGHT	ICAL:daylight	Appendix A # 3.6.5
VALARM	ICAL:valarm	Appendix A # 3.6.6

3.4. Properties (RFC5545 section 3.7 and 3.8)

iCalendar properties , whether they apply to the VCALENDAR object or to a component, are handled in a consistent way in the xCal format.

iCalendar properties are enclosed in the XML element ICAL:properties.

Each invidivual iCalendar property is represented in XML by an element of the same name as the iCalendar property, but in lowercase. For example, the CALSCALE property is represented in XML by the ICAL: calscale element.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    <properties>
      <calscale>...</calscale>
      <version>...</version>
      <prodid>...</prodid>
    </properties>
    <components>
    ...
    </components>
  </vcalendar>
</icalendar>
```

Each property can contain an ICAL:parameters XML element encapsulating any iCalendar parameters associated with the iCalendar property.

Each property will contain one or more "value" XML elements as described below representing the value of the iCalendar property.

Property	XML element	XML Definition
CALSCALE	ICAL:calscale	Appendix A # 3.7.1
METHOD	ICAL:method	Appendix A # 3.7.2
PRODID	ICAL:prodid	Appendix A # 3.7.3
VERSION	ICAL:version	Appendix A # 3.7.4
ATTACH	ICAL:attach	Appendix A # 3.8.1.1
CATEGORIES	ICAL:categories	Appendix A # 3.8.1.2
CLASS	ICAL:class	Appendix A # 3.8.1.3
COMMENT	ICAL:comment	Appendix A # 3.8.1.4
DESCRIPTION	ICAL:description	Appendix A # 3.8.1.5
GEO	ICAL:geo	Appendix A # 3.8.1.6

LOCATION	ICAL:location	Appendix A # 3.8.1.7
PERCENT-COMPLETE	ICAL:percent-complete	Appendix A # 3.8.1.8
PRIORITY	ICAL:priority	Appendix A # 3.8.1.9
RESOURCES	ICAL:resources	Appendix A # 3.8.1.10
STATUS	ICAL:status	Appendix A # 3.8.1.11
SUMMARY	ICAL:summary	Appendix A # 3.8.1.12
COMPLETED	ICAL:completed	Appendix A # 3.8.2.1
DTEND	ICAL:dtend	Appendix A # 3.8.2.2
DUE	ICAL:due	Appendix A # 3.8.2.3
DTSTART	ICAL:dtstart	Appendix A # 3.8.2.4
DURATION	ICAL:duration	Appendix A # 3.8.2.5
FREEBUSY	ICAL:freebusy	Appendix A # 3.8.2.6
TRANSP	ICAL:transp	Appendix A # 3.8.2.7
TZID	ICAL:tzid	Appendix A # 3.8.3.1
TZNAME	ICAL:tzname	Appendix A # 3.8.3.2
TZOFFSETFROM	ICAL:tzoffsetfrom	Appendix A # 3.8.3.3
TZOFFSETTO	ICAL:tzoffsetto	Appendix A # 3.8.3.4
TZURL	ICAL:tzurl	Appendix A # 3.8.3.5
ATTENDEE	ICAL:attendee	Appendix A # 3.8.4.1
CONTACT	ICAL:contact	Appendix A # 3.8.4.2
ORGANIZER	ICAL:organizer	Appendix A # 3.8.4.3
RECURRENCE-ID	ICAL:recurrence-id	Appendix A # 3.8.4.4
RELATED-TO	ICAL:related-to	Appendix A # 3.8.4.5
URL	ICAL:url	Appendix A # 3.8.4.6
UID	ICAL:uid	Appendix A # 3.8.4.7
EXDATE	ICAL:exdate	Appendix A # 3.8.5.1
RDATE	ICAL:rdate	Appendix A # 3.8.5.2
RRULE	ICAL:rrule	Appendix A # 3.8.5.3
ACTION	ICAL:action	Appendix A # 3.8.6.1
REPEAT	ICAL:repeat	Appendix A # 3.8.6.2
TRIGGER	ICAL:trigger	Appendix A # 3.8.6.3
CREATED	ICAL:created	Appendix A # 3.8.7.1
DTSTAMP	ICAL:dtstamp	Appendix A # 3.8.7.2
LAST-MODIFIED	ICAL:last-modified	Appendix A # 3.8.7.3
SEQUENCE	ICAL:sequence	Appendix A # 3.8.7.4
REQUEST-STATUS	ICAL:request-status	Appendix A # 3.8.8.3

3.4.1. Special Cases for Properties

Some properties in iCalendar can contain "structured" value data. This includes lists of "standard" value types, as well as values with specific "fields". In xCal, these "structured" values are represented as separate XML elements in various ways for ease of processing using standard XML tools.

3.4.1.1. Multi-valued Properties

The following iCalendar properties can have values that consist of a list of "standard" iCalendar values separated by a specific delimiter. In XML these properties are represented by an XML element that contains multiple "value" elements (Section 3.6).

Property	XML element	XML Definition
CATEGORIES	ICAL:categories	Appendix A # 3.8.1.2
RESOURCES	ICAL:resources	Appendix A # 3.8.1.10
FREEBUSY	ICAL:freebusy	Appendix A # 3.8.2.6
EXDATE	ICAL:exdate	Appendix A # 3.8.5.1
RDATE	ICAL:rdate	Appendix A # 3.8.5.2

3.4.1.2. GEO Property

In iCalendar, the GEO property value is defined as a semi-colon separated list of two FLOAT values, the first representing latitude and the second longitude.

In xCal, the value for the ICAL:geo element is represented by an ICAL:value element containing an ICAL:latitude element and an ICAL: longitude element, each of which contain text values representing the FLOAT values. See Appendix A # 3.8.1.6.

3.4.1.3. REQUEST-STATUS Property

In iCalendar, the REQUEST-STATUS property value is defined as a semi-colon separated list of two or three TEXT values. The first represents a code, the second a description, and the third (optional) additional data.

In xCal, the value for the ICAL:request-status element is represented by an ICAL:value element containing an ICAL:code element, and iCAL: description element, and optionally and ICAL:data element, each of which contain the corresponding TEXT values. See Appendix A # 3.8.8.3.

3.5. Parameters (RFC5545 section 3.2)

iCalendar parameters are enclosed in the XML element ICAL:parameters which optionally occurs once in each property XML element.

Each invidivual iCalendar parameter is represented in xCal by an element of the same name as the iCalendar parameter, but in

lowercase. For example, the PARTSTAT parameter is represented in XML by the ICAL:partstat element.

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    ...
    <components>
    ...
      <attendee>
        <parameters>
          <partstat>NEEDS-ACTION</partstat>
        </parameters>
        ...
    </components>
  </vcalendar>
</icalendar>
```

