

Network Working Group  
Request for Comments: 5545  
Obsoletes: 2445  
Category: Standards Track

B. Desruisseaux, Ed.  
Oracle  
September 2009

## **Internet Calendaring and Scheduling Core Object Specification (iCalendar)**

### **Abstract**

This document defines the iCalendar data format for representing and exchanging calendaring and scheduling information such as events, to-dos, journal entries, and free/busy information, independent of any particular calendar service or protocol.

### **Status of This Memo**

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### **Copyright and License Notice**

Copyright (c) 2009 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 BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format

it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1. Introduction . . . . .	5
2. Basic Grammar and Conventions . . . . .	6
2.1. Formatting Conventions . . . . .	6
2.2. Related Memos . . . . .	7
3. iCalendar Object Specification . . . . .	8
3.1. Content Lines . . . . .	8
3.1.1. List and Field Separators . . . . .	11
3.1.2. Multiple Values . . . . .	11
3.1.3. Binary Content . . . . .	11
3.1.4. Character Set . . . . .	12
3.2. Property Parameters . . . . .	12
3.2.1. Alternate Text Representation . . . . .	13
3.2.2. Common Name . . . . .	15
3.2.3. Calendar User Type . . . . .	15
3.2.4. Delegators . . . . .	16
3.2.5. Delegates . . . . .	16
3.2.6. Directory Entry Reference . . . . .	17
3.2.7. Inline Encoding . . . . .	17
3.2.8. Format Type . . . . .	18
3.2.9. Free/Busy Time Type . . . . .	19
3.2.10. Language . . . . .	20
3.2.11. Group or List Membership . . . . .	20
3.2.12. Participation Status . . . . .	21
3.2.13. Recurrence Identifier Range . . . . .	22
3.2.14. Alarm Trigger Relationship . . . . .	23
3.2.15. Relationship Type . . . . .	24
3.2.16. Participation Role . . . . .	25
3.2.17. RSVP Expectation . . . . .	25
3.2.18. Sent By . . . . .	26
3.2.19. Time Zone Identifier . . . . .	26
3.2.20. Value Data Types . . . . .	28
3.3. Property Value Data Types . . . . .	29
3.3.1. Binary . . . . .	29
3.3.2. Boolean . . . . .	30
3.3.3. Calendar User Address . . . . .	30
3.3.4. Date . . . . .	31
3.3.5. Date-Time . . . . .	31
3.3.6. Duration . . . . .	34
3.3.7. Float . . . . .	35
3.3.8. Integer . . . . .	35
3.3.9. Period of Time . . . . .	36
3.3.10. Recurrence Rule . . . . .	37
3.3.11. Text . . . . .	45

3.3.12. Time . . . . .	46
3.3.13. URI . . . . .	48
3.3.14. UTC Offset . . . . .	49
3.4. iCalendar Object . . . . .	49
3.5. Property . . . . .	50
3.6. Calendar Components . . . . .	50
3.6.1. Event Component . . . . .	52
3.6.2. To-Do Component . . . . .	56
3.6.3. Journal Component . . . . .	58
3.6.4. Free/Busy Component . . . . .	60
3.6.5. Time Zone Component . . . . .	63
3.6.6. Alarm Component . . . . .	72
3.7. Calendar Properties . . . . .	77
3.7.1. Calendar Scale . . . . .	77
3.7.2. Method . . . . .	78
3.7.3. Product Identifier . . . . .	79
3.7.4. Version . . . . .	80
3.8. Component Properties . . . . .	81
3.8.1. Descriptive Component Properties . . . . .	81
3.8.1.1. Attachment . . . . .	81
3.8.1.2. Categories . . . . .	82
3.8.1.3. Classification . . . . .	83
3.8.1.4. Comment . . . . .	84
3.8.1.5. Description . . . . .	85
3.8.1.6. Geographic Position . . . . .	87
3.8.1.7. Location . . . . .	88
3.8.1.8. Percent Complete . . . . .	89
3.8.1.9. Priority . . . . .	90
3.8.1.10. Resources . . . . .	92
3.8.1.11. Status . . . . .	93
3.8.1.12. Summary . . . . .	94
3.8.2. Date and Time Component Properties . . . . .	95
3.8.2.1. Date-Time Completed . . . . .	95
3.8.2.2. Date-Time End . . . . .	96
3.8.2.3. Date-Time Due . . . . .	97
3.8.2.4. Date-Time Start . . . . .	99
3.8.2.5. Duration . . . . .	100
3.8.2.6. Free/Busy Time . . . . .	101
3.8.2.7. Time Transparency . . . . .	102
3.8.3. Time Zone Component Properties . . . . .	103
3.8.3.1. Time Zone Identifier . . . . .	103
3.8.3.2. Time Zone Name . . . . .	105
3.8.3.3. Time Zone Offset From . . . . .	106
3.8.3.4. Time Zone Offset To . . . . .	106
3.8.3.5. Time Zone URL . . . . .	107
3.8.4. Relationship Component Properties . . . . .	108
3.8.4.1. Attendee . . . . .	108
3.8.4.2. Contact . . . . .	111

3.8.4.3.	Organizer . . . . .	113
3.8.4.4.	Recurrence ID . . . . .	114
3.8.4.5.	Related To . . . . .	117
3.8.4.6.	Uniform Resource Locator . . . . .	118
3.8.4.7.	Unique Identifier . . . . .	119
3.8.5.	Recurrence Component Properties . . . . .	120
3.8.5.1.	Exception Date-Times . . . . .	120
3.8.5.2.	Recurrence Date-Times . . . . .	122
3.8.5.3.	Recurrence Rule . . . . .	124
3.8.6.	Alarm Component Properties . . . . .	134
3.8.6.1.	Action . . . . .	134
3.8.6.2.	Repeat Count . . . . .	135
3.8.6.3.	Trigger . . . . .	135
3.8.7.	Change Management Component Properties . . . . .	138
3.8.7.1.	Date-Time Created . . . . .	138
3.8.7.2.	Date-Time Stamp . . . . .	139
3.8.7.3.	Last Modified . . . . .	140
3.8.7.4.	Sequence Number . . . . .	141
3.8.8.	Miscellaneous Component Properties . . . . .	142
3.8.8.1.	IANA Properties . . . . .	142
3.8.8.2.	Non-Standard Properties . . . . .	142
3.8.8.3.	Request Status . . . . .	144
4.	iCalendar Object Examples . . . . .	146
5.	Recommended Practices . . . . .	150
6.	Internationalization Considerations . . . . .	151
7.	Security Considerations . . . . .	151
8.	IANA Considerations . . . . .	151
8.1.	iCalendar Media Type Registration . . . . .	151
8.2.	New iCalendar Elements Registration . . . . .	155
8.2.1.	iCalendar Elements Registration Procedure . . . . .	155
8.2.2.	Registration Template for Components . . . . .	155
8.2.3.	Registration Template for Properties . . . . .	156
8.2.4.	Registration Template for Parameters . . . . .	156
8.2.5.	Registration Template for Value Data Types . . . . .	157
8.2.6.	Registration Template for Values . . . . .	157
8.3.	Initial iCalendar Elements Registries . . . . .	158
8.3.1.	Components Registry . . . . .	158
8.3.2.	Properties Registry . . . . .	158
8.3.3.	Parameters Registry . . . . .	161
8.3.4.	Value Data Types Registry . . . . .	162
8.3.5.	Calendar User Types Registry . . . . .	162
8.3.6.	Free/Busy Time Types Registry . . . . .	163
8.3.7.	Participation Statuses Registry . . . . .	163
8.3.8.	Relationship Types Registry . . . . .	164
8.3.9.	Participation Roles Registry . . . . .	164
8.3.10.	Actions Registry . . . . .	165
8.3.11.	Classifications Registry . . . . .	165
8.3.12.	Methods Registry . . . . .	165

9. Acknowledgments . . . . .	165
10. References . . . . .	166
10.1. Normative References . . . . .	166
10.2. Informative References . . . . .	167
Appendix A. Differences from RFC 2445 . . . . .	169
A.1. New Restrictions . . . . .	169
A.2. Restrictions Removed . . . . .	169
A.3. Deprecated Features . . . . .	169

## 1. Introduction

The use of calendaring and scheduling has grown considerably in the last decade. Enterprise and inter-enterprise business has become dependent on rapid scheduling of events and actions using this information technology. This memo is intended to progress the level of interoperability possible between dissimilar calendaring and scheduling applications. This memo defines a MIME content type for exchanging electronic calendaring and scheduling information. The Internet Calendaring and Scheduling Core Object Specification, or iCalendar, allows for the capture and exchange of information normally stored within a calendaring and scheduling application; such as a Personal Information Manager (PIM) or a Group-Scheduling product.

The iCalendar format is suitable as an exchange format between applications or systems. The format is defined in terms of a MIME content type. This will enable the object to be exchanged using several transports, including but not limited to SMTP, HTTP, a file system, desktop interactive protocols such as the use of a memory-based clipboard or drag/drop interactions, point-to-point asynchronous communication, wired-network transport, or some form of unwired transport such as infrared.

The memo also provides for the definition of iCalendar object methods that will map this content type to a set of messages for supporting calendaring and scheduling operations such as requesting, replying to, modifying, and canceling meetings or appointments, to-dos, and journal entries. The iCalendar object methods can be used to define other calendaring and scheduling operations such as requesting for and replying with free/busy time data. Such a scheduling protocol is defined in the iCalendar Transport-independent Interoperability Protocol (iTIP) defined in [2446bis].

The memo also includes a formal grammar for the content type based on the Internet ABNF defined in [RFC5234]. This ABNF is required for the implementation of parsers and to serve as the definitive reference when ambiguities or questions arise in interpreting the descriptive prose definition of the memo. Additional restrictions

that could not easily be expressed with the ABNF syntax are specified as comments in the ABNF. Comments with normative statements should be treated as such.

## 2. Basic Grammar and Conventions

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].

This memo makes use of both a descriptive prose and a more formal notation for defining the calendaring and scheduling format.

The notation used in this memo is the ABNF notation of [RFC5234]. Readers intending on implementing the format defined in this memo should be familiar with this notation in order to properly interpret the specifications of this memo.

All numeric values used in this memo are given in decimal notation.

All names of properties, property parameters, enumerated property values, and property parameter values are case-insensitive. However, all other property values are case-sensitive, unless otherwise stated.

Note: All indented editorial notes, such as this one, are intended to provide the reader with additional information. The information is not essential to the building of an implementation conformant with this memo. The information is provided to highlight a particular feature or characteristic of the memo.

The format for the iCalendar object is based on the syntax of the text/directory media type [RFC2425]. While the iCalendar object is not a profile of the text/directory media type [RFC2425], it does reuse a number of the elements from the [RFC2425] specification.

### 2.1. Formatting Conventions

The elements defined in this memo are defined in prose. Many of the terms used to describe these have common usage that is different than the standards usage of this memo. In order to reference, within this memo, elements of the calendaring and scheduling model, core object (this memo), or interoperability protocol [2446bis] some formatting conventions have been used. Calendaring and scheduling roles are referred to in quoted-strings of text with the first character of each word in uppercase. For example, "Organizer" refers to a role of a "Calendar User" within the scheduling protocol defined by [2446bis]. Calendar components defined by this memo are referred to

with capitalized, quoted-strings of text. All calendar components start with the letter "V". For example, "VEVENT" refers to the event calendar component, "VTODO" refers to the to-do calendar component, and "VJOURNAL" refers to the daily journal calendar component. Scheduling methods defined by iTIP [2446bis] are referred to with capitalized, quoted-strings of text. For example, "REQUEST" refers to the method for requesting a scheduling calendar component be created or modified, and "REPLY" refers to the method a recipient of a request uses to update their status with the "Organizer" of the calendar component.

The properties defined by this memo are referred to with capitalized, quoted-strings of text, followed by the word "property". For example, "ATTENDEE" property refers to the iCalendar property used to convey the calendar address of a calendar user. Property parameters defined by this memo are referred to with lowercase, quoted-strings of text, followed by the word "parameter". For example, "value" parameter refers to the iCalendar property parameter used to override the default value type for a property value. Enumerated values defined by this memo are referred to with capitalized text, either alone or followed by the word "value". For example, the "MINUTELY" value can be used with the "FREQ" component of the "RECUR" value type to specify repeating components based on an interval of one minute or more.

The following table lists the different characters from the [US-ASCII] character set that is referenced in this document. For each character, the table specifies the character name used throughout this document, along with its US-ASCII decimal codepoint.

Character name	Decimal codepoint
HTAB	9
LF	10
CR	13
DQUOTE	22
SPACE	32
PLUS SIGN	43
COMMA	44
HYPHEN-MINUS	45
PERIOD	46
SOLIDUS	47
COLON	58
SEMICOLON	59
LATIN CAPITAL LETTER N	78
LATIN CAPITAL LETTER T	84
LATIN CAPITAL LETTER X	88
LATIN CAPITAL LETTER Z	90
BACKSLASH	92
LATIN SMALL LETTER N	110

## 2.2. Related Memos

Implementers will need to be familiar with several other memos that, along with this memo, form a framework for Internet calendaring and scheduling standards. This memo specifies a core specification of objects, value types, properties, and property parameters.

- o iTIP [2446bis] specifies an interoperability protocol for scheduling between different implementations;
- o iCalendar Message-Based Interoperability Protocol (iMIP) [2447bis] specifies an Internet email binding for [2446bis].

This memo does not attempt to repeat the specification of concepts or definitions from these other memos. Where possible, references are made to the memo that provides for the specification of these concepts or definitions.

## 3. iCalendar Object Specification

The following sections define the details of a Calendaring and Scheduling Core Object Specification. The Calendaring and Scheduling Core Object is a collection of calendaring and scheduling information. Typically, this information will consist of an iCalendar stream with one or more iCalendar objects. The body of the



iCalendar object consists of a sequence of calendar properties and one or more calendar components.

Section 3.1 defines the content line format; Section 3.2 defines the property parameter format; Section 3.3 defines the data types for property values; Section 3.4 defines the iCalendar object format; Section 3.5 defines the iCalendar property format; Section 3.6 defines the calendar component format; Section 3.7 defines calendar properties; and Section 3.8 defines calendar component properties.

This information is intended to be an integral part of the MIME content type registration. In addition, this information can be used independent of such content registration. In particular, this memo has direct applicability for use as a calendaring and scheduling exchange format in file-, memory-, or network-based transport mechanisms.

### 3.1. Content Lines

The iCalendar object is organized into individual lines of text, called content lines. Content lines are delimited by a line break, which is a CRLF sequence (CR character followed by LF character).

Lines of text SHOULD NOT be longer than 75 octets, excluding the line break. Long content lines SHOULD be split into a multiple line representations using a line "folding" technique. That is, a long line can be split between any two characters by inserting a CRLF immediately followed by a single linear white-space character (i.e., SPACE or HTAB). Any sequence of CRLF followed immediately by a single linear white-space character is ignored (i.e., removed) when processing the content type.

For example, the line:

DESCRIPTION:This is a long description that exists on a long line.

Can be represented as:

DESCRIPTION:This is a lo  
ng description  
that exists on a long line.

The process of moving from this folded multiple-line representation to its single-line representation is called "unfolding". Unfolding is accomplished by removing the CRLF and the linear white-space character that immediately follows.

When parsing a content line, folded lines MUST first be unfolded according to the unfolding procedure described above.

Note: It is possible for very simple implementations to generate improperly folded lines in the middle of a UTF-8 multi-octet sequence. For this reason, implementations need to unfold lines in such a way to properly restore the original sequence.

The content information associated with an iCalendar object is formatted using a syntax similar to that defined by [RFC2425]. That is, the content information consists of CRLF-separated content lines.