Each parameter contains either text, or one or more child XML elements representing iCalendar value types.

Parameter	XML element	XML Definition
ALTREP	ICAL:altrep	Appendix A # 3.2.1
CN	ICAL:cn	Appendix A # 3.2.2
CUTYPE	ICAL:cutype	Appendix A # 3.2.3
DELEGATED-FROM	ICAL:delegated-from	Appendix A # 3.2.4
DELEGATED-TO	ICAL:delegated-to	Appendix A # 3.2.5
DIR	ICAL:dir	Appendix A # 3.2.6
ENCODING	ICAL:encoding	Appendix A # 3.2.7
FMTTYPE	ICAL:fmttype	Appendix A # 3.2.8
FBTYPE	ICAL:fbtype	Appendix A # 3.2.9
LANGUAGE	ICAL:language	Appendix A # 3.2.10
MEMBER	ICAL:member	Appendix A # 3.2.11
PARTSTAT	ICAL:partstat	Appendix A # 3.2.12
RANGE	ICAL:range	Appendix A # 3.2.13
RELATED	ICAL:related	Appendix A # 3.2.14
RELTYPE	ICAL:reltype	Appendix A # 3.2.15
ROLE	ICAL:role	Appendix A # 3.2.16
RSVP	ICAL:rsvp	Appendix A # 3.2.17
SENT-BY	ICAL:sent-by	Appendix A # 3.2.18
TZID	ICAL:tzid	Appendix A # 3.2.19

3.5.1. VALUE parameter

iCalendar defines a VALUE parameter (Section 3.2.20 of [RFC5545]. This parameter is not mapped to an xCal XML element. Instead, the value type is handled by having different XML elements for each value, and these appear inside of ICAL:property elements. Thus, when converting from iCalendar to XML, any VALUE parameters are skipped. When converting from XML into iCalendar, the appropriate VALUE parameter MUST be included in the iCalendar property if the value type is not the default value type for that property.

3.6. Values (RFC5545 section 3.3)

iCalendar value types are mapped into XML elements with a matching name in all lowercase. In some cases, iCalendar defines "structured" values and these are mapped into separate child elements in each value element, as described by the simple DTD definitions below.

Some properties allow for multiple values and these are represented by separate matching value XML elements.

3.6.1. Binary (RFC5545 section 3.3.1)

Description: iCalendar BINARY property values are represented by the ICAL:binary XML element. The content of the element is base64 encoded data. Whitespace MAY be inserted into the data at any point to "wrap" the data to reasonable line lengths. When converting back to iCalendar the whitespace MUST first be removed.

XML Definition: Appendix A # 3.3.1

3.6.2. Boolean (RFC5545 section 3.3.2)

Description: iCalendar BOOLEAN property values are represented by the ICAL:boolean XML element. The content of the element is text containing either of "TRUE" or "FALSE".

XML Definition: Appendix A # 3.3.2

3.6.3. Calendar User Address (RFC5545 section 3.3.3)

Description: iCalendar CAL-ADDRESS property values are represented by the ICAL:cal-address XML element. The content of the element is a URI.

XML Definition: Appendix A # 3.3.3

3.6.4. Date (RFC5545 section 3.3.4)

Description: iCalendar DATE property values are represented by the ICAL:date XML element. The content of the element is the same date value specified by RFC5545.

XML Definition: Appendix A # 3.3.4

3.6.5. Date-Time (RFC5545 section 3.3.5)

Description: iCalendar DATE-TIME property values are represented by the ICAL:date-time XML element. The content of the element is the same date-time value specified by RFC5545.

XML Definition: Appendix A # 3.3.5

3.6.6. Duration (RFC5545 section 3.3.6)

Description: iCalendar DURATION property values are represented by the ICAL:duration XML element. The content of the element is the same duration value specified by RFC5545.

XML Definition: Appendix A # 3.3.6

3.6.7. Float (RFC5545 section 3.3.7)

Description: iCalendar FLOAT property values are represented by the ICAL:float XML element. The content of the element is a text representation of a floating point number.

XML Definition: Appendix A # 3.3.7

3.6.8. Integer (RFC5545 section 3.3.8)

Description: iCalendar INTEGER property values are represented by the ICAL:integer XML element. The content of the element is a text representation of an integer number.

XML Definition: Appendix A # 3.3.8

3.6.9. Period of Time (RFC5545 section 3.3.9)

Description: iCalendar PERIOD property values are represented by the ICAL:period XML element. The content of the element is the same period value specified by RFC5545.

XML Definition: Appendix A # 3.3.9

3.6.10. Recurrence Rule (RFC5545 section 3.3.10)

Description: iCalendar RECUR property values are represented by the ICAL:recur XML element. The content of the element is child elements representing the various components of a recurrence rule.

XML Definition: Appendix A # 3.3.10

3.6.11. Text (RFC5545 section 3.3.11)

Description: iCalendar TEXT property values are represented by the ICAL:text XML element. The content of the element is simple text.

XML Definition: Appendix A # 3.3.11

3.6.12. Time (RFC5545 section 3.3.12)

Description: iCalendar TIME property values are represented by the ICAL:time XML element. The content of the element is the same time value specified by RFC5545.

XML Definition: Appendix A # 3.3.12

3.6.13. URI (RFC5545 section 3.3.13)

Description: iCalendar URI property values are represented by the ICAL:uri XML element. The content of the element is a URI.

XML Definition: Appendix A # 3.3.13

3.6.14. UTC Offset (RFC5545 section 3.3.14)

Description: iCalendar UTC-OFFSET property values are represented by the ICAL:utc-offset XML element. The content of the element is the same UTC offset value specified by RFC5545.

XML Definition: Appendix A # 3.3.14

3.7. Extensions

iCalendar extension properties and parameters (those with an "X-" prefix in their name) are handled in the same way as other properties and parameters: the property or parameter is represented by an XML element with the same name, but in lowercase. e.g., the "X-FOO" property in iCalendar turns into the ICAL:x-foo element in XML.

4. Converting from XML into iCalendar

When converting component, property and parameter values, the names SHOULD be converted to uppercase. Although iCalendar names are case insensitive, common practice is to keep them all uppercase following the actual definitions in [RFC5545].

Backslash escaping and line folding MUST be applied to the resulting iCalendar data as required by [RFC5545].

4.1. Converting XML Extensions into iCalendar

XML extensions are converted back to iCalendar in one of two ways, depending on whether the extensions are in the iCalendar XML namespace, or in an external namespace.

Extensions that are part of the iCalendar XML namespace MUST have element names that being with "x-", and will be converted back to the equivalent extension property in iCalendar. For example, the "x-foo" element will convert to the "X-FOO" iCalendar property.

Extensions that are in a namespace other than the iCalendar XML namespace SHOULD be preserved in the iCalendar representation using the XML iCalendar property described in Section 4.2.

4.2. The XML property for iCalendar

This section describes an extension property for iCalendar, as covered in section 8.2.3 of [RFC5545].

Property name: XML

Purpose: To embed XML-encoded calendar data in the iCalendar format.

Value type: A single text value.

Property parameters: VALUE, ENCODING.

Conformance: The property can appear on any iCalendar component.

Description: The value of this property is an XML element. The XML property MUST NOT be used to contain properties that are already defined in iCalendar, or properties that use the "X-" iCalendar extension property syntax. Since all elements in the urn:ietf:params:xml:ns:icalendar-2.0 namespace convert to a well-defined iCalendar object, the elements in this property MUST NOT be in the urn:ietf:params:xml:ns:icalendar-2.0 namespace. The XML element which is the value of this property MUST have an XML namespace declaration.