The following notation defines the lines of content in an iCalendar object:

```
contentline = name *(";" param) ":" value CRLF
; This ABNF is just a general definition for an initial parsing
; of the content line into its property name, parameter list,
; and value string

; When parsing a content line, folded lines MUST first
; be unfolded according to the unfolding procedure
; described above. When generating a content line, lines
; longer than 75 octets SHOULD be folded according to
; the folding procedure described above.

name = iana-token / x-name

iana-token = 1*(ALPHA / DIGIT / "-")
; iCalendar identifier registered with IANA

x-name = "X-" [vendorid "-"] 1*(ALPHA / DIGIT / "-")
; Reserved for experimental use.

vendorid = 3*(ALPHA / DIGIT)
; Vendor identification

param = param-name "=" param-value *("," param-value)
; Each property defines the specific ABNF for the parameters
; allowed on the property. Refer to specific properties for
; precise parameter ABNF.

param-name = iana-token / x-name

param-value = paramtext / quoted-string

paramtext = *SAFE-CHAR
```

```

value          = *VALUE-CHAR

quoted-string  = DQUOTE *QSAFE-CHAR DQUOTE

QSAFE-CHAR     = WSP / %x21 / %x23-7E / NON-US-ASCII
; Any character except CONTROL and DQUOTE

SAFE-CHAR      = WSP / %x21 / %x23-2B / %x2D-39 / %x3C-7E
                / NON-US-ASCII
; Any character except CONTROL, DQUOTE, ";", ":", ",", "

VALUE-CHAR     = WSP / %x21-7E / NON-US-ASCII
; Any textual character

NON-US-ASCII   = UTF8-2 / UTF8-3 / UTF8-4
; UTF8-2, UTF8-3, and UTF8-4 are defined in [RFC3629]

CONTROL        = %x00-08 / %x0A-1F / %x7F
; All the controls except HTAB

```

The property value component of a content line has a format that is property specific. Refer to the section describing each property for a definition of this format.

All names of properties, property parameters, enumerated property values and property parameter values are case-insensitive. However, all other property values are case-sensitive, unless otherwise stated.

### 3.1.1.1. List and Field Separators

Some properties and parameters allow a list of values. Values in a list of values **MUST** be separated by a COMMA character. There is no significance to the order of values in a list. For those parameter values (such as those that specify URI values) that are specified in quoted-strings, the individual quoted-strings are separated by a COMMA character.

Some property values are defined in terms of multiple parts. These structured property values **MUST** have their value parts separated by a SEMICOLON character.

Some properties allow a list of parameters. Each property parameter in a list of property parameters **MUST** be separated by a SEMICOLON character.

Property parameters with values containing a COLON character, a SEMICOLON character or a COMMA character MUST be placed in quoted text.

For example, in the following properties, a SEMICOLON is used to separate property parameters from each other and a COMMA character is used to separate property values in a value list.

```
ATTENDEE;RSVP=TRUE;ROLE=REQ-PARTICIPANT:mailto:jsmith@example.com
```

```
RDATE;VALUE=DATE:19970304,19970504,19970704,19970904
```

### 3.1.2. Multiple Values

Some properties defined in the iCalendar object can have multiple values. The general rule for encoding multi-valued items is to simply create a new content line for each value, including the property name. However, it should be noted that some properties support encoding multiple values in a single property by separating the values with a COMMA character. Individual property definitions should be consulted for determining whether a specific property allows multiple values and in which of these two forms. Multi-valued properties MUST NOT be used to specify multiple language variants of the same value. Calendar applications SHOULD display all values.

### 3.1.3. Binary Content

Binary content information in an iCalendar object SHOULD be referenced using a URI within a property value. That is, the binary content information SHOULD be placed in an external MIME entity that can be referenced by a URI from within the iCalendar object. In applications where this is not feasible, binary content information can be included within an iCalendar object, but only after first encoding it into text using the "BASE64" encoding method defined in [RFC4648]. Inline binary content SHOULD only be used in applications whose special circumstances demand that an iCalendar object be expressed as a single entity. A property containing inline binary content information MUST specify the "ENCODING" property parameter. Binary content information placed external to the iCalendar object MUST be referenced by a uniform resource identifier (URI).

The following example specifies an "ATTACH" property that references an attachment external to the iCalendar object with a URI reference:

```
ATTACH:http://example.com/public/quarterly-report.doc
```

The following example specifies an "ATTACH" property with inline binary encoded content information:

```
ATTACH;FMTTYPE=text/plain;ENCODING=BASE64;VALUE=BINARY:VGh1IH  
FlaWNrIGJyb3duIGZveCBqdWlwcYBvdmVYIHROZSBsYXp5IGRvZy4
```

#### 3.1.4. Character Set

There is not a property parameter to declare the charset used in a property value. The default charset for an iCalendar stream is UTF-8 as defined in [RFC3629].

The "charset" Content-Type parameter MUST be used in MIME transports to specify the charset being used.

#### 3.2. Property Parameters

A property can have attributes with which it is associated. These "property parameters" contain meta-information about the property or the property value. Property parameters are provided to specify such information as the location of an alternate text representation for a property value, the language of a text property value, the value type of the property value, and other attributes.

Property parameter values that contain the COLON, SEMICOLON, or COMMA character separators MUST be specified as quoted-string text values. Property parameter values MUST NOT contain the DQUOTE character. The DQUOTE character is used as a delimiter for parameter values that contain restricted characters or URI text. For example:

```
DESCRIPTION;ALTREP="cid:part1.0001@example.org":The Fall'98 Wild  
Wizards Conference - - Las Vegas\, NV\, USA
```

Property parameter values that are not in quoted-strings are case-insensitive.

The general property parameters defined by this memo are defined by the following notation:

```

icalparameter = altrepparam      ; Alternate text representation
                        / cnparam      ; Common name
                        / cutypeparam   ; Calendar user type
                        / delfromparam  ; Delegator
                        / deltoparam    ; Delegatee
                        / dirparam      ; Directory entry
                        / encodingparam  ; Inline encoding
                        / fmttypeparam   ; Format type
                        / fbtypeparam    ; Free/busy time type
                        / languageparam  ; Language for text
                        / memberparam    ; Group or list membership
                        / partstatparam   ; Participation status
                        / rangeparam     ; Recurrence identifier range
                        / trigrelparam   ; Alarm trigger relationship
                        / reltypeparam   ; Relationship type
                        / roleparam      ; Participation role
                        / rsvpparam      ; RSVP expectation
                        / sentbyparam    ; Sent by
                        / tzidparam      ; Reference to time zone object
                        / valuetyparam   ; Property value data type
                        / other-param

```

```

other-param  = (iana-param / x-param)

```

```

iana-param  = iana-token "=" param-value *("," param-value)
; Some other IANA-registered iCalendar parameter.

```

```

x-param      = x-name "=" param-value *("," param-value)
; A non-standard, experimental parameter.

```

Applications MUST ignore x-param and iana-param values they don't recognize.

### 3.2.1. Alternate Text Representation

Parameter Name: ALTREP

Purpose: To specify an alternate text representation for the property value.

Format Definition: This property parameter is defined by the following notation:

```
altrepparam = "ALTREP" "=" DQUOTE uri DQUOTE
```

Description: This parameter specifies a URI that points to an alternate representation for a textual property value. A property specifying this parameter MUST also include a value that reflects

the default representation of the text value. The URI parameter value **MUST** be specified in a quoted-string.

Note: While there is no restriction imposed on the URI schemes allowed for this parameter, Content Identifier (CID) [RFC2392], HTTP [RFC2616], and HTTPS [RFC2818] are the URI schemes most commonly used by current implementations.

**Example:**

```
DESCRIPTION;ALTREP="CID:part3.msg.970415T083000@example.com":
Project XYZ Review Meeting will include the following agenda
items: (a) Market Overview\, (b) Finances\, (c) Project Man
agement
```

The "ALTREP" property parameter value might point to a "text/html" content portion.

```
Content-Type:text/html
Content-Id:<part3.msg.970415T083000@example.com>
```

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <p>
      <b>Project XYZ Review Meeting</b> will include
      the following agenda items:
      <ol>
        <li>Market Overview</li>
        <li>Finances</li>
        <li>Project Management</li>
      </ol>
    </p>
  </body>
</html>
```

### 3.2.2. Common Name

Parameter Name: CN

Purpose: To specify the common name to be associated with the calendar user specified by the property.

Format Definition: This property parameter is defined by the following notation:

cnparam = "CN" "=" param-value

**Description:** This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter specifies the common name to be associated with the calendar user specified by the property. The parameter value is text. The parameter value can be used for display text to be associated with the calendar address specified by the property.

**Example:**

ORGANIZER;CN="John Smith":mailto:jsmith@example.com

### 3.2.3. Calendar User Type

**Parameter Name:** CUTYPE

**Purpose:** To identify the type of calendar user specified by the property.

**Format Definition:** This property parameter is defined by the following notation:

```
cutypeparam      = "CUTYPE" "="
                  ( "INDIVIDUAL"    ; An individual
                  / "GROUP"          ; A group of individuals
                  / "RESOURCE"       ; A physical resource
                  / "ROOM"           ; A room resource
                  / "UNKNOWN"        ; Otherwise not known
                  / x-name            ; Experimental type
                  / iana-token)      ; Other IANA-registered
                                  ; type
; Default is INDIVIDUAL
```

**Description:** This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter identifies the type of calendar user specified by the property. If not specified on a property that allows this parameter, the default is INDIVIDUAL. Applications **MUST** treat x-name and iana-token values they don't recognize the same way as they would the UNKNOWN value.

**Example:**

ATTENDEE;CUTYPE=GROUP:mailto:ietf-calsch@example.org



### 3.2.4. Delegators

Parameter Name: DELEGATED-FROM

Purpose: To specify the calendar users that have delegated their participation to the calendar user specified by the property.

Format Definition: This property parameter is defined by the following notation:

```
delfromparam      = "DELEGATED-FROM" "=" DQUOTE cal-address
                    DQUOTE *("," DQUOTE cal-address DQUOTE)
```

Description: This parameter can be specified on properties with a CAL-ADDRESS value type. This parameter specifies those calendar users that have delegated their participation in a group-scheduled event or to-do to the calendar user specified by the property. The individual calendar address parameter values MUST each be specified in a quoted-string.

Example:

```
ATTENDEE;DELEGATED-FROM="mailto:jsmith@example.com":mailto:
jdoe@example.com
```

### 3.2.5. Delegates

Parameter Name: DELEGATED-TO

Purpose: To specify the calendar users to whom the calendar user specified by the property has delegated participation.

Format Definition: This property parameter is defined by the following notation:

```
deltoparam = "DELEGATED-TO" "=" DQUOTE cal-address DQUOTE
            *("," DQUOTE cal-address DQUOTE)
```

Description: This parameter can be specified on properties with a CAL-ADDRESS value type. This parameter specifies those calendar users whom have been delegated participation in a group-scheduled event or to-do by the calendar user specified by the property. The individual calendar address parameter values MUST each be specified in a quoted-string.

**Example:**

```
ATTENDEE;DELEGATED-TO="mailto:jdoe@example.com","mailto:jgpublic@example.com":mailto:jsmith@example.com
```

**3.2.6. Directory Entry Reference**

Parameter Name: DIR

Purpose: To specify reference to a directory entry associated with the calendar user specified by the property.

Format Definition: This property parameter is defined by the following notation:

```
dirparam = "DIR" "=" DQUOTE uri DQUOTE
```

Description: This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter specifies a reference to the directory entry associated with the calendar user specified by the property. The parameter value is a URI. The URI parameter value MUST be specified in a quoted-string.

Note: While there is no restriction imposed on the URI schemes allowed for this parameter, CID [RFC2392], DATA [RFC2397], FILE [RFC1738], FTP [RFC1738], HTTP [RFC2616], HTTPS [RFC2818], LDAP [RFC4516], and MID [RFC2392] are the URI schemes most commonly used by current implementations.

**Example:**

```
ORGANIZER;DIR="ldap://example.com:6666/o=ABC%20Industries,c=US???(cn=Jim%20Dolittle)":mailto:jimdo@example.com
```

**3.2.7. Inline Encoding**

Parameter Name: ENCODING

Purpose: To specify an alternate inline encoding for the property value.

Format Definition: This property parameter is defined by the following notation:

```

encodingparam      = "ENCODING" "="
                    ( "8BIT"
                      ; "8bit" text encoding is defined in [RFC2045]
                      / "BASE64"
                      ; "BASE64" binary encoding format is defined in [RFC4648]
                    )

```

**Description:** This property parameter identifies the inline encoding used in a property value. The default encoding is "8BIT", corresponding to a property value consisting of text. The "BASE64" encoding type corresponds to a property value encoded using the "BASE64" encoding defined in [RFC2045].

If the value type parameter is ";VALUE=BINARY", then the inline encoding parameter **MUST** be specified with the value ";ENCODING=BASE64".

**Example:**

```

ATTACH;FMTTYPE=text/plain;ENCODING=BASE64;VALUE=BINARY:TG9yZW
0gaXBzdW0gZG9sb3Igc2l0IGFtZXQsIGNvbnNlY3RldHVyIGFkaXBpc2ljaW
5nIGVsaXQsIHNLZCBkbyBlaXVzbW9kIHRlbXBvcjBpbmNpZGlkdW50IHV0IG
xhYm9yZSBldCBkb2xvcuUgbWFnbmEgYWxpXVhLiBVdCB1bm1tIGFkIGlpm
ltIHZlbn1hbSwgcXVpcyBub3N0cnVkiGV4ZXJjaXRhdGlvbiB1bGxhbWNvIG
xhYm9yaXMgbmlzaSBldCBhbGlxdWlwIGV4IGVhIGNvbWl1vZG8gY29uc2VxdW
F0LiBEdWl1IGFldGUgaXJlcmUgZG9sb3IgaW4gcmVwcmVoZW5kZXJpdCBpbi
B2b2xlcHRhdGUgdmVsaXQgZXNzZSBjaWxsZW0gZG9sb3JlIGVlIGZlZ2lhdC
BudWxsYSBwYXJpYXRlcj4gRXhjZXB0ZXVYIHNPbnQgb2NjYWVjYXQgY3VwaW
RhdGF0IG5vbiBwcm9pZGVudCwgc3VudCBpbiBjdWxwYSBxdWkgb2ZmaWNpYS
BkZXNlcnVudCBtb2xsaXQgYW5pbSBpZCB1c3QgbGFib3J1bS4=

```

### 3.2.8. Format Type

**Parameter Name:** FMTTYPE

**Purpose:** To specify the content type of a referenced object.

**Format Definition:** This property parameter is defined by the following notation:

```

fmttypeparam = "FMTTYPE" "=" type-name "/" subtype-name
              ; Where "type-name" and "subtype-name" are
              ; defined in Section 4.2 of [RFC4288].

```

**Description:** This parameter can be specified on properties that are used to reference an object. The parameter specifies the media type [RFC4288] of the referenced object. For example, on the "ATTACH" property, an FTP type URI value does not, by itself,

necessarily convey the type of content associated with the resource. The parameter value **MUST** be the text for either an IANA-registered media type or a non-standard media type.

Example:

```
ATTACH;FMTTYPE=application/msword:ftp://example.com/pub/docs/
agenda.doc
```

### 3.2.9. Free/Busy Time Type

Parameter Name: FBTYPE

Purpose: To specify the free or busy time type.

Format Definition: This property parameter is defined by the following notation:

```
fbtypeparam          = "FBTYPE" "=" ( "FREE" / "BUSY"
                                   / "BUSY-UNAVAILABLE" / "BUSY-TENTATIVE"
                                   / x-name
                                   ; Some experimental iCalendar free/busy type.
                                   / iana-token
                                   ; Some other IANA-registered iCalendar free/busy type.
```

Description: This parameter specifies the free or busy time type. The value **FREE** indicates that the time interval is free for scheduling. The value **BUSY** indicates that the time interval is busy because one or more events have been scheduled for that interval. The value **BUSY-UNAVAILABLE** indicates that the time interval is busy and that the interval can not be scheduled. The value **BUSY-TENTATIVE** indicates that the time interval is busy because one or more events have been tentatively scheduled for that interval. If not specified on a property that allows this parameter, the default is **BUSY**. Applications **MUST** treat **x-name** and **iana-token** values they don't recognize the same way as they would the **BUSY** value.

Example: The following is an example of this parameter on a "FREEBUSY" property.