Note that the source XML may contain characters not allowed in iCalendar such as control characters. If this is the case, then the XML data MUST be base64 encoded. As required by [RFC5545], the ENCODING paramater MUST be present and set to "BASE64", and the VALUE parameter must be present and set to "BINARY".

There can be more than one XML property present for a given iCalendar object. The ordering of XML properties is not preserved in the conversion between XML and iCalendar.

Format definition: This property is defined by the following notation:

```
xml = "XML:" text CRLF
```

Example: The following is an example of an iCalendar event with a location embedded in KML markup inside the XML property.

```
BEGIN:VCALENDAR
CALSCALE:GREGORIAN
PRODID:-//Example Inc.//Example Calendar//EN
VERSION:2.0
BEGIN:VEVENT
DTSTAMP:20080205T191224Z
DTSTART:20081006
SUMMARY:Planning meeting
UID:4088E990AD89CB3DBB484909
XML:<kml xmlns="http://www.opengis.net/kml/2.2">...</kml>
END:VEVENT
END:VCALENDAR
```

5. Security Considerations

This extension does not introduce any new security concerns than those already described in iCalendar.

6. IANA Considerations

This document defines a new URN to identify a new XML namespace for iCalendar data. The URN conforms to a registry mechanism described in [RFC3688].

This document defines a new media type. The registration is in Section 6.2.

This document defines a new property for iCalendar. The registration is in Section 4.2.

6.1. Namespace Registration

Registration request for the iCalendar namespace:

URI: urn:ietf:params:xml:ns:icalendar-2.0

Registrant Contact: See the "Authors' Addresses" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

6.2. Media Type

This section defines the MIME media type for use with iCalendar in XML data.

To: ietf-types@iana.org

Subject: Registration of media type application/xml+calendar

Type name: application

Subtype name: calendar+xml

Required parameters: none

Optional parameters: charset, method, component and optinfo as defined for the text/calendar media type

Encoding considerations: iCalendar data is typically UTF-8 and thus the XML representation will follow that. As a result, for 7-bit transports, data in UTF-8 MUST be encoded in quoted-printable or base64.

Security considerations: See Section 5.

Interoperability considerations: This media type provides an alternative syntax to iCalendar data based on XML.

Published specification: This specification.

Applications which use this media type: Applications that currently make use of the text/calendar media type can use this as an alternative.

Additional information:

Magic number(s): None

File extension(s): XML data should use "xml" as the file extension.

Macintosh file type code(s): None specified.

Person & email address to contact for further information: See the "Author's Address" section of this document.

Intended usage: COMMON

Restrictions on usage: There are no restrictions on where this media type can be used.

Author: See the "Author's Address" section of this document.

Change controller: IETF

7. Acknowledgments

This specification originated from the work of the XML technical committee of the Calendaring and Scheduling Consortium.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC5545] Desruisseaux, B., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, September 2009.

[W3C.REC-xml-20040204] Bray, T., Maler, E., Sperberg-McQueen, C., Paoli, J., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium FirstEdition REC-xml-20040204, February 2004, <http://www.w3.org/TR/2004/REC-xml-20040204>.

8.2. Informative References

Appendix A. Relax NG Schema

Below is a Relax NG schema for iCalendar in XML. This schema uses the compact notation of Relax NG. The numeric section numbers given in the comments refer to section in [RFC5545]. The ordering of elements follows the section ordering of [RFC5545].

The Relax NG compact notation "?" operator is used to indicate an unordered list of items. However, that operator, as defined, allows "mixing" each element that it operates on at any depth within the other elements, rather than just allowing "mixing" of siblings only. As a result, the schema provided allows certain constructs that are not allowed in iCalendar. Given that there is no sibling-only unordered list operator in RelaxNG, this is the best representation that can be given.

```
# Relax NG Schema for iCalendar in XML

default namespace = "urn:ietf:params:xml:ns:icalendar-2.0"

# 3.2 Property Parameters

# 3.2.1 Alternate Text Representation

altrepparam = element altrep { value-uri }

# 3.2.2 Common Name

cnparam = element cn { text }

# 3.2.3 Calendar User Type

cutypeparam = element cutype {
    "INDIVIDUAL" |
    "GROUP" |
```

```
        "RESOURCE" |
        "ROOM" |
        "UNKNOWN"
    }

    # 3.2.4 Delegators

    delfromparam = element delegated-from { value-cal-address+ }

    # 3.2.5 Delegatees

    deltoparam = element delegated-to { value-cal-address+ }

    # 3.2.6 Directory Entry Reference

    dirparam = element dir { value-uri }

    # 3.2.7 Inline Encoding

    encodingparam = element encoding {
        "8BIT" |
        "BASE64"
    }


    # 3.2.8 Format Type

    fmttypeparam = element fmttype { text }

    # 3.2.9 Free/Busy Time Type

    fbtypeparam = element fbtype {
        "FREE" |
        "BUSY" |
        "BUSY-UNAVAILABLE" |
        "BUSY-TENTATIVE"
    }

    # 3.2.10 Language

    languageparam = element language { text }

    # 3.2.11 Group or List Membership

    memberparam = element member { value-cal-address+ }

    # 3.2.12 Participation Status
```

```
partstatparam = element partstat {
    type-partstat-event |
    type-partstat-todo |
    type-partstat-jour
}

type-partstat-event = (
    "NEEDS-ACTION" |
    "ACCEPTED" |
    "DECLINED" |
    "TENTATIVE" |
    "DELEGATED"
)

type-partstat-todo = (
    "NEEDS-ACTION" |
    "ACCEPTED" |
    "DECLINED" |
    "TENTATIVE" |
    "DELEGATED" |
    "COMPLETED" |
    "IN-PROCESS"
)

type-partstat-jour = (
    "NEEDS-ACTION" |
    "ACCEPTED" |
    "DECLINED"
)

# 3.2.13 Recurrence Identifier Range

rangeparam = element range {
    "THISANDFUTURE"
}

# 3.2.14 Alarm Trigger Relationship

trigrelparam = element related {
    "START" |
    "END"
}

# 3.2.15 Relationship Type

reltypeparam = element reltype {
    "PARENT" |
    "CHILD" |
```

```
        "SIBLING"
    }

    # 3.2.16 Participation Role

    roleparam = element role {
        "CHAIR" |
        "REQ-PARTICIPANT" |
        "OPT-PARTICIPANT" |
        "NON-PARTICIPANT"
    }

    # 3.2.17 RSVP Expectation

    rsvpparam = element rsvp {
        "TRUE" |
        "FALSE"
    }

    # 3.2.18 Sent By

    sentbyparam = element sent-by { value-cal-address }

    # 3.2.19 Time Zone Identifier

    tzidparam = element tzid { text }

    # 3.3 Property Value Data Types

    # 3.3.1 BINARY

    value-binary =  element binary { text }

    # 3.3.2 BOOLEAN

    value-boolean = element boolean {
        ("TRUE" | "FALSE")
    }

    # 3.3.3 CAL-ADDRESS

    value-cal-address = element cal-address { text }

    # 3.3.4 DATE

    value-date = element date {
            text
    }
```

```
# 3.3.5 DATE-TIME

value-date-time = element date-time {
        text
}

# 3.3.6 DURATION

value-duration = element duration {
        text
}

# 3.3.7 FLOAT

value-float = element float { text }

# 3.3.8 INTEGER

value-integer = element integer { text }

# 3.3.9 PERIOD

value-period = element period {
        text
}

# 3.3.10 RECUR

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

type-freq = element freq {
    "SECONDLY" |
    "MINUTELY" |
    "HOURLY"   |
```

```
        "DAILY"    |
        "WEEKLY"   |
        "MONTHLY"  |
        "YEARLY"
    }

    type-until = element until {
        type-date |
        type-date-time
    }

    type-count = element count { text }

    type-weekday = (
        "SU" |
        "MO" |
        "TU" |
        "WE" |
        "TH" |
        "FR" |
        "SA"
    )

    type-byday = element byday {
        element ordwk {
            text
        }?,
        element weekday { type-weekday }
    }

    type-bymonthday = element bymonthday {
        text
    }

    type-byyearday = element byyearday {
        text
    }

    type-byweekno = element byweekno {
        text
    }

    type-bysetpos = element bysetpos {
        text
    }

    # 3.3.11 TEXT
```

```
value-text = element text { text }

# 3.3.12 TIME

value-time = element time { text }

# 3.3.13 URI

value-uri = element uri { text }

# 3.3.14 UTC-OFFSET

value-utc-offset = element utc-offset { text }


# 3.4 iCalendar Stream

start = element icalendar { vcalendar+ }

# 3.6 Calendar Components

vcalendar = element vcalendar {
    type-calprops,
    type-component
}

type-calprops = element properties {
    property-prodid &
    property-version &
    property-calscale? &
    property-method?
}

type-component = element components {
    (
        component-vevent |
        component-vtodo |
        component-vjournal |
        component-vfreebusy |
        component-vtimezone
    )*
}

# 3.6.1 Event Component

component-vevent = element vevent {
    type-eventprop,
    element components {
```

```
            component-valarm+
        }?
    }

    type-eventprop = element properties {
        property-dtstamp &
        property-dtstart &
        property-uid &

        property-class? &
        property-created? &
        property-description? &
        property-geo? &
        property-last-mod? &
        property-location? &
        property-organizer? &
        property-priority? &
        property-seq? &
        property-status-event? &
        property-summary? &
        property-transp? &
        property-url? &
        property-recurid? &

        property-rrule? &

        (property-dtend | property-duration)? &

        property-attach* &
        property-attendee* &
        property-categories* &
        property-comment* &
        property-contact* &
        property-exdate* &
        property-rstatus* &
        property-related* &
        property-resources* &