```
FREEBUSY;FBTYPE=BUSY:19980415T133000Z/19980415T170000Z
```

### 3.2.10. Language

Parameter Name: LANGUAGE

Purpose: To specify the language for text values in a property or property parameter.

Format Definition: This property parameter is defined by the following notation:

```
languageparam = "LANGUAGE" "=" language
```

```
language = Language-Tag  
          ; As defined in [RFC5646].
```

Description: This parameter identifies the language of the text in the property value and of all property parameter values of the property. The value of the "LANGUAGE" property parameter is that defined in [RFC5646].

For transport in a MIME entity, the Content-Language header field can be used to set the default language for the entire body part. Otherwise, no default language is assumed.

Example: The following are examples of this parameter on the "SUMMARY" and "LOCATION" properties:

```
SUMMARY;LANGUAGE=en-US:Company Holiday Party
```

```
LOCATION;LANGUAGE=en:Germany
```

```
LOCATION;LANGUAGE=no:Tyskland
```

### 3.2.11. Group or List Membership

Parameter Name: MEMBER

Purpose: To specify the group or list membership of the calendar user specified by the property.

Format Definition: This property parameter is defined by the following notation:

```
memberparam      = "MEMBER" "=" DQUOTE cal-address DQUOTE  
                  *("," DQUOTE cal-address DQUOTE)
```

**Description:** This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter identifies the groups or list membership for the calendar user specified by the property. The parameter value is either a single calendar address in a quoted-string or a COMMA-separated list of calendar addresses, each in a quoted-string. The individual calendar address parameter values **MUST** each be specified in a quoted-string.

**Example:**

```
ATTENDEE;MEMBER="mailto:ietf-calsch@example.org":mailto:
jsmith@example.com
```

```
ATTENDEE;MEMBER="mailto:projectA@example.com","mailto:pr
ojectB@example.com":mailto:janedoe@example.com
```

### 3.2.12. Participation Status

**Parameter Name:** PARTSTAT

**Purpose:** To specify the participation status for the calendar user specified by the property.

**Format Definition:** This property parameter is defined by the following notation:

```
partstatparam      = "PARTSTAT" "="
                    (partstat-event
                     / partstat-todo
                     / partstat-jour)

partstat-event      = ("NEEDS-ACTION"      ; Event needs action
                     / "ACCEPTED"          ; Event accepted
                     / "DECLINED"          ; Event declined
                     / "TENTATIVE"         ; Event tentatively
                                     ; accepted
                     / "DELEGATED"         ; Event delegated
                     / x-name              ; Experimental status
                     / iana-token          ; Other IANA-registered
                                     ; status
                     ; These are the participation statuses for a "VEVENT".
                     ; Default is NEEDS-ACTION.

partstat-todo       = ("NEEDS-ACTION"      ; To-do needs action
                     / "ACCEPTED"          ; To-do accepted
                     / "DECLINED"          ; To-do declined
                     / "TENTATIVE"         ; To-do tentatively
                                     ; accepted
```

```

/ "DELEGATED"           ; To-do delegated
/ "COMPLETED"          ; To-do completed
                        ; COMPLETED property has
                        ; DATE-TIME completed
/ "IN-PROCESS"          ; To-do in process of
                        ; being completed
/ x-name                 ; Experimental status
/ iana-token)            ; Other IANA-registered
                        ; status
; These are the participation statuses for a "VTODO".
; Default is NEEDS-ACTION.

```

```

partstat-jour    = ("NEEDS-ACTION"      ; Journal needs action
/ "ACCEPTED"      ; Journal accepted
/ "DECLINED"      ; Journal declined
/ x-name          ; Experimental status
/ iana-token)     ; Other IANA-registered
                  ; status
; These are the participation statuses for a "VJOURNAL".
; Default is NEEDS-ACTION.

```

**Description:** This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter identifies the participation status for the calendar user specified by the property value. The parameter values differ depending on whether they are associated with a group-scheduled "VEVENT", "VTODO", or "VJOURNAL". The values MUST match one of the values allowed for the given calendar component. If not specified on a property that allows this parameter, the default value is NEEDS-ACTION. Applications MUST treat x-name and iana-token values they don't recognize the same way as they would the NEEDS-ACTION value.

**Example:**

```
ATTENDEE;PARTSTAT=DECLINED:mailto:jsmith@example.com
```

### 3.2.13. Recurrence Identifier Range

**Parameter Name:** RANGE

**Purpose:** To specify the effective range of recurrence instances from the instance specified by the recurrence identifier specified by the property.

**Format Definition:** This property parameter is defined by the following notation:

```

rangeparam = "RANGE" "=" "THISANDFUTURE"
; To specify the instance specified by the recurrence identifier
; and all subsequent recurrence instances.

```

**Description:** This parameter can be specified on a property that specifies a recurrence identifier. The parameter specifies the effective range of recurrence instances that is specified by the property. The effective range is from the recurrence identifier specified by the property. If this parameter is not specified on an allowed property, then the default range is the single instance specified by the recurrence identifier value of the property. The parameter value can only be "THISANDFUTURE" to indicate a range defined by the recurrence identifier and all subsequent instances. The value "THISANDPRIOR" is deprecated by this revision of iCalendar and MUST NOT be generated by applications.

**Example:**

```

RECURRENCE-ID;RANGE=THISANDFUTURE:19980401T133000Z

```

### 3.2.14. Alarm Trigger Relationship

**Parameter Name:** RELATED

**Purpose:** To specify the relationship of the alarm trigger with respect to the start or end of the calendar component.

**Format Definition:** This property parameter is defined by the following notation:

```

trigrelparam      = "RELATED" "="
                   ( "START"          ; Trigger off of start
                   / "END"            ; Trigger off of end

```

**Description:** This parameter can be specified on properties that specify an alarm trigger with a "DURATION" value type. The parameter specifies whether the alarm will trigger relative to the start or end of the calendar component. The parameter value START will set the alarm to trigger off the start of the calendar component; the parameter value END will set the alarm to trigger off the end of the calendar component. If the parameter is not specified on an allowable property, then the default is START.

**Example:**

```

TRIGGER;RELATED=END:PT5M

```



### 3.2.15. Relationship Type

Parameter Name: RELTYPE

Purpose: To specify the type of hierarchical relationship associated with the calendar component specified by the property.

Format Definition: This property parameter is defined by the following notation:

```
reltypeparam      = "RELTYPE" "="  
                    ( "PARENT"      ; Parent relationship - Default  
                      / "CHILD"      ; Child relationship  
                      / "SIBLING"    ; Sibling relationship  
                      / iana-token    ; Some other IANA-registered  
                                  ; iCalendar relationship type  
                      / x-name)      ; A non-standard, experimental  
                                  ; relationship type
```

Description: This parameter can be specified on a property that references another related calendar. The parameter specifies the hierarchical relationship type of the calendar component referenced by the property. The parameter value can be PARENT, to indicate that the referenced calendar component is a superior of calendar component; CHILD to indicate that the referenced calendar component is a subordinate of the calendar component; or SIBLING to indicate that the referenced calendar component is a peer of the calendar component. If this parameter is not specified on an allowable property, the default relationship type is PARENT. Applications MUST treat x-name and iana-token values they don't recognize the same way as they would the PARENT value.

Example:

```
RELATED-TO;RELTYPE=SIBLING:19960401-080045-4000F192713@  
example.com
```

### 3.2.16. Participation Role

Parameter Name: ROLE

Purpose: To specify the participation role for the calendar user specified by the property.

Format Definition: This property parameter is defined by the following notation:

```

roleparam = "ROLE" "="
            ("CHAIR"                ; Indicates chair of the
            / "REQ-PARTICIPANT"     ; calendar entity
            / "OPT-PARTICIPANT"     ; Indicates a participant whose
            / "NON-PARTICIPANT"     ; participation is required
            / "NON-PARTICIPANT"     ; Indicates a participant whose
            / "NON-PARTICIPANT"     ; participation is optional
            / "NON-PARTICIPANT"     ; Indicates a participant who
            / "NON-PARTICIPANT"     ; is copied for information
            / "NON-PARTICIPANT"     ; purposes only
            / x-name                 ; Experimental role
            / iana-token)            ; Other IANA role
; Default is REQ-PARTICIPANT

```

**Description:** This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter specifies the participation role for the calendar user specified by the property in the group schedule calendar component. If not specified on a property that allows this parameter, the default value is REQ-PARTICIPANT. Applications **MUST** treat x-name and iana-token values they don't recognize the same way as they would the REQ-PARTICIPANT value.

**Example:**

```
ATTENDEE;ROLE=CHAIR:mailto:mrbig@example.com
```

### 3.2.17. RSVP Expectation

**Parameter Name:** RSVP

**Purpose:** To specify whether there is an expectation of a favor of a reply from the calendar user specified by the property value.

**Format Definition:** This property parameter is defined by the following notation:

```

rsvpparam = "RSVP" "=" ("TRUE" / "FALSE")
; Default is FALSE

```

**Description:** This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter identifies the expectation of a reply from the calendar user specified by the property value. This parameter is used by the "Organizer" to request a participation status reply from an "Attendee" of a group-scheduled event or to-do. If not specified on a property that allows this parameter, the default value is FALSE.

Example:

```
ATTENDEE;RSVP=TRUE:mailto:jsmith@example.com
```

### 3.2.18. Sent By

Parameter Name: SENT-BY

Purpose: To specify the calendar user that is acting on behalf of the calendar user specified by the property.

Format Definition: This property parameter is defined by the following notation:

```
sentbyparam = "SENT-BY" "=" DQUOTE cal-address DQUOTE
```

Description: This parameter can be specified on properties with a CAL-ADDRESS value type. The parameter specifies the calendar user that is acting on behalf of the calendar user specified by the property. The parameter value MUST be a mailto URI as defined in [RFC2368]. The individual calendar address parameter values MUST each be specified in a quoted-string.

Example:

```
ORGANIZER;SENT-BY="mailto:sray@example.com":mailto:  
jsmith@example.com
```

### 3.2.19. Time Zone Identifier

Parameter Name: TZID

Purpose: To specify the identifier for the time zone definition for a time component in the property value.

Format Definition: This property parameter is defined by the following notation:

```
tzidparam = "TZID" "=" [tzidprefix] paramtext
```

```
tzidprefix = "/"
```

Description: This parameter MUST be specified on the "DTSTART", "DTEND", "DUE", "EXDATE", and "RDATE" properties when either a DATE-TIME or TIME value type is specified and when the value is neither a UTC or a "floating" time. Refer to the DATE-TIME or TIME value type definition for a description of UTC and "floating time" formats. This property parameter specifies a text value

that uniquely identifies the "VTIMEZONE" calendar component to be used when evaluating the time portion of the property. The value of the "TZID" property parameter will be equal to the value of the "TZID" property for the matching time zone definition. An individual "VTIMEZONE" calendar component **MUST** be specified for each unique "TZID" parameter value specified in the iCalendar object.

The parameter **MUST** be specified on properties with a DATE-TIME value if the DATE-TIME is not either a UTC or a "floating" time. Failure to include and follow VTIMEZONE definitions in iCalendar objects may lead to inconsistent understanding of the local time at any given location.

The presence of the SOLIDUS character as a prefix, indicates that this "TZID" represents a unique ID in a globally defined time zone registry (when such registry is defined).

Note: This document does not define a naming convention for time zone identifiers. Implementers may want to use the naming conventions defined in existing time zone specifications such as the public-domain TZ database [TZDB]. The specification of globally unique time zone identifiers is not addressed by this document and is left for future study.

The following are examples of this property parameter:

```
DTSTART;TZID=America/New_York:19980119T020000
```

```
DTEND;TZID=America/New_York:19980119T030000
```

The "TZID" property parameter **MUST NOT** be applied to DATE properties and DATE-TIME or TIME properties whose time values are specified in UTC.

The use of local time in a DATE-TIME or TIME value without the "TZID" property parameter is to be interpreted as floating time, regardless of the existence of "VTIMEZONE" calendar components in the iCalendar object.

For more information, see the sections on the value types DATE-TIME and TIME.

### 3.2.20. Value Data Types

Parameter Name: VALUE

Purpose: To explicitly specify the value type format for a property value.

Format Definition: This property parameter is defined by the following notation:

```
valuetypeparam = "VALUE" "=" valuetype
```

```
valuetype = ("BINARY"  
             / "BOOLEAN"  
             / "CAL-ADDRESS"  
             / "DATE"  
             / "DATE-TIME"  
             / "DURATION"  
             / "FLOAT"  
             / "INTEGER"  
             / "PERIOD"  
             / "RECUR"  
             / "TEXT"  
             / "TIME"  
             / "URI"  
             / "UTC-OFFSET"  
             / x-name  
             ; Some experimental iCalendar value type.  
             / iana-token)  
             ; Some other IANA-registered iCalendar value type.
```

Description: This parameter specifies the value type and format of the property value. The property values MUST be of a single value type. For example, a "RDATE" property cannot have a combination of DATE-TIME and TIME value types.

If the property's value is the default value type, then this parameter need not be specified. However, if the property's default value type is overridden by some other allowable value type, then this parameter MUST be specified.

Applications MUST preserve the value data for x-name and iana-token values that they don't recognize without attempting to interpret or parse the value data.

### 3.3. Property Value Data Types

The properties in an iCalendar object are strongly typed. The definition of each property restricts the value to be one of the value data types, or simply value types, defined in this section. The value type for a property will either be specified implicitly as the default value type or will be explicitly specified with the "VALUE" parameter. If the value type of a property is one of the alternate valid types, then it **MUST** be explicitly specified with the "VALUE" parameter.

#### 3.3.1. Binary

Value Name: BINARY

Purpose: This value type is used to identify properties that contain a character encoding of inline binary data. For example, an inline attachment of a document might be included in an iCalendar object.

Format Definition: This value type is defined by the following notation:

binary = \*(4b-char) [b-end]  
; A "BASE64" encoded character string, as defined by [RFC4648].

b-end = (2b-char "==") / (3b-char "=")

b-char = ALPHA / DIGIT / "+" / "/"

Description: Property values with this value type **MUST** also include the inline encoding parameter sequence of ";ENCODING=BASE64". That is, all inline binary data **MUST** first be character encoded using the "BASE64" encoding method defined in [RFC2045]. No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

**Example:** The following is an example of a "BASE64" encoded binary value data:

```
ATTACH;FMTTYPE=image/vnd.microsoft.icon;ENCODING=BASE64;VALUE
=BINARY:AAABAAEAEBAQAAEABAAoAQAAFgAAACgAAAAQAAAAIAAAAAEABAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGIAAAICAgADAwMAA///AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ACECQ0QgEgAAQxQzM0E0AABERCRCREQAADRDJEJEQwAAAhA0QwEQAAAAAERE
AAAAAAAAAREQAAAAAAAKQgAAAAAAAAAMgAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA
```

### 3.3.2. Boolean

**Value Name:** BOOLEAN

**Purpose:** This value type is used to identify properties that contain either a "TRUE" or "FALSE" Boolean value.

**Format Definition:** This value type is defined by the following notation:

```
boolean      = "TRUE" / "FALSE"
```

**Description:** These values are case-insensitive text. No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

**Example:** The following is an example of a hypothetical property that has a BOOLEAN value type:

```
TRUE
```

### 3.3.3. Calendar User Address

**Value Name:** CAL-ADDRESS

**Purpose:** This value type is used to identify properties that contain a calendar user address.

**Format Definition:** This value type is defined by the following notation:

```
cal-address      = uri
```

**Description:** The value is a URI as defined by [RFC3986] or any other IANA-registered form for a URI. When used to address an Internet

email transport address for a calendar user, the value MUST be a mailto URI, as defined by [RFC2368]. No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

Example:

mailto:jane\_doe@example.com

### 3.3.4. Date

Value Name: DATE

Purpose: This value type is used to identify values that contain a calendar date.

Format Definition: This value type is defined by the following notation:

```

date                = date-value

date-value           = date-fullyear date-month date-mday
date-fullyear        = 4DIGIT
date-month           = 2DIGIT          ;01-12
date-mday            = 2DIGIT          ;01-28, 01-29, 01-30, 01-31
                        ;based on month/year

```

Description: If the property permits, multiple "date" values are specified as a COMMA-separated list of values. The format for the value type is based on the [ISO.8601.2004] complete representation, basic format for a calendar date. The textual format specifies a four-digit year, two-digit month, and two-digit day of the month. There are no separator characters between the year, month, and day component text.

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

Example: The following represents July 14, 1997:

19970714

### 3.3.5. Date-Time

Value Name: DATE-TIME

Purpose: This value type is used to identify values that specify a precise calendar date and time of day.



**Format Definition:** This value type is defined by the following notation:

```
date-time = date "T" time ;As specified in the DATE and TIME
           ;value definitions
```

**Description:** If the property permits, multiple "DATE-TIME" values are specified as a COMMA-separated list of values. No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

The "DATE-TIME" value type is used to identify values that contain a precise calendar date and time of day. The format is based on the [ISO.8601.2004] complete representation, basic format for a calendar date and time of day. The text format is a concatenation of the "date", followed by the LATIN CAPITAL LETTER T character, the time designator, followed by the "time" format.

The "DATE-TIME" value type expresses time values in three forms:

The form of date and time with UTC offset MUST NOT be used. For example, the following is not valid for a DATE-TIME value:

```
19980119T230000-0800 ;Invalid time format
```

#### FORM #1: DATE WITH LOCAL TIME

The date with local time form is simply a DATE-TIME value that does not contain the UTC designator nor does it reference a time zone. For example, the following represents January 18, 1998, at 11 PM:

```
19980118T230000
```

DATE-TIME values of this type are said to be "floating" and are not bound to any time zone in particular. They are used to represent the same hour, minute, and second value regardless of which time zone is currently being observed. For example, an event can be defined that indicates that an individual will be busy from 11:00 AM to 1:00 PM every day, no matter which time zone the person is in. In these cases, a local time can be specified. The recipient of an iCalendar object with a property value consisting of a local time, without any relative time zone information, SHOULD interpret the value as being fixed to whatever time zone the "ATTENDEE" is in at any given moment. This means that two "Attendees", in different time zones, receiving the same event definition as a floating time, may be participating in the

event at different actual times. Floating time SHOULD only be used where that is the reasonable behavior.

In most cases, a fixed time is desired. To properly communicate a fixed time in a property value, either UTC time or local time with time zone reference MUST be specified.

The use of local time in a DATE-TIME value without the "TZID" property parameter is to be interpreted as floating time, regardless of the existence of "VTIMEZONE" calendar components in the iCalendar object.

#### FORM #2: DATE WITH UTC TIME

The date with UTC time, or absolute time, is identified by a LATIN CAPITAL LETTER Z suffix character, the UTC designator, appended to the time value. For example, the following represents January 19, 1998, at 0700 UTC:

```
19980119T070000Z
```

The "TZID" property parameter MUST NOT be applied to DATE-TIME properties whose time values are specified in UTC.

#### FORM #3: DATE WITH LOCAL TIME AND TIME ZONE REFERENCE

The date and local time with reference to time zone information is identified by the use the "TZID" property parameter to reference the appropriate time zone definition. "TZID" is discussed in detail in Section 3.2.19. For example, the following represents 2:00 A.M. in New York on January 19, 1998:

```
TZID=America/New_York:19980119T020000
```

If, based on the definition of the referenced time zone, the local time described occurs more than once (when changing from daylight to standard time), the DATE-TIME value refers to the first occurrence of the referenced time. Thus, TZID=America/New\_York:20071104T013000 indicates November 4, 2007 at 1:30 A.M. EDT (UTC-04:00). If the local time described does not occur (when changing from standard to daylight time), the DATE-TIME value is interpreted using the UTC offset before the gap in local times. Thus, TZID=America/New\_York:20070311T023000 indicates March 11, 2007 at 3:30 A.M. EDT (UTC-04:00), one hour after 1:30 A.M. EST (UTC-05:00).

A time value **MUST** only specify the second 60 when specifying a positive leap second. For example:

```
19970630T235960Z
```

Implementations that do not support leap seconds **SHOULD** interpret the second 60 as equivalent to the second 59.

**Example:** The following represents July 14, 1997, at 1:30 PM in New York City in each of the three time formats, using the "DTSTART" property.

```
DTSTART:19970714T133000          ; Local time
DTSTART:19970714T173000Z        ; UTC time
DTSTART;TZID=America/New_York:19970714T133000
                                ; Local time and time
                                ; zone reference
```

### 3.3.6. Duration

**Value Name:** DURATION

**Purpose:** This value type is used to identify properties that contain a duration of time.

**Format Definition:** This value type is defined by the following notation:

```
dur-value = ([ "+" ] / "-") "P" (dur-date / dur-time / dur-week)

dur-date   = dur-day [dur-time]
dur-time   = "T" (dur-hour / dur-minute / dur-second)
dur-week   = 1*DIGIT "W"
dur-hour   = 1*DIGIT "H" [dur-minute]
dur-minute = 1*DIGIT "M" [dur-second]
dur-second = 1*DIGIT "S"
dur-day    = 1*DIGIT "D"
```

**Description:** If the property permits, multiple "duration" values are specified by a COMMA-separated list of values. The format is based on the [ISO.8601.2004] complete representation basic format with designators for the duration of time. The format can represent nominal durations (weeks and days) and accurate durations (hours, minutes, and seconds). Note that unlike [ISO.8601.2004], this value type doesn't support the "Y" and "M" designators to specify durations in terms of years and months.

The duration of a week or a day depends on its position in the calendar. In the case of discontinuities in the time scale, such as the change from standard time to daylight time and back, the computation of the exact duration requires the subtraction or addition of the change of duration of the discontinuity. Leap seconds **MUST NOT** be considered when computing an exact duration. When computing an exact duration, the greatest order time components **MUST** be added first, that is, the number of days **MUST** be added first, followed by the number of hours, number of minutes, and number of seconds.

Negative durations are typically used to schedule an alarm to trigger before an associated time (see Section 3.8.6.3).

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) are defined for this value type.

Example: A duration of 15 days, 5 hours, and 20 seconds would be:

P15DT5H0M20S

A duration of 7 weeks would be:

P7W

### 3.3.7. Float

Value Name: FLOAT

Purpose: This value type is used to identify properties that contain a real-number value.

Format Definition: This value type is defined by the following notation:

float = ([ "+" ] / "-" ) 1 \* DIGIT [ "." 1 \* DIGIT ]

Description: If the property permits, multiple "float" values are specified by a COMMA-separated list of values.

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

Example:

1000000.0000001  
1.333  
-3.14

### 3.3.8. Integer

Value Name: INTEGER

Purpose: This value type is used to identify properties that contain a signed integer value.

Format Definition: This value type is defined by the following notation:

integer = (["+" / "-"] 1\*DIGIT

Description: If the property permits, multiple "integer" values are specified by a COMMA-separated list of values. The valid range for "integer" is -2147483648 to 2147483647. If the sign is not specified, then the value is assumed to be positive.

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

Example:

1234567890  
-1234567890  
+1234567890  
432109876

### 3.3.9. Period of Time

Value Name: PERIOD

Purpose: This value type is used to identify values that contain a precise period of time.

Format Definition: This value type is defined by the following notation:

period = period-explicit / period-start

period-explicit = date-time "/" date-time  
; [ISO.8601.2004] complete representation basic format for a  
; period of time consisting of a start and end. The start MUST  
; be before the end.

period-start = date-time "/" dur-value  
; [ISO.8601.2004] complete representation basic format for a  
; period of time consisting of a start and positive duration  
; of time.

**Description:** If the property permits, multiple "period" values are specified by a COMMA-separated list of values. There are two forms of a period of time. First, a period of time is identified by its start and its end. This format is based on the [ISO.8601.2004] complete representation, basic format for "DATE-TIME" start of the period, followed by a SOLIDUS character followed by the "DATE-TIME" of the end of the period. The start of the period MUST be before the end of the period. Second, a period of time can also be defined by a start and a positive duration of time. The format is based on the [ISO.8601.2004] complete representation, basic format for the "DATE-TIME" start of the period, followed by a SOLIDUS character, followed by the [ISO.8601.2004] basic format for "DURATION" of the period.

**Example:** The period starting at 18:00:00 UTC, on January 1, 1997 and ending at 07:00:00 UTC on January 2, 1997 would be:

```
19970101T180000Z/19970102T070000Z
```

The period start at 18:00:00 on January 1, 1997 and lasting 5 hours and 30 minutes would be:

```
19970101T180000Z/PT5H30M
```

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

### 3.3.10. Recurrence Rule

**Value Name:** RECUR

**Purpose:** This value type is used to identify properties that contain a recurrence rule specification.

**Format Definition:** This value type is defined by the following notation:

```
recur          = recur-rule-part *( ";" recur-rule-part )
                ;
                ; The rule parts are not ordered in any
                ; particular sequence.
                ;
                ; The FREQ rule part is REQUIRED,
                ; but MUST NOT occur more than once.
                ;
                ; The UNTIL or COUNT rule parts are OPTIONAL,
                ; but they MUST NOT occur in the same 'recur'.
                ;
```

; The other rule parts are OPTIONAL,  
; but MUST NOT occur more than once.

```
recur-rule-part = ( "FREQ" "=" freq )
                  / ( "UNTIL" "=" enddate )
                  / ( "COUNT" "=" 1*DIGIT )
                  / ( "INTERVAL" "=" 1*DIGIT )
                  / ( "BYSECOND" "=" byseclist )
                  / ( "BYMINUTE" "=" byminlist )
                  / ( "BYHOUR" "=" byhrlist )
                  / ( "BYDAY" "=" byweekdaylist )
                  / ( "BYMONTHDAY" "=" bymodaylist )
                  / ( "BYYEARDAY" "=" byyrdaylist )
                  / ( "BYWEEKNO" "=" bywknolist )
                  / ( "BYMONTH" "=" bymolist )
                  / ( "BYSETPOS" "=" bysplist )
                  / ( "WKST" "=" weekday )

freq             = "SECONDLY" / "MINUTELY" / "HOURLY" / "DAILY"
                  / "WEEKLY" / "MONTHLY" / "YEARLY"

enddate         = date / date-time

byseclist       = ( seconds *("," seconds) )

seconds         = 1*2DIGIT           ;0 to 60

byminlist       = ( minutes *("," minutes) )

minutes         = 1*2DIGIT           ;0 to 59

byhrlist        = ( hour *("," hour) )

hour            = 1*2DIGIT           ;0 to 23

byweekdaylist   = ( weekdaynum *("," weekdaynum) )

weekdaynum      = [[plus / minus] ordwk] weekday

plus            = "+"

minus           = "-"

ordwk           = 1*2DIGIT           ;1 to 53

weekday         = "SU" / "MO" / "TU" / "WE" / "TH" / "FR" / "SA"
;Corresponding to SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY,
;FRIDAY, and SATURDAY days of the week.
```

```

bymodaylist = ( monthdaynum *("," monthdaynum) )

monthdaynum = [plus / minus] ordmoday

ordmoday    = 1*2DIGIT          ;1 to 31

byyrdaylist = ( yeardaynum *("," yeardaynum) )

yeardaynum  = [plus / minus] ordyrday

ordyrday    = 1*3DIGIT          ;1 to 366

bywknolist  = ( weeknum *("," weeknum) )

weeknum     = [plus / minus] ordwk

bymolist    = ( monthnum *("," monthnum) )

monthnum    = 1*2DIGIT          ;1 to 12

bysplist    = ( setposday *("," setposday) )

setposday   = yeardaynum

```

**Description:** This value type is a structured value consisting of a list of one or more recurrence grammar parts. Each rule part is defined by a NAME=VALUE pair. The rule parts are separated from each other by the SEMICOLON character. The rule parts are not ordered in any particular sequence. Individual rule parts **MUST** only be specified once. Compliant applications **MUST** accept rule parts ordered in any sequence, but to ensure backward compatibility with applications that pre-date this revision of iCalendar the FREQ rule part **MUST** be the first rule part specified in a RECUR value.

The FREQ rule part identifies the type of recurrence rule. This rule part **MUST** be specified in the recurrence rule. Valid values include SECONDLY, to specify repeating events based on an interval of a second or more; MINUTELY, to specify repeating events based on an interval of a minute or more; HOURLY, to specify repeating events based on an interval of an hour or more; DAILY, to specify repeating events based on an interval of a day or more; WEEKLY, to specify repeating events based on an interval of a week or more; MONTHLY, to specify repeating events based on an interval of a month or more; and YEARLY, to specify repeating events based on an interval of a year or more.



The INTERVAL rule part contains a positive integer representing at which intervals the recurrence rule repeats. The default value is "1", meaning every second for a SECONDLY rule, every minute for a MINUTELY rule, every hour for an HOURLY rule, every day for a DAILY rule, every week for a WEEKLY rule, every month for a MONTHLY rule, and every year for a YEARLY rule. For example, within a DAILY rule, a value of "8" means every eight days.

The UNTIL rule part defines a DATE or DATE-TIME value that bounds the recurrence rule in an inclusive manner. If the value specified by UNTIL is synchronized with the specified recurrence, this DATE or DATE-TIME becomes the last instance of the recurrence. The value of the UNTIL rule part MUST have the same value type as the "DTSTART" property. Furthermore, if the "DTSTART" property is specified as a date with local time, then the UNTIL rule part MUST also be specified as a date with local time. If the "DTSTART" property is specified as a date with UTC time or a date with local time and time zone reference, then the UNTIL rule part MUST be specified as a date with UTC time. In the case of the "STANDARD" and "DAYLIGHT" sub-components the UNTIL rule part MUST always be specified as a date with UTC time. If specified as a DATE-TIME value, then it MUST be specified in a UTC time format. If not present, and the COUNT rule part is also not present, the "RRULE" is considered to repeat forever.

The COUNT rule part defines the number of occurrences at which to range-bound the recurrence. The "DTSTART" property value always counts as the first occurrence.

The BYSECOND rule part specifies a COMMA-separated list of seconds within a minute. Valid values are 0 to 60. The BYMINUTE rule part specifies a COMMA-separated list of minutes within an hour. Valid values are 0 to 59. The BYHOUR rule part specifies a COMMA-separated list of hours of the day. Valid values are 0 to 23. The BYSECOND, BYMINUTE and BYHOUR rule parts MUST NOT be specified when the associated "DTSTART" property has a DATE value type. These rule parts MUST be ignored in RECUR value that violate the above requirement (e.g., generated by applications that pre-date this revision of iCalendar).

The BYDAY rule part specifies a COMMA-separated list of days of the week; SU indicates Sunday; MO indicates Monday; TU indicates Tuesday; WE indicates Wednesday; TH indicates Thursday; FR indicates Friday; and SA indicates Saturday.

Each BYDAY value can also be preceded by a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of a specific day within the MONTHLY or YEARLY "RRULE".

For example, within a MONTHLY rule, +1MO (or simply 1MO) represents the first Monday within the month, whereas -1MO represents the last Monday of the month. The numeric value in a BYDAY rule part with the FREQ rule part set to YEARLY corresponds to an offset within the month when the BYMONTH rule part is present, and corresponds to an offset within the year when the BYWEEKNO or BYMONTH rule parts are present. If an integer modifier is not present, it means all days of this type within the specified frequency. For example, within a MONTHLY rule, MO represents all Mondays within the month. The BYDAY rule part MUST NOT be specified with a numeric value when the FREQ rule part is not set to MONTHLY or YEARLY. Furthermore, the BYDAY rule part MUST NOT be specified with a numeric value with the FREQ rule part set to YEARLY when the BYWEEKNO rule part is specified.

The BYMONTHDAY rule part specifies a COMMA-separated list of days of the month. Valid values are 1 to 31 or -31 to -1. For example, -10 represents the tenth to the last day of the month. The BYMONTHDAY rule part MUST NOT be specified when the FREQ rule part is set to WEEKLY.

The BYYEARDAY rule part specifies a COMMA-separated list of days of the year. Valid values are 1 to 366 or -366 to -1. For example, -1 represents the last day of the year (December 31st) and -306 represents the 306th to the last day of the year (March 1st). The BYYEARDAY rule part MUST NOT be specified when the FREQ rule part is set to DAILY, WEEKLY, or MONTHLY.

The BYWEEKNO rule part specifies a COMMA-separated list of ordinals specifying weeks of the year. Valid values are 1 to 53 or -53 to -1. This corresponds to weeks according to week numbering as defined in [ISO.8601.2004]. A week is defined as a seven day period, starting on the day of the week defined to be the week start (see WKST). Week number one of the calendar year is the first week that contains at least four (4) days in that calendar year. This rule part MUST NOT be used when the FREQ rule part is set to anything other than YEARLY. For example, 3 represents the third week of the year.

Note: Assuming a Monday week start, week 53 can only occur when Thursday is January 1 or if it is a leap year and Wednesday is January 1.

The BYMONTH rule part specifies a COMMA-separated list of months of the year. Valid values are 1 to 12.

The WKST rule part specifies the day on which the workweek starts. Valid values are MO, TU, WE, TH, FR, SA, and SU. This is

significant when a WEEKLY "RRULE" has an interval greater than 1, and a BYDAY rule part is specified. This is also significant when in a YEARLY "RRULE" when a BYWEEKNO rule part is specified. The default value is MO.

The BYSETPOS rule part specifies a COMMA-separated list of values that corresponds to the nth occurrence within the set of recurrence instances specified by the rule. BYSETPOS operates on a set of recurrence instances in one interval of the recurrence rule. For example, in a WEEKLY rule, the interval would be one week. A set of recurrence instances starts at the beginning of the interval defined by the FREQ rule part. Valid values are 1 to 366 or -366 to -1. It MUST only be used in conjunction with another BYxxx rule part. For example "the last work day of the month" could be represented as:

```
FREQ=MONTHLY;BYDAY=MO,TU,WE,TH,FR;BYSETPOS=-1
```

Each BYSETPOS value can include a positive (+n) or negative (-n) integer. If present, this indicates the nth occurrence of the specific occurrence within the set of occurrences specified by the rule.

Recurrence rules may generate recurrence instances with an invalid date (e.g., February 30) or nonexistent local time (e.g., 1:30 AM on a day where the local time is moved forward by an hour at 1:00 AM). Such recurrence instances MUST be ignored and MUST NOT be counted as part of the recurrence set.

Information, not contained in the rule, necessary to determine the various recurrence instance start time and dates are derived from the Start Time ("DTSTART") component attribute. For example, "FREQ=YEARLY;BYMONTH=1" doesn't specify a specific day within the month or a time. This information would be the same as what is specified for "DTSTART".

BYxxx rule parts modify the recurrence in some manner. BYxxx rule parts for a period of time that is the same or greater than the frequency generally reduce or limit the number of occurrences of the recurrence generated. For example, "FREQ=DAILY;BYMONTH=1" reduces the number of recurrence instances from all days (if BYMONTH rule part is not present) to all days in January. BYxxx rule parts for a period of time less than the frequency generally increase or expand the number of occurrences of the recurrence. For example, "FREQ=YEARLY;BYMONTH=1,2" increases the number of days within the yearly recurrence set from 1 (if BYMONTH rule part is not present) to 2.

If multiple BYxxx rule parts are specified, then after evaluating the specified FREQ and INTERVAL rule parts, the BYxxx rule parts are applied to the current set of evaluated occurrences in the following order: BYMONTH, BYWEEKNO, BYYEARDAY, BYMONTHDAY, BYDAY, BYHOUR, BYMINUTE, BYSECOND and BYSETPOS; then COUNT and UNTIL are evaluated.

The table below summarizes the dependency of BYxxx rule part expand or limit behavior on the FREQ rule part value.

The term "N/A" means that the corresponding BYxxx rule part MUST NOT be used with the corresponding FREQ value.

BYDAY has some special behavior depending on the FREQ value and this is described in separate notes below the table.

	SECONDLY	MINUTELY	HOURLY	DAILY	WEEKLY	MONTHLY	YEARLY
BYMONTH	Limit	Limit	Limit	Limit	Limit	Limit	Expand
BYWEEKNO	N/A	N/A	N/A	N/A	N/A	N/A	Expand
BYYEARDAY	Limit	Limit	Limit	N/A	N/A	N/A	Expand
BYMONTHDAY	Limit	Limit	Limit	Limit	N/A	Expand	Expand
BYDAY	Limit	Limit	Limit	Limit	Expand	Note 1	Note 2
BYHOUR	Limit	Limit	Limit	Expand	Expand	Expand	Expand
BYMINUTE	Limit	Limit	Expand	Expand	Expand	Expand	Expand
BYSECOND	Limit	Expand	Expand	Expand	Expand	Expand	Expand
BYSETPOS	Limit	Limit	Limit	Limit	Limit	Limit	Limit

Note 1: Limit if BYMONTHDAY is present; otherwise, special expand for MONTHLY.

Note 2: Limit if BYYEARDAY or BYMONTHDAY is present; otherwise, special expand for WEEKLY if BYWEEKNO present; otherwise, special expand for MONTHLY if BYMONTH present; otherwise, special expand for YEARLY.

Here is an example of evaluating multiple BYxxx rule parts.

```
DTSTART;TZID=America/New_York:19970105T083000
RRULE:FREQ=YEARLY;INTERVAL=2;BYMONTH=1;BYDAY=SU;BYHOUR=8,9;
BYMINUTE=30
```

First, the "INTERVAL=2" would be applied to "FREQ=YEARLY" to arrive at "every other year". Then, "BYMONTH=1" would be applied to arrive at "every January, every other year". Then, "BYDAY=SU" would be applied to arrive at "every Sunday in January, every other year". Then, "BYHOUR=8,9" would be applied to arrive at "every Sunday in January at 8 AM and 9 AM, every other year". Then, "BYMINUTE=30" would be applied to arrive at "every Sunday in January at 8:30 AM and 9:30 AM, every other year". Then, lacking information from "RRULE", the second is derived from "DTSTART", to end up in "every Sunday in January at 8:30:00 AM and 9:30:00 AM, every other year". Similarly, if the BYMINUTE, BYHOUR, BYDAY, BYMONTHDAY, or BYMONTH rule part were missing, the appropriate minute, hour, day, or month would have been retrieved from the "DTSTART" property.

If the computed local start time of a recurrence instance does not exist, or occurs more than once, for the specified time zone, the time of the recurrence instance is interpreted in the same manner as an explicit DATE-TIME value describing that date and time, as specified in Section 3.3.5.

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

Example: The following is a rule that specifies 10 occurrences that occur every other day:

```
FREQ=DAILY;COUNT=10;INTERVAL=2
```

There are other examples specified in Section 3.8.5.3.

### 3.3.11. Text

Value Name: TEXT

Purpose: This value type is used to identify values that contain human-readable text.

Format Definition: This value type is defined by the following notation:

```

text          = *(TSAFE-CHAR / ":" / DQUOTE / ESCAPED-CHAR)
               ; Folded according to description above

ESCAPED-CHAR = ("\" / "\\;" / "\", " / "\N" / "\n")
               ; \\ encodes \, \N or \n encodes newline
               ; \; encodes ;, \, encodes ,

TSAFE-CHAR = WSP / %x21 / %x23-2B / %x2D-39 / %x3C-5B /
              %x5D-7E / NON-US-ASCII
               ; Any character except CONTROLS not needed by the current
               ; character set, DQUOTE, ";", ":", "\", ", "

```

**Description:** If the property permits, multiple TEXT values are specified by a COMMA-separated list of values.

The language in which the text is represented can be controlled by the "LANGUAGE" property parameter.

An intentional formatted text line break **MUST** only be included in a "TEXT" property value by representing the line break with the character sequence of BACKSLASH, followed by a LATIN SMALL LETTER N or a LATIN CAPITAL LETTER N, that is "\n" or "\N".

The "TEXT" property values may also contain special characters that are used to signify delimiters, such as a COMMA character for lists of values or a SEMICOLON character for structured values. In order to support the inclusion of these special characters in "TEXT" property values, they **MUST** be escaped with a BACKSLASH character. A BACKSLASH character in a "TEXT" property value **MUST** be escaped with another BACKSLASH character. A COMMA character in a "TEXT" property value **MUST** be escaped with a BACKSLASH character. A SEMICOLON character in a "TEXT" property value **MUST** be escaped with a BACKSLASH character. However, a COLON character in a "TEXT" property value **SHALL NOT** be escaped with a BACKSLASH character.

**Example:** A multiple line value of:

```

Project XYZ Final Review
Conference Room - 3B
Come Prepared.

```

would be represented as:

```

Project XYZ Final Review\nConference Room - 3B\nCome Prepared.

```

**3.3.12. Time**

Value Name: TIME

Purpose: This value type is used to identify values that contain a time of day.

Format Definition: This value type is defined by the following notation:

```
time                = time-hour time-minute time-second [time-utc]

time-hour           = 2DIGIT           ;00-23
time-minute         = 2DIGIT           ;00-59
time-second         = 2DIGIT           ;00-60
;The "60" value is used to account for positive "leap" seconds.

time-utc            = "Z"
```

Description: If the property permits, multiple "time" values are specified by a COMMA-separated list of values. No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

The "TIME" value type is used to identify values that contain a time of day. The format is based on the [ISO.8601.2004] complete representation, basic format for a time of day. The text format consists of a two-digit, 24-hour of the day (i.e., values 00-23), two-digit minute in the hour (i.e., values 00-59), and two-digit seconds in the minute (i.e., values 00-60). The seconds value of 60 MUST only be used to account for positive "leap" seconds. Fractions of a second are not supported by this format.

In parallel to the "DATE-TIME" definition above, the "TIME" value type expresses time values in three forms:

The form of time with UTC offset MUST NOT be used. For example, the following is not valid for a time value:

```
230000-0800          ;Invalid time format
```

**FORM #1 LOCAL TIME**

The local time form is simply a time value that does not contain the UTC designator nor does it reference a time zone. For example, 11:00 PM:

```
230000
```

Time values of this type are said to be "floating" and are not bound to any time zone in particular. They are used to represent the same hour, minute, and second value regardless of which time zone is currently being observed. For example, an event can be defined that indicates that an individual will be busy from 11:00 AM to 1:00 PM every day, no matter which time zone the person is in. In these cases, a local time can be specified. The recipient of an iCalendar object with a property value consisting of a local time, without any relative time zone information, SHOULD interpret the value as being fixed to whatever time zone the "ATTENDEE" is in at any given moment. This means that two "Attendees", may participate in the same event at different UTC times; floating time SHOULD only be used where that is reasonable behavior.

In most cases, a fixed time is desired. To properly communicate a fixed time in a property value, either UTC time or local time with time zone reference MUST be specified.

The use of local time in a TIME value without the "TZID" property parameter is to be interpreted as floating time, regardless of the existence of "VTIMEZONE" calendar components in the iCalendar object.

#### FORM #2: UTC TIME

UTC time, or absolute time, is identified by a LATIN CAPITAL LETTER Z suffix character, the UTC designator, appended to the time value. For example, the following represents 07:00 AM UTC:

070000Z

The "TZID" property parameter MUST NOT be applied to TIME properties whose time values are specified in UTC.

#### FORM #3: LOCAL TIME AND TIME ZONE REFERENCE

The local time with reference to time zone information form is identified by the use the "TZID" property parameter to reference the appropriate time zone definition. "TZID" is discussed in detail in Section 3.2.19.

Example: The following represents 8:30 AM in New York in winter, five hours behind UTC, in each of the three formats:

083000

133000Z

TZID=America/New\_York:083000



### 3.3.13. URI

Value Name: URI

Purpose: This value type is used to identify values that contain a uniform resource identifier (URI) type of reference to the property value.

Format Definition: This value type is defined by the following notation:

uri = <As defined in Section 3 of [RFC3986]>

Description: This value type might be used to reference binary information, for values that are large, or otherwise undesirable to include directly in the iCalendar object.

Property values with this value type MUST follow the generic URI syntax defined in [RFC3986].

When a property parameter value is a URI value type, the URI MUST be specified as a quoted-string value.

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

Example: The following is a URI for a network file:

http://example.com/my-report.txt

### 3.3.14. UTC Offset

Value Name: UTC-OFFSET

Purpose: This value type is used to identify properties that contain an offset from UTC to local time.

Format Definition: This value type is defined by the following notation:

utc-offset = time-numzone

time-numzone = ("+" / "-") time-hour time-minute [time-second]

Description: The PLUS SIGN character MUST be specified for positive UTC offsets (i.e., ahead of UTC). The HYPHEN-MINUS character MUST be specified for negative UTC offsets (i.e., behind of UTC). The

value of "-0000" and "-000000" are not allowed. The time-second, if present, MUST NOT be 60; if absent, it defaults to zero.

No additional content value encoding (i.e., BACKSLASH character encoding, see Section 3.3.11) is defined for this value type.

**Example:** The following UTC offsets are given for standard time for New York (five hours behind UTC) and Geneva (one hour ahead of UTC):

-0500

+0100

### 3.4. iCalendar Object

The Calendaring and Scheduling Core Object is a collection of calendaring and scheduling information. Typically, this information will consist of an iCalendar stream with a single iCalendar object. However, multiple iCalendar objects can be sequentially grouped together in an iCalendar stream. The first line and last line of the iCalendar object MUST contain a pair of iCalendar object delimiter strings. The syntax for an iCalendar stream is as follows:

```
icalstream = 1*icalobject
```

```
icalobject = "BEGIN" ":" "VCALENDAR" CRLF
             icalbody
             "END" ":" "VCALENDAR" CRLF
```

The following is a simple example of an iCalendar object:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
BEGIN:VEVENT
UID:19970610T172345Z-AF23B2@example.com
DTSTAMP:19970610T172345Z
DTSTART:19970714T170000Z
DTEND:19970715T040000Z
SUMMARY:Bastille Day Party
END:VEVENT
END:VCALENDAR
```

### 3.5. Property

A property is the definition of an individual attribute describing a calendar object or a calendar component. A property takes the form defined by the "contentline" notation defined in Section 3.1.

The following is an example of a property:

```
DTSTART:19960415T133000Z
```

This memo imposes no ordering of properties within an iCalendar object.

Property names, parameter names, and enumerated parameter values are case-insensitive. For example, the property name "DUE" is the same as "due" and "Due", DTSTART;TZID=America/New\_York:19980714T120000 is the same as DtStart;TzID=America/New\_York:19980714T120000.

### 3.6. Calendar Components

The body of the iCalendar object consists of a sequence of calendar properties and one or more calendar components. The calendar properties are attributes that apply to the calendar object as a whole. The calendar components are collections of properties that express a particular calendar semantic. For example, the calendar component can specify an event, a to-do, a journal entry, time zone information, free/busy time information, or an alarm.

The body of the iCalendar object is defined by the following notation:

```
icalbody    = calprops component

calprops    = *(
                ;
                ; The following are REQUIRED,
                ; but MUST NOT occur more than once.
                ;
                prodid / version /
                ;
                ; The following are OPTIONAL,
                ; but MUST NOT occur more than once.
                ;
                calscale / method /
                ;
                ; The following are OPTIONAL,
                ; and MAY occur more than once.
                ;
            )
```

```

        x-prop / iana-prop
        ;
    )

    component = 1*(eventc / todoc / journalc / freebusyc /
        timezonec / iana-comp / x-comp)

    iana-comp = "BEGIN" ":" iana-token CRLF
        1*contentline
        "END" ":" iana-token CRLF

    x-comp    = "BEGIN" ":" x-name CRLF
        1*contentline
        "END" ":" x-name CRLF

```

An iCalendar object MUST include the "PRODID" and "VERSION" calendar properties. In addition, it MUST include at least one calendar component. Special forms of iCalendar objects are possible to publish just busy time (i.e., only a "VFREEBUSY" calendar component) or time zone (i.e., only a "VTIMEZONE" calendar component) information. In addition, a complex iCalendar object that is used to capture a complete snapshot of the contents of a calendar is possible (e.g., composite of many different calendar components). More commonly, an iCalendar object will consist of just a single "VEVENT", "VTODO", or "VJOURNAL" calendar component. Applications MUST ignore x-comp and iana-comp values they don't recognize. Applications that support importing iCalendar objects SHOULD support all of the component types defined in this document, and SHOULD NOT silently drop any components as that can lead to user data loss.

### 3.6.1. Event Component

Component Name: VEVENT

Purpose: Provide a grouping of component properties that describe an event.

Format Definition: A "VEVENT" calendar component is defined by the following notation:

```

    eventc    = "BEGIN" ":" "VEVENT" CRLF
        eventprop *alarmc
        "END" ":" "VEVENT" CRLF

    eventprop = *(
        ;
        ; The following are REQUIRED,
        ; but MUST NOT occur more than once.

```

```

;
dtstamp / uid /
;
; The following is REQUIRED if the component
; appears in an iCalendar object that doesn't
; specify the "METHOD" property; otherwise, it
; is OPTIONAL; in any case, it MUST NOT occur
; more than once.
;
dtstart /
;
; The following are OPTIONAL,
; but MUST NOT occur more than once.
;
class / created / description / geo /
last-mod / location / organizer / priority /
seq / status / summary / transp /
url / recurid /
;
; The following is OPTIONAL,
; but SHOULD NOT occur more than once.
;
rrule /
;
; Either 'dtend' or 'duration' MAY appear in
; a 'eventprop', but 'dtend' and 'duration'
; MUST NOT occur in the same 'eventprop'.
;
dtend / duration /
;
; The following are OPTIONAL,
; and MAY occur more than once.
;
attach / attendee / categories / comment /
contact / exdate / rstatus / related /
resources / rdate / x-prop / iana-prop
;
)

```

**Description:** A "VEVENT" calendar component is a grouping of component properties, possibly including "VALARM" calendar components, that represents a scheduled amount of time on a calendar. For example, it can be an activity; such as a one-hour long, department meeting from 8:00 AM to 9:00 AM, tomorrow. Generally, an event will take up time on an individual calendar. Hence, the event will appear as an opaque interval in a search for busy time. Alternately, the event can have its Time Transparency

set to "TRANSPARENT" in order to prevent blocking of the event in searches for busy time.

The "VEVENT" is also the calendar component used to specify an anniversary or daily reminder within a calendar. These events have a DATE value type for the "DTSTART" property instead of the default value type of DATE-TIME. If such a "VEVENT" has a "DTEND" property, it MUST be specified as a DATE value also. The anniversary type of "VEVENT" can span more than one date (i.e., "DTEND" property value is set to a calendar date after the "DTSTART" property value). If such a "VEVENT" has a "DURATION" property, it MUST be specified as a "dur-day" or "dur-week" value.