        property-rdate*
    }

    # 3.6.2 To-do Component

    component-vtodo = element vtodo {
        type-todoprop,
        element components {
            component-valarm+
        }?
    }
```

```
type-todoprop = element properties {
    property-dtstamp &
    property-uid &

    property-class? &
    property-completed? &
    property-created? &
    property-description? &
    property-geo? &
    property-last-mod? &
    property-location? &
    property-organizer? &
    property-percent? &
    property-priority? &
    property-recurid? &
    property-seq? &
    property-status-todo? &
    property-summary? &
    property-url? &

    property-rrule? &

    (
        (property-dtstart?, property-dtend? ) |
        (property-dtstart, property-duration)?
    ) &

    property-attach* &
    property-attendee* &
    property-categories* &
    property-comment* &
    property-contact* &
    property-exdate* &
    property-rstatus* &
    property-related* &
    property-resources* &
    property-rdate*
}

# 3.6.3 Journal Component

component-vjournal = element vjournal {
    type-jourprop
}

type-jourprop = element properties {
    property-dtstamp &
    property-uid &
```

```
        property-class? &
        property-created? &
        property-dtstart? &
        property-last-mod? &
        property-organizer? &
        property-recurid? &
        property-seq? &
        property-status-jour? &
        property-summary? &
        property-url? &

        property-rrule? &

        property-attach* &
        property-attendee* &
        property-categories* &
        property-comment* &
        property-contact* &
        property-description? &
        property-exdate* &
        property-related* &
        property-rdate* &
        property-rstatus*
    }

    # 3.6.4 Free/Busy Component

    component-vfreebusy = element vfreebusy {
        type-fbprop
    }

    type-fbprop = element properties {
        property-dtstamp &
        property-uid &

        property-contact? &
        property-dtstart? &
        property-dtend? &
        property-duration? &
        property-organizer? &
        property-url? &

        property-attendee* &
        property-comment* &
        property-freebusy* &
        property-rstatus*
    }
```

```
# 3.6.5 Time Zone Component

component-vtimezone = element vtimezone {
    element properties {
        property-tzid &

        property-last-mod? &
        property-tzuurl?
    },
    element components {
        (component-standard | component-daylight) &
        component-standard* &
        component-daylight*
    }
}

component-standard = element standard {
    type-tzprop
}

component-daylight = element daylight {
    type-tzprop
}

type-tzprop = element properties {
    property-dtstart &
    property-tzoffsetto &
    property-tzoffsetfrom &

    property-rrule? &

    property-comment* &
    property-rdate* &
    property-tzname*
}

# 3.6.6 Alarm Component

component-valarm = element valarm {
    audioprop | dispprop | emailprop
}

type-audioprop = element properties {
    property-action &
    property-trigger &

    (property-duration, property-repeat)? &
```

```
        property-attach?
    }

    type-dispprop = element properties {
        property-action &
        property-description &
        property-trigger &
        property-summary &

        property-attendee+ &

        (property-duration, property-repeat)? &

        property-attach*
    }

    type-emailprop = element properties {
        property-action &
        property-description &
        property-trigger &

        (property-duration, property-repeat)?
    }

    # 3.7 Calendar Properties

    # 3.7.1 Calendar Scale

    property-calscale = element calscale {

        element parameters { empty }?,

        element text { "GREGORIAN" }
    }

    # 3.7.2 Method

    property-method = element method {

        element parameters { empty }?,

        value-text
    }

    # 3.7.3 Product Identifier

    property-prodid = element prodid {
```

```
        element parameters { empty }?,

        value-text
    }

    # 3.7.4 Version

    property-version = element version {

        element parameters { empty }?,

        value-text
    }

    # 3.8 Component Properties

    # 3.8.1 Descriptive Component Properties

    # 3.8.1.1 Attachment

    property-attach = element attach {

        element parameters {
            fmttypeparam? &
            encodingparam?
        }?,

        value-uri | value-binary
    }

    # 3.8.1.2 Categories

    property-categories = element categories {

        element parameters {
            languageparam? &
        }?,

        value-text+
    }

    # 3.8.1.3 Classification

    property-class = element class {

        element parameters { empty }?,

        element text {
```

```
            "PUBLIC" |
            "PRIVATE" |
            "CONFIDENTIAL"
        }
    }

    # 3.8.1.4 Comment

    property-comment = element comment {

        element parameters {
            altrepparam? &
            languageparam?
        }?,

        value-text
    }

    # 3.8.1.5 Description

    property-description = element description {

        element parameters {
            altrepparam? &
            languageparam?
        }?,

        value-text
    }

    # 3.8.1.6 Geographic Position

    property-geo = element geo {

        element parameters { empty }?,

        element value {
            element latitude  { text },
            element longitude { text }
        }
    }

    # 3.8.1.7 Location

    property-location = element location {

        element parameters {
            altrepparam? &
```

```
            languageparam?
        }?,

        value-text
    }

    # 3.8.1.8 Percent Complete

    property-percent = element percent-complete {

        element parameters { empty }?,

        value-integer
    }

    # 3.8.1.9 Priority

    property-priority = element priority {

        element parameters { empty }?,

        value-integer
    }

    # 3.8.1.10 Resources

    property-resources = element resources {

        element parameters {
            altrepparam? &
            languageparam?
        }?,