The "DTSTART" property for a "VEVENT" specifies the inclusive start of the event. For recurring events, it also specifies the very first instance in the recurrence set. The "DTEND" property for a "VEVENT" calendar component specifies the non-inclusive end of the event. For cases where a "VEVENT" calendar component specifies a "DTSTART" property with a DATE value type but no "DTEND" nor "DURATION" property, the event's duration is taken to be one day. For cases where a "VEVENT" calendar component specifies a "DTSTART" property with a DATE-TIME value type but no "DTEND" property, the event ends on the same calendar date and time of day specified by the "DTSTART" property.

The "VEVENT" calendar component cannot be nested within another calendar component. However, "VEVENT" calendar components can be related to each other or to a "VTODO" or to a "VJOURNAL" calendar component with the "RELATED-TO" property.

**Example:** The following is an example of the "VEVENT" calendar component used to represent a meeting that will also be opaque to searches for busy time:

```
BEGIN:VEVENT
UID:19970901T130000Z-123401@example.com
DTSTAMP:19970901T130000Z
DTSTART:19970903T163000Z
DTEND:19970903T190000Z
SUMMARY:Annual Employee Review
CLASS:PRIVATE
CATEGORIES:BUSINESS,HUMAN RESOURCES
END:VEVENT
```

The following is an example of the "VEVENT" calendar component used to represent a reminder that will not be opaque, but rather transparent, to searches for busy time:

```
BEGIN:VEVENT
UID:19970901T130000Z-123402@example.com
DTSTAMP:19970901T130000Z
DTSTART:19970401T163000Z
DTEND:19970402T010000Z
SUMMARY:Laurel is in sensitivity awareness class.
CLASS:PUBLIC
CATEGORIES:BUSINESS,HUMAN RESOURCES
TRANSP:TRANSPARENT
END:VEVENT
```

The following is an example of the "VEVENT" calendar component used to represent an anniversary that will occur annually:

```
BEGIN:VEVENT
UID:19970901T130000Z-123403@example.com
DTSTAMP:19970901T130000Z
DTSTART;VALUE=DATE:19971102
SUMMARY:Our Blissful Anniversary
TRANSP:TRANSPARENT
CLASS:CONFIDENTIAL
CATEGORIES:ANNIVERSARY,PERSONAL,SPECIAL OCCASION
RRULE:FREQ=YEARLY
END:VEVENT
```

The following is an example of the "VEVENT" calendar component used to represent a multi-day event scheduled from June 28th, 2007 to July 8th, 2007 inclusively. Note that the "DTEND" property is set to July 9th, 2007, since the "DTEND" property specifies the non-inclusive end of the event.

```
BEGIN:VEVENT
UID:20070423T123432Z-541111@example.com
DTSTAMP:20070423T123432Z
DTSTART;VALUE=DATE:20070628
DTEND;VALUE=DATE:20070709
SUMMARY:Festival International de Jazz de Montreal
TRANSP:TRANSPARENT
END:VEVENT
```

### 3.6.2. To-Do Component

Component Name: VTOD

Purpose: Provide a grouping of calendar properties that describe a to-do.

**Format Definition:** A "VTODO" calendar component is defined by the following notation:

```

todoc      = "BEGIN" ":" "VTODO" CRLF
            todoprop *alarmc
            "END" ":" "VTODO" CRLF

todoprop   = *(
            ;
            ; The following are REQUIRED,
            ; but MUST NOT occur more than once.
            ;
            dtstamp / uid /
            ;
            ; The following are OPTIONAL,
            ; but MUST NOT occur more than once.
            ;
            class / completed / created / description /
            dtstart / geo / last-mod / location / organizer /
            percent / priority / recurid / seq / status /
            summary / url /
            ;
            ; The following is OPTIONAL,
            ; but SHOULD NOT occur more than once.
            ;
            rrule /
            ;
            ; Either 'due' or 'duration' MAY appear in
            ; a 'todoprop', but 'due' and 'duration'
            ; MUST NOT occur in the same 'todoprop'.
            ; If 'duration' appear in a 'todoprop',
            ; then 'dtstart' MUST also appear in
            ; the same 'todoprop'.
            ;
            due / duration /
            ;
            ; The following are OPTIONAL,
            ; and MAY occur more than once.
            ;
            attach / attendee / categories / comment / contact /
            exdate / rstatus / related / resources /
            rdate / x-prop / iana-prop
            ;
            )

```

**Description:** A "VTODO" calendar component is a grouping of component properties and possibly "VALARM" calendar components that represent an action-item or assignment. For example, it can be



used to represent an item of work assigned to an individual; such as "turn in travel expense today".

The "VTODO" calendar component cannot be nested within another calendar component. However, "VTODO" calendar components can be related to each other or to a "VEVENT" or to a "VJOURNAL" calendar component with the "RELATED-TO" property.

A "VTODO" calendar component without the "DTSTART" and "DUE" (or "DURATION") properties specifies a to-do that will be associated with each successive calendar date, until it is completed.

**Examples:** The following is an example of a "VTODO" calendar component that needs to be completed before May 1st, 2007. On midnight May 1st, 2007 this to-do would be considered overdue.

```
BEGIN:VTODO
UID:20070313T123432Z-456553@example.com
DTSTAMP:20070313T123432Z
DUE;VALUE=DATE:20070501
SUMMARY:Submit Quebec Income Tax Return for 2006
CLASS:CONFIDENTIAL
CATEGORIES:FAMILY,FINANCE
STATUS:NEEDS-ACTION
END:VTODO
```

The following is an example of a "VTODO" calendar component that was due before 1:00 P.M. UTC on July 9th, 2007 and was completed on July 7th, 2007 at 10:00 A.M. UTC.

```
BEGIN:VTODO
UID:20070514T103211Z-123404@example.com
DTSTAMP:20070514T103211Z
DTSTART:20070514T110000Z
DUE:20070709T130000Z
COMPLETED:20070707T100000Z
SUMMARY:Submit Revised Internet-Draft
PRIORITY:1
STATUS:NEEDS-ACTION
END:VTODO
```

### 3.6.3. Journal Component

**Component Name:** VJOURNAL

**Purpose:** Provide a grouping of component properties that describe a journal entry.

**Format Definition:** A "VJOURNAL" calendar component is defined by the following notation:

```
journalc    = "BEGIN" ":" "VJOURNAL" CRLF
              jourprop
              "END" ":" "VJOURNAL" CRLF

jourprop    = *(
              ;
              ; The following are REQUIRED,
              ; but MUST NOT occur more than once.
              ;
              dtstamp / uid /
              ;
              ; The following are OPTIONAL,
              ; but MUST NOT occur more than once.
              ;
              class / created / dtstart /
              last-mod / organizer / recurid / seq /
              status / summary / url /
              ;
              ; The following is OPTIONAL,
              ; but SHOULD NOT occur more than once.
              ;
              rrule /
              ;
              ; The following are OPTIONAL,
              ; and MAY occur more than once.
              ;
              attach / attendee / categories / comment /
              contact / description / exdate / related / rdate /
              rstatus / x-prop / iana-prop
              ;
              )
```

**Description:** A "VJOURNAL" calendar component is a grouping of component properties that represent one or more descriptive text notes associated with a particular calendar date. The "DTSTART" property is used to specify the calendar date with which the journal entry is associated. Generally, it will have a DATE value data type, but it can also be used to specify a DATE-TIME value data type. Examples of a journal entry include a daily record of a legislative body or a journal entry of individual telephone contacts for the day or an ordered list of accomplishments for the day. The "VJOURNAL" calendar component can also be used to associate a document with a calendar date.

The "VJOURNAL" calendar component does not take up time on a calendar. Hence, it does not play a role in free or busy time searches -- it is as though it has a time transparency value of TRANSPARENT. It is transparent to any such searches.

The "VJOURNAL" calendar component cannot be nested within another calendar component. However, "VJOURNAL" calendar components can be related to each other or to a "VEVENT" or to a "VTODO" calendar component, with the "RELATED-TO" property.

Example: The following is an example of the "VJOURNAL" calendar component:

```
BEGIN:VJOURNAL
UID:19970901T130000Z-123405@example.com
DTSTAMP:19970901T130000Z
DTSTART;VALUE=DATE:19970317
SUMMARY:Staff meeting minutes
DESCRIPTION:1. Staff meeting: Participants include Joe\,
  Lisa\, and Bob. Aurora project plans were reviewed.
  There is currently no budget reserves for this project.
  Lisa will escalate to management. Next meeting on Tuesday.\n
2. Telephone Conference: ABC Corp. sales representative
  called to discuss new printer. Promised to get us a demo by
  Friday.\n3. Henry Miller (Handsoff Insurance): Car was
  totaled by tree. Is looking into a loaner car. 555-2323
  (tel).
END:VJOURNAL
```

#### 3.6.4. Free/Busy Component

Component Name: VFREEBUSY

Purpose: Provide a grouping of component properties that describe either a request for free/busy time, describe a response to a request for free/busy time, or describe a published set of busy time.

Format Definition: A "VFREEBUSY" calendar component is defined by the following notation:

```
freebusyc = "BEGIN" ":" "VFREEBUSY" CRLF
           fbprop
           "END" ":" "VFREEBUSY" CRLF

fbprop    = *(
           ;
           ; The following are REQUIRED,
```

```

; but MUST NOT occur more than once.
;
dtstamp / uid /
;
; The following are OPTIONAL,
; but MUST NOT occur more than once.
;
contact / dtstart / dtend /
organizer / url /
;
; The following are OPTIONAL,
; and MAY occur more than once.
;
attendee / comment / freebusy / rstatus / x-prop /
iana-prop
;
)

```

**Description:** A "VFREEBUSY" calendar component is a grouping of component properties that represents either a request for free or busy time information, a reply to a request for free or busy time information, or a published set of busy time information.

When used to request free/busy time information, the "ATTENDEE" property specifies the calendar users whose free/busy time is being requested; the "ORGANIZER" property specifies the calendar user who is requesting the free/busy time; the "DTSTART" and "DTEND" properties specify the window of time for which the free/busy time is being requested; the "UID" and "DTSTAMP" properties are specified to assist in proper sequencing of multiple free/busy time requests.

When used to reply to a request for free/busy time, the "ATTENDEE" property specifies the calendar user responding to the free/busy time request; the "ORGANIZER" property specifies the calendar user that originally requested the free/busy time; the "FREEBUSY" property specifies the free/busy time information (if it exists); and the "UID" and "DTSTAMP" properties are specified to assist in proper sequencing of multiple free/busy time replies.

When used to publish busy time, the "ORGANIZER" property specifies the calendar user associated with the published busy time; the "DTSTART" and "DTEND" properties specify an inclusive time window that surrounds the busy time information; the "FREEBUSY" property specifies the published busy time information; and the "DTSTAMP" property specifies the DATE-TIME that iCalendar object was created.

The "VFREEBUSY" calendar component cannot be nested within another calendar component. Multiple "VFREEBUSY" calendar components can be specified within an iCalendar object. This permits the grouping of free/busy information into logical collections, such as monthly groups of busy time information.

The "VFREEBUSY" calendar component is intended for use in iCalendar object methods involving requests for free time, requests for busy time, requests for both free and busy, and the associated replies.

Free/Busy information is represented with the "FREEBUSY" property. This property provides a terse representation of time periods. One or more "FREEBUSY" properties can be specified in the "VFREEBUSY" calendar component.

When present in a "VFREEBUSY" calendar component, the "DTSTART" and "DTEND" properties SHOULD be specified prior to any "FREEBUSY" properties.

The recurrence properties ("RRULE", "RDATE", "EXDATE") are not permitted within a "VFREEBUSY" calendar component. Any recurring events are resolved into their individual busy time periods using the "FREEBUSY" property.

Example: The following is an example of a "VFREEBUSY" calendar component used to request free or busy time information:

```
BEGIN:VFREEBUSY
UID:19970901T082949Z-FA43EF@example.com
ORGANIZER:mailto:jane_doe@example.com
ATTENDEE:mailto:john_public@example.com
DTSTART:19971015T050000Z
DTEND:19971016T050000Z
DTSTAMP:19970901T083000Z
END:VFREEBUSY
```

The following is an example of a "VFREEBUSY" calendar component used to reply to the request with busy time information:

```
BEGIN:VFREEBUSY
UID:19970901T095957Z-76A912@example.com
ORGANIZER:mailto:jane_doe@example.com
ATTENDEE:mailto:john_public@example.com
DTSTAMP:19970901T100000Z
FREEBUSY:19971015T050000Z/PT8H30M,
  19971015T160000Z/PT5H30M,19971015T223000Z/PT6H30M
URL:http://example.com/pub/busy/jpublic-01.ifb
COMMENT:This iCalendar file contains busy time information for
  the next three months.
END:VFREEBUSY
```

The following is an example of a "VFREEBUSY" calendar component used to publish busy time information:

```
BEGIN:VFREEBUSY
UID:19970901T115957Z-76A912@example.com
DTSTAMP:19970901T120000Z
ORGANIZER:jsmith@example.com
DTSTART:19980313T141711Z
DTEND:19980410T141711Z
FREEBUSY:19980314T233000Z/19980315T003000Z
FREEBUSY:19980316T153000Z/19980316T163000Z
FREEBUSY:19980318T030000Z/19980318T040000Z
URL:http://www.example.com/calendar/busytime/jsmith.ifb
END:VFREEBUSY
```

### 3.6.5. Time Zone Component

Component Name: VTIMEZONE

Purpose: Provide a grouping of component properties that defines a time zone.

Format Definition: A "VTIMEZONE" calendar component is defined by the following notation:

```
timezonec = "BEGIN" ":" "VTIMEZONE" CRLF
           *(
             ;
             ; 'tzid' is REQUIRED, but MUST NOT occur more
             ; than once.
             ;
             tzid /
             ;
           )
```

```

; 'last-mod' and 'tzurl' are OPTIONAL,
; but MUST NOT occur more than once.
;
last-mod / tzurl /
;
; One of 'standardc' or 'daylightc' MUST occur
; and each MAY occur more than once.
;
standardc / daylightc /
;
; The following are OPTIONAL,
; and MAY occur more than once.
;
x-prop / iana-prop
)
"END" ":" "VTIMEZONE" CRLF

standardc = "BEGIN" ":" "STANDARD" CRLF
           tzprop
           "END" ":" "STANDARD" CRLF

daylightc = "BEGIN" ":" "DAYLIGHT" CRLF
            tzprop
            "END" ":" "DAYLIGHT" CRLF

tzprop    = *(
;
; The following are REQUIRED,
; but MUST NOT occur more than once.
;
dtstart / tzoffsetto / tzoffsetfrom /
;
; The following is OPTIONAL,
; but SHOULD NOT occur more than once.
;
rrule /
;
; The following are OPTIONAL,
; and MAY occur more than once.
;
comment / rdate / tzname / x-prop / iana-prop
;
)

```

**Description:** A time zone is unambiguously defined by the set of time measurement rules determined by the governing body for a given geographic area. These rules describe, at a minimum, the base

offset from UTC for the time zone, often referred to as the Standard Time offset. Many locations adjust their Standard Time forward or backward by one hour, in order to accommodate seasonal changes in number of daylight hours, often referred to as Daylight Saving Time. Some locations adjust their time by a fraction of an hour. Standard Time is also known as Winter Time. Daylight Saving Time is also known as Advanced Time, Summer Time, or Legal Time in certain countries. The following table shows the changes in time zone rules in effect for New York City starting from 1967. Each line represents a description or rule for a particular observance.

Effective Observance Rule

Date	(Date-Time)	Offset	Abbreviation
1967-1973	last Sun in Apr, 02:00	-0400	EDT
1967-2006	last Sun in Oct, 02:00	-0500	EST
1974-1974	Jan 6, 02:00	-0400	EDT
1975-1975	Feb 23, 02:00	-0400	EDT
1976-1986	last Sun in Apr, 02:00	-0400	EDT
1987-2006	first Sun in Apr, 02:00	-0400	EDT
2007-*	second Sun in Mar, 02:00	-0400	EDT
2007-*	first Sun in Nov, 02:00	-0500	EST

Note: The specification of a global time zone registry is not addressed by this document and is left for future study. However, implementers may find the TZ database [TZDB] a useful reference. It is an informal, public-domain collection of time zone information, which is currently being maintained by volunteer Internet participants, and is used in several operating systems. This database contains current and historical time zone information for a wide variety of locations around the globe; it provides a time zone identifier for every unique time zone rule set in actual use since 1970, with historical data going back to the introduction of standard time.



Interoperability between two calendaring and scheduling applications, especially for recurring events, to-dos or journal entries, is dependent on the ability to capture and convey date and time information in an unambiguous format. The specification of current time zone information is integral to this behavior.