        value-text+
    }

    # 3.8.1.11 Status

    property-status-event = element status {

        element parameters { empty }?,

        element text {
            "TENTATIVE" |
            "CONFIRMED" |
            "CANCELLED"
        }
    }
```

```
property-status-todo = element status {

    element parameters { empty }?,

    element text {
        "NEEDS-ACTION" |
        "COMPLETED" |
        "IN-PROCESS" |
        "CANCELLED"
    }
}

property-status-jour = element status {

    element parameters { empty }?,

    element text {
        "DRAFT" |
        "FINAL" |
        "CANCELLED"
    }
}

# 3.8.1.12 Summary

property-summary = element summary {

    element parameters {
        altrepparam? &
        languageparam?
    }?,

    value-text
}

# 3.8.2 Date and Time Component Properties

# 3.8.2.1 Date/Time Completed

property-completed = element completed {

    element parameters { empty }?,

    value-date-time
}

# 3.8.2.2 Date/Time End
```

```
property-dtend = element dtend {

    element parameters {
        tzidparam?
    }?,

    value-date-time |
    value-date
}

# 3.8.2.3 Date/Time Due

property-due = element due {

    element parameters {
        tzidparam?
    }?,

    value-date-time |
    value-date
}

# 3.8.2.4 Date/Time Start

property-dtstart = element dtstart {

    element parameters {
        tzidparam?
    }?,

    value-date-time |
    value-date
}

# 3.8.2.5 Duration

property-duration = element duration {

    element parameters { empty }?,

    value-duration
}

# 3.8.2.6 Free/Busy Time

property-freebusy = element freebusy {

    element parameters {
```

```
            fbtypeparam?
        }?,


        value-period+
    }

    # 3.8.2.7 Time Transparency

    property-transp = element transp {

        element parameters { empty }?,

        element text {
            "OPAQUE" |
            "TRANSPARENT"
        }
    }

    # 3.8.3 Time Zone Component Properties

    # 3.8.3.1 Time Zone Identifier

    property-tzid = element tzid {

        element parameters { empty }?,

        value-text
    }

    # 3.8.3.2 Time Zone Name

    property-tzname = element tzname {

        element parameters {
            languageparam?
        }?,

        value-text
    }

    # 3.8.3.3 Time Zone Offset From

    property-tzoffsetfrom = element tzoffsetfrom {

        element parameters { empty }?,

        value-utc-offset
```

```
}

# 3.8.3.4 Time Zone Offset To

property-tzoffsetto = element tzoffsetto {

    element parameters { empty }?,

    value-utc-offset
}

# 3.8.3.5 Time Zone URL

property-tzurl = element tzurl {

    element parameters { empty }?,

    value-uri
}

# 3.8.4 Relationship Component Properties

# 3.8.4.1 Attendee

property-attendee = element attendee {

    element parameters {
        cutypeparam? &
        memberparam? &
        roleparam? &
        partstatparam? &
        rsvpparam? &
        deltoparam? &
        delfromparam? &
        sentbyparam? &
        cnparam? &
        dirparam? &
        languageparam?
    }?,

    value-cal-address
}

# 3.8.4.2 Contact

property-contact = element contact {

    element parameters {
```

```
            altrepparam? &
            languageparam?
        }?,

        value-text
    }

    # 3.8.4.3 Organizer

    property-organizer = element organizer {

        element parameters {
            cnparam? &
            dirparam? &
            sentbyparam? &
            languageparam?
        }?,

        value-cal-address
    }

    # 3.8.4.4 Recurrence ID

    property-recurid = element recurrence-id {

        element parameters {
            tzidparam? &
            rangeparam?
        }?,

        value-date-time |
        value-date
    }

    # 3.8.4.5 Related-To

    property-related = element related-to {

        element parameters {
            reltypeparam?
        }?,

        value-text
    }

    # 3.8.4.6 Uniform Resource Locator

    property-url = element url {
```

```
        element parameters { empty }?,

        value-uri
    }

    # 3.8.4.7 Unique Identifier

    property-uid = element uid {

        element parameters { empty }?,

        value-text
    }

    # 3.8.5 Recurrence Component Properties

    # 3.8.5.1 Exception Date/Times

    property-exdate = element exdate {

        element parameters {
            tzidparam?
        }?,

        value-date-time+ |
        value-date+
    }

    # 3.8.5.2 Recurrence Date/Times

    property-rdate = element rdate {

        element parameters {
            tzidparam?
        }?,

        value-date-time+ |
        value-date+ |
        value-period+
    }

    # 3.8.5.3 Recurrence Rule

    property-rrule = element rrule {

        element parameters { empty }?,

        value-recur
```

```
}

# 3.8.6 Alarm Component Properties

# 3.8.6.1 Action

property-action = element action {

    element parameters { empty }?,

    element text {
        "AUDIO" |
        "DISPLAY" |
        "EMAIL"
    }
}

# 3.8.6.2 Repeat Count

property-repeat = element repeat {

    element parameters { empty }?,

    value-integer
}

# 3.8.6.3 Trigger

property-repeat = element repeat {

    (
        element parameters {
            trigrelparam?
        }?,

        value-duration
    ) |
    (
        element parameters { empty }?,

        value-date-time
    )
}

# 3.8.7 Change Management Component Properties

# 3.8.7.1 Date/Time Created
```

```
property-created = element created {

    element parameters { empty }?,

    value-date-time
}

# 3.8.7.2 Date/Time Stamp

property-dtstamp = element dtstamp {

    element parameters { empty }?,

    value-date-time
}

# 3.8.7.3 Last Modified

property-last-mod = element last-modified {

    element parameters { empty }?,

    value-date-time
}

# 3.8.7.4 Sequence Number

property-seq = element sequence {

    element parameters { empty }?,

    value-integer
}

# 3.8.8 Miscellaneous Component Properties

# 3.8.8.3 Request Status

property-rstatus = element request-status {

    element parameters {
        languageparam?
    }?,

    element value {
        element code { text },
        element description { text },
        element data { text }?
```

```
        }
    }
```

Appendix B. Example

Below is some example iCalendar data and its representation in XML as defined by this specification.

B.1. iCalendar Data

```
BEGIN:VCALENDAR
CALSCALE:GREGORIAN
PRODID:-//Example Inc.//Example Calendar//EN
VERSION:2.0
BEGIN:VEVENT
DTSTAMP:20080205T191224Z
DTSTART:20081006
SUMMARY:Planning meeting
UID:4088E990AD89CB3DBB484909
END:VEVENT
END:VCALENDAR
```

B.2. XML Data

```
<?xml version="1.0" encoding="utf-8"?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
 <vcalendar>
  <properties>
   <calscale><text>GREGORIAN</text></calscale>
   <prodid>
    <text>-//Example Inc.//Example Calendar//EN</text>
   </prodid>
   <version><text>2.0</text></version>
  </properties>
  <components>
   <vevent>
    <properties>
     <dtstamp><date-time>20080205T191224Z</date-time></dtstamp>
     <dtstart><date>20081006</date></dtstart>
     <summary>
      <text>Planning meeting</text>
     </summary>
     <uid>
      <text>4088E990AD89CB3DBB484909</text>
     </uid>
    </properties>
   </vevent>
  </components>
 </vcalendar>
</icalendar>
```

Appendix C. Change History (to be removed prior to publication as an RFC)

Changes from -04:

1. Added description to XML extension discussing how to handle binary data in XML.

2. Removed empty Appendix B.

Changes from -03:

1. Changed the proposed MIME type from xml+calendar to calendar+xml.

2. Fixed several references to sections of RFC5545.

3. Updated example in Appendix C.

4. Corrected the definition and grammar for TIME and UTC-OFFSET properties.

Changes from -02:

1. Removed the LINK extension and related sections. The concept will be addressed in a separate specification.

2. Various minor edits for clarity and consistency.

Changes from -01:

1. Added LINK extension to iCalendar and section discussing links in XML format.

2. Adopted "xCal" as the short name for the specification.

Changes from -00:

1. Changed 2445bis references to RFC5545.

2. Added a version number to the XML namespace for iCalendar.

3. Changed the values for the date, date-time, period, and duration elements to exactly match the values specified in RFC5545. Previously these were broken out into separate elements for day, month, year, etc.

4. Added specification for XML property in iCalendar.

Authors' Addresses

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

EMail: cyrus@daboo.name
URI: http://www.apple.com/

Mike Douglass
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
USA

EMail: douglm@rpi.edu
URI: http://www.rpi.edu/

Steven Lees
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA

EMail: steven.lees@microsoft.com
URI: http://www.microsoft.com/