If present, the "VTIMEZONE" calendar component defines the set of Standard Time and Daylight Saving Time observances (or rules) for a particular time zone for a given interval of time. The "VTIMEZONE" calendar component cannot be nested within other calendar components. Multiple "VTIMEZONE" calendar components can exist in an iCalendar object. In this situation, each "VTIMEZONE" MUST represent a unique time zone definition. This is necessary for some classes of events, such as airline flights, that start in one time zone and end in another.

The "VTIMEZONE" calendar component MUST include the "TZID" property and at least one definition of a "STANDARD" or "DAYLIGHT" sub-component. The "STANDARD" or "DAYLIGHT" sub-component MUST include the "DTSTART", "TZOFFSETFROM", and "TZOFFSETTO" properties.

An individual "VTIMEZONE" calendar component MUST be specified for each unique "TZID" parameter value specified in the iCalendar object. In addition, a "VTIMEZONE" calendar component, referred to by a recurring calendar component, MUST provide valid time zone information for all recurrence instances.

Each "VTIMEZONE" calendar component consists of a collection of one or more sub-components that describe the rule for a particular observance (either a Standard Time or a Daylight Saving Time observance). The "STANDARD" sub-component consists of a collection of properties that describe Standard Time. The "DAYLIGHT" sub-component consists of a collection of properties that describe Daylight Saving Time. In general, this collection of properties consists of:

- \* the first onset DATE-TIME for the observance;
- \* the last onset DATE-TIME for the observance, if a last onset is known;
- \* the offset to be applied for the observance;
- \* a rule that describes the day and time when the observance takes effect;
- \* an optional name for the observance.

For a given time zone, there may be multiple unique definitions of the observances over a period of time. Each observance is described using either a "STANDARD" or "DAYLIGHT" sub-component. The collection of these sub-components is used to describe the time zone for a given period of time. The offset to apply at any given time is found by locating the observance that has the last onset date and time before the time in question, and using the offset value from that observance.

The top-level properties in a "VTIMEZONE" calendar component are:

The mandatory "TZID" property is a text value that uniquely identifies the "VTIMEZONE" calendar component within the scope of an iCalendar object.

The optional "LAST-MODIFIED" property is a UTC value that specifies the date and time that this time zone definition was last updated.

The optional "TZURL" property is a url value that points to a published "VTIMEZONE" definition. "TZURL" SHOULD refer to a resource that is accessible by anyone who might need to interpret the object. This SHOULD NOT normally be a "file" URL or other URL that is not widely accessible.

The collection of properties that are used to define the "STANDARD" and "DAYLIGHT" sub-components include:

The mandatory "DTSTART" property gives the effective onset date and local time for the time zone sub-component definition. "DTSTART" in this usage MUST be specified as a date with a local time value.

The mandatory "TZOFFSETFROM" property gives the UTC offset that is in use when the onset of this time zone observance begins. "TZOFFSETFROM" is combined with "DTSTART" to define the effective onset for the time zone sub-component definition. For example, the following represents the time at which the observance of Standard Time took effect in Fall 1967 for New York City:

DTSTART:19671029T020000

TZOFFSETFROM:-0400

The mandatory "TZOFFSETTO" property gives the UTC offset for the time zone sub-component (Standard Time or Daylight Saving Time) when this observance is in use.

The optional "TZNAME" property is the customary name for the time zone. This could be used for displaying dates.

The onset DATE-TIME values for the observance defined by the time zone sub-component is defined by the "DTSTART", "RRULE", and "RDATE" properties.

The "RRULE" property defines the recurrence rule for the onset of the observance defined by this time zone sub-component. Some specific requirements for the usage of "RRULE" for this purpose include:

- \* If observance is known to have an effective end date, the "UNTIL" recurrence rule parameter MUST be used to specify the last valid onset of this observance (i.e., the UNTIL DATE-TIME will be equal to the last instance generated by the recurrence pattern). It MUST be specified in UTC time.
- \* The "DTSTART" and the "TZOFFSETFROM" properties MUST be used when generating the onset DATE-TIME values (instances) from the "RRULE".

The "RDATE" property can also be used to define the onset of the observance by giving the individual onset date and times. "RDATE" in this usage MUST be specified as a date with local time value, relative to the UTC offset specified in the "TZOFFSETFROM" property.

The optional "COMMENT" property is also allowed for descriptive explanatory text.

Example: The following are examples of the "VTIMEZONE" calendar component:

This is an example showing all the time zone rules for New York City since April 30, 1967 at 03:00:00 EDT.

```
BEGIN:VTIMEZONE
TZID:America/New_York
LAST-MODIFIED:20050809T050000Z
BEGIN:DAYLIGHT
DTSTART:19670430T020000
RRULE:FREQ=YEARLY;BYMONTH=4;BYDAY=-1SU;UNTIL=19730429T070000Z
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
BEGIN:STANDARD
```

```
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYMONTH=10;BYDAY=-1SU;UNTIL=20061029T060000Z
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19740106T020000
RDATE:19750223T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
BEGIN:DAYLIGHT
DTSTART:19760425T020000
RRULE:FREQ=YEARLY;BYMONTH=4;BYDAY=-1SU;UNTIL=19860427T070000Z
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYMONTH=4;BYDAY=1SU;UNTIL=20060402T070000Z
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
BEGIN:DAYLIGHT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
BEGIN:STANDARD
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
END:VTIMEZONE
```

This is an example showing time zone information for New York City using only the "DTSTART" property. Note that this is only suitable for a recurring event that starts on or later than March 11, 2007 at 03:00:00 EDT (i.e., the earliest effective transition date and time) and ends no later than March 9, 2008 at 01:59:59

EST (i.e., latest valid date and time for EST in this scenario). For example, this can be used for a recurring event that occurs every Friday, 8:00 A.M.-9:00 A.M., starting June 1, 2007, ending December 31, 2007,

```
BEGIN:VTIMEZONE
TZID:America/New_York
LAST-MODIFIED:20050809T050000Z
BEGIN:STANDARD
DTSTART:20071104T020000
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:20070311T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
END:VTIMEZONE
```

This is a simple example showing the current time zone rules for New York City using a "RRULE" recurrence pattern. Note that there is no effective end date to either of the Standard Time or Daylight Time rules. This information would be valid for a recurring event starting today and continuing indefinitely.

```
BEGIN:VTIMEZONE
TZID:America/New_York
LAST-MODIFIED:20050809T050000Z
TZURL:http://zones.example.com/tz/America-New_York.ics
BEGIN:STANDARD
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
END:VTIMEZONE
```

This is an example showing a set of rules for a fictitious time zone where the Daylight Time rule has an effective end date (i.e., after that date, Daylight Time is no longer observed).

```
BEGIN:VTIMEZONE
TZID:Fictitious
LAST-MODIFIED:19870101T000000Z
BEGIN:STANDARD
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4;UNTIL=19980404T070000Z
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
END:VTIMEZONE
```

This is an example showing a set of rules for a fictitious time zone where the first Daylight Time rule has an effective end date. There is a second Daylight Time rule that picks up where the other left off.

```
BEGIN:VTIMEZONE
TZID:Fictitious
LAST-MODIFIED:19870101T000000Z
BEGIN:STANDARD
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4;UNTIL=19980404T070000Z
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
BEGIN:DAYLIGHT
DTSTART:19990424T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=4
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:EDT
END:DAYLIGHT
END:VTIMEZONE
```

### 3.6.6. Alarm Component

Component Name: VALARM

Purpose: Provide a grouping of component properties that define an alarm.

Format Definition: A "VALARM" calendar component is defined by the following notation:

```
alarmc      = "BEGIN" ":" "VALARM" CRLF
              (audioprop / dispprop / emailprop)
              "END" ":" "VALARM" CRLF

audioprop   = *(
              ;
              ; 'action' and 'trigger' are both REQUIRED,
```

```

; but MUST NOT occur more than once.
;
action / trigger /
;
; 'duration' and 'repeat' are both OPTIONAL,
; and MUST NOT occur more than once each;
; but if one occurs, so MUST the other.
;
duration / repeat /
;
; The following is OPTIONAL,
; but MUST NOT occur more than once.
;
attach /
;
; The following is OPTIONAL,
; and MAY occur more than once.
;
x-prop / iana-prop
;
)

dispprop = *(
;
; The following are REQUIRED,
; but MUST NOT occur more than once.
;
action / description / trigger /
;
; 'duration' and 'repeat' are both OPTIONAL,
; and MUST NOT occur more than once each;
; but if one occurs, so MUST the other.
;
duration / repeat /
;
; The following is OPTIONAL,
; and MAY occur more than once.
;
x-prop / iana-prop
;
)

emailprop = *(
;
; The following are all REQUIRED,
; but MUST NOT occur more than once.
;
action / description / trigger / summary /

```



```

;
; The following is REQUIRED,
; and MAY occur more than once.
;
attendee /
;
; 'duration' and 'repeat' are both OPTIONAL,
; and MUST NOT occur more than once each;
; but if one occurs, so MUST the other.
;
duration / repeat /
;
; The following are OPTIONAL,
; and MAY occur more than once.
;
attach / x-prop / iana-prop
;
)

```

**Description:** A "VALARM" calendar component is a grouping of component properties that is a reminder or alarm for an event or a to-do. For example, it may be used to define a reminder for a pending event or an overdue to-do.

The "VALARM" calendar component MUST include the "ACTION" and "TRIGGER" properties. The "ACTION" property further constrains the "VALARM" calendar component in the following ways:

When the action is "AUDIO", the alarm can also include one and only one "ATTACH" property, which MUST point to a sound resource, which is rendered when the alarm is triggered.

When the action is "DISPLAY", the alarm MUST also include a "DESCRIPTION" property, which contains the text to be displayed when the alarm is triggered.

When the action is "EMAIL", the alarm MUST include a "DESCRIPTION" property, which contains the text to be used as the message body, a "SUMMARY" property, which contains the text to be used as the message subject, and one or more "ATTENDEE" properties, which contain the email address of attendees to receive the message. It can also include one or more "ATTACH" properties, which are intended to be sent as message attachments. When the alarm is triggered, the email message is sent.

The "VALARM" calendar component MUST only appear within either a "VEVENT" or "VTODO" calendar component. "VALARM" calendar components cannot be nested. Multiple mutually independent

"VALARM" calendar components can be specified for a single "VEVENT" or "VTODO" calendar component.

The "TRIGGER" property specifies when the alarm will be triggered. The "TRIGGER" property specifies a duration prior to the start of an event or a to-do. The "TRIGGER" edge may be explicitly set to be relative to the "START" or "END" of the event or to-do with the "RELATED" parameter of the "TRIGGER" property. The "TRIGGER" property value type can alternatively be set to an absolute calendar date with UTC time.

In an alarm set to trigger on the "START" of an event or to-do, the "DTSTART" property MUST be present in the associated event or to-do. In an alarm in a "VEVENT" calendar component set to trigger on the "END" of the event, either the "DTEND" property MUST be present, or the "DTSTART" and "DURATION" properties MUST both be present. In an alarm in a "VTODO" calendar component set to trigger on the "END" of the to-do, either the "DUE" property MUST be present, or the "DTSTART" and "DURATION" properties MUST both be present.

The alarm can be defined such that it triggers repeatedly. A definition of an alarm with a repeating trigger MUST include both the "DURATION" and "REPEAT" properties. The "DURATION" property specifies the delay period, after which the alarm will repeat. The "REPEAT" property specifies the number of additional repetitions that the alarm will be triggered. This repetition count is in addition to the initial triggering of the alarm. Both of these properties MUST be present in order to specify a repeating alarm. If one of these two properties is absent, then the alarm will not repeat beyond the initial trigger.

The "ACTION" property is used within the "VALARM" calendar component to specify the type of action invoked when the alarm is triggered. The "VALARM" properties provide enough information for a specific action to be invoked. It is typically the responsibility of a "Calendar User Agent" (CUA) to deliver the alarm in the specified fashion. An "ACTION" property value of AUDIO specifies an alarm that causes a sound to be played to alert the user; DISPLAY specifies an alarm that causes a text message to be displayed to the user; and EMAIL specifies an alarm that causes an electronic email message to be delivered to one or more email addresses.

In an AUDIO alarm, if the optional "ATTACH" property is included, it MUST specify an audio sound resource. The intention is that the sound will be played as the alarm effect. If an "ATTACH" property is specified that does not refer to a sound resource, or

if the specified sound resource cannot be rendered (because its format is unsupported, or because it cannot be retrieved), then the CUA or other entity responsible for playing the sound may choose a fallback action, such as playing a built-in default sound, or playing no sound at all.

In a DISPLAY alarm, the intended alarm effect is for the text value of the "DESCRIPTION" property to be displayed to the user.

In an EMAIL alarm, the intended alarm effect is for an email message to be composed and delivered to all the addresses specified by the "ATTENDEE" properties in the "VALARM" calendar component. The "DESCRIPTION" property of the "VALARM" calendar component MUST be used as the body text of the message, and the "SUMMARY" property MUST be used as the subject text. Any "ATTACH" properties in the "VALARM" calendar component SHOULD be sent as attachments to the message.

Note: Implementations should carefully consider whether they accept alarm components from untrusted sources, e.g., when importing calendar objects from external sources. One reasonable policy is to always ignore alarm components that the calendar user has not set herself, or at least ask for confirmation in such a case.

Example: The following example is for a "VALARM" calendar component that specifies an audio alarm that will sound at a precise time and repeat 4 more times at 15-minute intervals:

```
BEGIN:VALARM
TRIGGER;VALUE=DATE-TIME:19970317T133000Z
REPEAT:4
DURATION:PT15M
ACTION:AUDIO
ATTACH;FMTTYPE=audio/basic:ftp://example.com/pub/
sounds/bell-01.aud
END:VALARM
```

The following example is for a "VALARM" calendar component that specifies a display alarm that will trigger 30 minutes before the scheduled start of the event or of the to-do it is associated with and will repeat 2 more times at 15-minute intervals:

```
BEGIN:VALARM
TRIGGER:-PT30M
REPEAT:2
DURATION:PT15M
ACTION:DISPLAY
DESCRIPTION:Breakfast meeting with executive\
  team at 8:30 AM EST.
END:VALARM
```

The following example is for a "VALARM" calendar component that specifies an email alarm that will trigger 2 days before the scheduled due DATE-TIME of a to-do with which it is associated. It does not repeat. The email has a subject, body, and attachment link.

```
BEGIN:VALARM
TRIGGER;RELATED=END:-P2D
ACTION:EMAIL
ATTENDEE:mailto:john_doe@example.com
SUMMARY:*** REMINDER: SEND AGENDA FOR WEEKLY STAFF MEETING ***
DESCRIPTION:A draft agenda needs to be sent out to the attendees
  to the weekly managers meeting (MGR-LIST). Attached is a
  pointer the document template for the agenda file.
ATTACH;FMTTYPE=application/msword:http://example.com/
  templates/agenda.doc
END:VALARM
```

### 3.7. Calendar Properties

The Calendar Properties are attributes that apply to the iCalendar object, as a whole. These properties do not appear within a calendar component. They SHOULD be specified after the "BEGIN:VCALENDAR" delimiter string and prior to any calendar component.

#### 3.7.1. Calendar Scale

Property Name: CALSCALE

Purpose: This property defines the calendar scale used for the calendar information specified in the iCalendar object.

Value Type: TEXT

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified once in an iCalendar object. The default value is "GREGORIAN".

**Description:** This memo is based on the Gregorian calendar scale. The Gregorian calendar scale is assumed if this property is not specified in the iCalendar object. It is expected that other calendar scales will be defined in other specifications or by future versions of this memo.

**Format Definition:** This property is defined by the following notation:

calscale = "CALSCALE" calparam ":" calvalue CRLF

calparam = \*(";" other-param)

calvalue = "GREGORIAN"

**Example:** The following is an example of this property:

CALSCALE:GREGORIAN

### 3.7.2. Method

**Property Name:** METHOD

**Purpose:** This property defines the iCalendar object method associated with the calendar object.

**Value Type:** TEXT

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** This property can be specified once in an iCalendar object.

**Description:** When used in a MIME message entity, the value of this property MUST be the same as the Content-Type "method" parameter value. If either the "METHOD" property or the Content-Type "method" parameter is specified, then the other MUST also be specified.

No methods are defined by this specification. This is the subject of other specifications, such as the iCalendar Transport-independent Interoperability Protocol (iTIP) defined by [2446bis].

If this property is not present in the iCalendar object, then a scheduling transaction MUST NOT be assumed. In such cases, the iCalendar object is merely being used to transport a snapshot of

some calendar information; without the intention of conveying a scheduling semantic.

**Format Definition:** This property is defined by the following notation:

```
method      = "METHOD" metparam ":" metvalue CRLF
metparam    = *(";" other-param)
metvalue    = iana-token
```

**Example:** The following is a hypothetical example of this property to convey that the iCalendar object is a scheduling request:

```
METHOD:REQUEST
```

### 3.7.3. Product Identifier

**Property Name:** PRODID

**Purpose:** This property specifies the identifier for the product that created the iCalendar object.

**Value Type:** TEXT

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** The property **MUST** be specified once in an iCalendar object.

**Description:** The vendor of the implementation **SHOULD** assure that this is a globally unique identifier; using some technique such as an FPI value, as defined in [ISO.9070.1991].

This property **SHOULD NOT** be used to alter the interpretation of an iCalendar object beyond the semantics specified in this memo. For example, it is not to be used to further the understanding of non-standard properties.

**Format Definition:** This property is defined by the following notation:

```
prodid      = "PRODID" pidparam ":" pidvalue CRLF
pidparam    = *(";" other-param)
```

pidvalue = text  
;Any text that describes the product and version  
;and that is generally assured of being unique.

Example: The following is an example of this property. It does not imply that English is the default language.

PRODID:-//ABC Corporation//NONSGML My Product//EN

#### 3.7.4. Version

Property Name: VERSION

Purpose: This property specifies the identifier corresponding to the highest version number or the minimum and maximum range of the iCalendar specification that is required in order to interpret the iCalendar object.

Value Type: TEXT

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property MUST be specified once in an iCalendar object.

Description: A value of "2.0" corresponds to this memo.

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

version = "VERSION" verparam ":" vervalue CRLF

verparam = \*(";" other-param)

vervalue = "2.0" ;This memo  
/ maxver  
/ (minver ";" maxver)

minver = <A IANA-registered iCalendar version identifier>  
;Minimum iCalendar version needed to parse the iCalendar object.

maxver = <A IANA-registered iCalendar version identifier>  
;Maximum iCalendar version needed to parse the iCalendar object.

Example: The following is an example of this property:

```
VERSION:2.0
```

### 3.8. Component Properties

The following properties can appear within calendar components, as specified by each component property definition.

#### 3.8.1. Descriptive Component Properties

The following properties specify descriptive information about calendar components.

##### 3.8.1.1. Attachment

Property Name: ATTACH

Purpose: This property provides the capability to associate a document object with a calendar component.

Value Type: The default value type for this property is URI. The value type can also be set to BINARY to indicate inline binary encoded content information.

Property Parameters: IANA, non-standard, inline encoding, and value data type property parameters can be specified on this property. The format type parameter can be specified on this property and is RECOMMENDED for inline binary encoded content information.

Conformance: This property can be specified multiple times in a "VEVENT", "VTODO", "VJOURNAL", or "VALARM" calendar component with the exception of AUDIO alarm that only allows this property to occur once.

Description: This property is used in "VEVENT", "VTODO", and "VJOURNAL" calendar components to associate a resource (e.g., document) with the calendar component. This property is used in "VALARM" calendar components to specify an audio sound resource or an email message attachment. This property can be specified as a URI pointing to a resource or as inline binary encoded content.

When this property is specified as inline binary encoded content, calendar applications MAY attempt to guess the media type of the resource via inspection of its content if and only if the media type of the resource is not given by the "FMTPYPE" parameter. If the media type remains unknown, calendar applications SHOULD treat it as type "application/octet-stream".



**Format Definition:** This property is defined by the following notation:

```
attach      = "ATTACH" attachparam ( ":" uri ) /
              (
                ";" "ENCODING" "=" "BASE64"
                ";" "VALUE" "=" "BINARY"
                ":" binary
              )
              CRLF

attachparam = *(
              ;
              ; The following is OPTIONAL for a URI value,
              ; RECOMMENDED for a BINARY value,
              ; and MUST NOT occur more than once.
              ;
              ( ";" fmttypeparam ) /
              ;
              ; The following is OPTIONAL,
              ; and MAY occur more than once.
              ;
              ( ";" other-param )
              ;
              )
```

**Example:** The following are examples of this property:

```
ATTACH:CID:jsmith.part3.960817T083000.xyzMail@example.com
```

```
ATTACH;FMPTYPE=application/postscript:ftp://example.com/pub/
reports/r-960812.ps
```

### 3.8.1.2. Categories

**Property Name:** CATEGORIES

**Purpose:** This property defines the categories for a calendar component.

**Value Type:** TEXT

**Property Parameters:** IANA, non-standard, and language property parameters can be specified on this property.

**Conformance:** The property can be specified within "VEVENT", "VTOD", or "VJOURNAL" calendar components.

**Description:** This property is used to specify categories or subtypes of the calendar component. The categories are useful in searching for a calendar component of a particular type and category. Within the "VEVENT", "VTODO", or "VJOURNAL" calendar components, more than one category can be specified as a COMMA-separated list of categories.

**Format Definition:** This property is defined by the following notation:

```
categories = "CATEGORIES" catparam ":" text *("," text)
              CRLF
```

```
catparam    = *(
              ;
              ; The following is OPTIONAL,
              ; but MUST NOT occur more than once.
              ;
              (";" languageparam ) /
              ;
              ; The following is OPTIONAL,
              ; and MAY occur more than once.
              ;
              (";" other-param)
              ;
              )
```

**Example:** The following are examples of this property:

```
CATEGORIES:APPOINTMENT,EDUCATION
```

```
CATEGORIES:MEETING
```

### 3.8.1.3. Classification

**Property Name:** CLASS

**Purpose:** This property defines the access classification for a calendar component.

**Value Type:** TEXT

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** The property can be specified once in a "VEVENT", "VTODO", or "VJOURNAL" calendar components.

**Description:** An access classification is only one component of the general security system within a calendar application. It provides a method of capturing the scope of the access the calendar owner intends for information within an individual calendar entry. The access classification of an individual iCalendar component is useful when measured along with the other security components of a calendar system (e.g., calendar user authentication, authorization, access rights, access role, etc.). Hence, the semantics of the individual access classifications cannot be completely defined by this memo alone. Additionally, due to the "blind" nature of most exchange processes using this memo, these access classifications cannot serve as an enforcement statement for a system receiving an iCalendar object. Rather, they provide a method for capturing the intention of the calendar owner for the access to the calendar component. If not specified in a component that allows this property, the default value is PUBLIC. Applications MUST treat x-name and iana-token values they don't recognize the same way as they would the PRIVATE value.

**Format Definition:** This property is defined by the following notation:

```
class      = "CLASS" classparam ":" classvalue CRLF
classparam = *(";" other-param)

classvalue = "PUBLIC" / "PRIVATE" / "CONFIDENTIAL" / iana-token
            / x-name
;Default is PUBLIC
```

**Example:** The following is an example of this property:

```
CLASS:PUBLIC
```

#### 3.8.1.4. Comment

**Property Name:** COMMENT

**Purpose:** This property specifies non-processing information intended to provide a comment to the calendar user.

**Value Type:** TEXT

**Property Parameters:** IANA, non-standard, alternate text representation, and language property parameters can be specified on this property.

**Conformance:** This property can be specified multiple times in "VEVENT", "VTODO", "VJOURNAL", and "VFREEBUSY" calendar components as well as in the "STANDARD" and "DAYLIGHT" sub-components.

**Description:** This property is used to specify a comment to the calendar user.

**Format Definition:** This property is defined by the following notation:

```
comment      = "COMMENT" commparam ":" text CRLF

commparam    = *(
                ;
                ; The following are OPTIONAL,
                ; but MUST NOT occur more than once.
                ;
                (";" altrepparam) / (";" languageparam) /
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
                ;
                (";" other-param)
                ;
                )
```

**Example:** The following is an example of this property:

```
COMMENT:The meeting really needs to include both ourselves
and the customer. We can't hold this meeting without them.
As a matter of fact\, the venue for the meeting ought to be at
their site. - - John
```

### 3.8.1.5. Description

**Property Name:** DESCRIPTION

**Purpose:** This property provides a more complete description of the calendar component than that provided by the "SUMMARY" property.

**Value Type:** TEXT

**Property Parameters:** IANA, non-standard, alternate text representation, and language property parameters can be specified on this property.

**Conformance:** The property can be specified in the "VEVENT", "VTODO", "VJOURNAL", or "VALARM" calendar components. The property can be specified multiple times only within a "VJOURNAL" calendar component.

**Description:** This property is used in the "VEVENT" and "VTODO" to capture lengthy textual descriptions associated with the activity.

This property is used in the "VJOURNAL" calendar component to capture one or more textual journal entries.

This property is used in the "VALARM" calendar component to capture the display text for a DISPLAY category of alarm, and to capture the body text for an EMAIL category of alarm.

**Format Definition:** This property is defined by the following notation:

```
description = "DESCRIPTION" descparam ":" text CRLF

descparam   = *(
                ;
                ; The following are OPTIONAL,
                ; but MUST NOT occur more than once.
                ;
                (";" altrepparam) / (";" languageparam) /
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
                ;
                (";" other-param)
                ;
                )
```

**Example:** The following is an example of this property with formatted line breaks in the property value:

```
DESCRIPTION:Meeting to provide technical review for "Phoenix"
design.\nHappy Face Conference Room. Phoenix design team
MUST attend this meeting.\nRSVP to team leader.
```

### 3.8.1.6. Geographic Position

**Property Name:** GEO

**Purpose:** This property specifies information related to the global position for the activity specified by a calendar component.

**Value Type:** FLOAT. The value **MUST** be two SEMICOLON-separated FLOAT values.

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** This property can be specified in "VEVENT" or "VTODO" calendar components.

**Description:** This property value specifies latitude and longitude, in that order (i.e., "LAT LON" ordering). The longitude represents the location east or west of the prime meridian as a positive or negative real number, respectively. The longitude and latitude values **MAY** be specified up to six decimal places, which will allow for accuracy to within one meter of geographical position. Receiving applications **MUST** accept values of this precision and **MAY** truncate values of greater precision.

Values for latitude and longitude shall be expressed as decimal fractions of degrees. Whole degrees of latitude shall be represented by a two-digit decimal number ranging from 0 through 90. Whole degrees of longitude shall be represented by a decimal number ranging from 0 through 180. When a decimal fraction of a degree is specified, it shall be separated from the whole number of degrees by a decimal point.

Latitudes north of the equator shall be specified by a plus sign (+), or by the absence of a minus sign (-), preceding the digits designating degrees. Latitudes south of the Equator shall be designated by a minus sign (-) preceding the digits designating degrees. A point on the Equator shall be assigned to the Northern Hemisphere.

Longitudes east of the prime meridian shall be specified by a plus sign (+), or by the absence of a minus sign (-), preceding the digits designating degrees. Longitudes west of the meridian shall be designated by minus sign (-) preceding the digits designating degrees. A point on the prime meridian shall be assigned to the Eastern Hemisphere. A point on the 180th meridian shall be assigned to the Western Hemisphere. One exception to this last convention is permitted. For the special condition of describing a band of latitude around the earth, the East Bounding Coordinate data element shall be assigned the value +180 (180) degrees.

Any spatial address with a latitude of +90 (90) or -90 degrees will specify the position at the North or South Pole, respectively. The component for longitude may have any legal value.

With the exception of the special condition described above, this form is specified in [ANSI INCITS 61-1986].

The simple formula for converting degrees-minutes-seconds into decimal degrees is:

$$\text{decimal} = \text{degrees} + \text{minutes}/60 + \text{seconds}/3600.$$

**Format Definition:** This property is defined by the following notation:

geo = "GEO" geoparam ":" geovalue CRLF

geoparam = \*(";" other-param)

geovalue = float ";" float  
;Latitude and Longitude components

**Example:** The following is an example of this property:

GEO:37.386013;-122.082932

### 3.8.1.7. Location

**Property Name:** LOCATION

**Purpose:** This property defines the intended venue for the activity defined by a calendar component.

**Value Type:** TEXT

**Property Parameters:** IANA, non-standard, alternate text representation, and language property parameters can be specified on this property.

**Conformance:** This property can be specified in "VEVENT" or "VTODO" calendar component.

**Description:** Specific venues such as conference or meeting rooms may be explicitly specified using this property. An alternate representation may be specified that is a URI that points to directory information with more structured specification of the location. For example, the alternate representation may specify either an LDAP URL [RFC4516] pointing to an LDAP server entry or a CID URL [RFC2392] pointing to a MIME body part containing a Virtual-Information Card (vCard) [RFC2426] for the location.

**Format Definition:** This property is defined by the following notation:

```
location    = "LOCATION" locparam ":" text CRLF

locparam    = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" altrepparam) / (";" languageparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )
```

**Example:** The following are some examples of this property:

```
LOCATION:Conference Room - F123\, Bldg. 002
```

```
LOCATION;ALTREP="http://xyzcorp.com/conf-rooms/f123.vcf":
Conference Room - F123\, Bldg. 002
```

### 3.8.1.8. Percent Complete

**Property Name:** PERCENT-COMPLETE

**Purpose:** This property is used by an assignee or delegatee of a to-do to convey the percent completion of a to-do to the "Organizer".

**Value Type:** INTEGER

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** This property can be specified once in a "VTODO" calendar component.

**Description:** The property value is a positive integer between 0 and 100. A value of "0" indicates the to-do has not yet been started. A value of "100" indicates that the to-do has been completed. Integer values in between indicate the percent partially complete.



When a to-do is assigned to multiple individuals, the property value indicates the percent complete for that portion of the to-do assigned to the assignee or delegatee. For example, if a to-do is assigned to both individuals "A" and "B". A reply from "A" with a percent complete of "70" indicates that "A" has completed 70% of the to-do assigned to them. A reply from "B" with a percent complete of "50" indicates "B" has completed 50% of the to-do assigned to them.

**Format Definition:** This property is defined by the following notation:

percent = "PERCENT-COMPLETE" pctparam ":" integer CRLF

pctparam = \*(";" other-param)

**Example:** The following is an example of this property to show 39% completion:

PERCENT-COMPLETE:39

#### 3.8.1.9. Priority

**Property Name:** PRIORITY

**Purpose:** This property defines the relative priority for a calendar component.

**Value Type:** INTEGER

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** This property can be specified in "VEVENT" and "VTODO" calendar components.

**Description:** This priority is specified as an integer in the range 0 to 9. A value of 0 specifies an undefined priority. A value of 1 is the highest priority. A value of 2 is the second highest priority. Subsequent numbers specify a decreasing ordinal priority. A value of 9 is the lowest priority.

A CUA with a three-level priority scheme of "HIGH", "MEDIUM", and "LOW" is mapped into this property such that a property value in the range of 1 to 4 specifies "HIGH" priority. A value of 5 is the normal or "MEDIUM" priority. A value in the range of 6 to 9 is "LOW" priority.

A CUA with a priority schema of "A1", "A2", "A3", "B1", "B2", ..., "C3" is mapped into this property such that a property value of 1 specifies "A1", a property value of 2 specifies "A2", a property value of 3 specifies "A3", and so forth up to a property value of 9 specifies "C3".

Other integer values are reserved for future use.

Within a "VEVENT" calendar component, this property specifies a priority for the event. This property may be useful when more than one event is scheduled for a given time period.

Within a "VTODO" calendar component, this property specified a priority for the to-do. This property is useful in prioritizing multiple action items for a given time period.

**Format Definition:** This property is defined by the following notation:

```
priority    = "PRIORITY" prioparam ":" priovalue CRLF
;Default is zero (i.e., undefined).
```

```
prioparam   = *(";" other-param)
```

```
priovalue   = integer           ;Must be in the range [0..9]
; All other values are reserved for future use.
```

**Example:** The following is an example of a property with the highest priority:

```
PRIORITY:1
```

The following is an example of a property with a next highest priority:

```
PRIORITY:2
```

The following is an example of a property with no priority. This is equivalent to not specifying the "PRIORITY" property:

```
PRIORITY:0
```

### 3.8.1.10. Resources

Property Name: RESOURCES

Purpose: This property defines the equipment or resources anticipated for an activity specified by a calendar component.

Value Type: TEXT

Property Parameters: IANA, non-standard, alternate text representation, and language property parameters can be specified on this property.

Conformance: This property can be specified once in "VEVENT" or "VTODO" calendar component.

Description: The property value is an arbitrary text. More than one resource can be specified as a COMMA-separated list of resources.

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

```
resources = "RESOURCES" resrcparam ":" text *("," text) CRLF

resrcparam = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" altrepparam) / (";" languageparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
)
```

Example: The following is an example of this property:

```
RESOURCES:EASEL,PROJECTOR,VCR
```

```
RESOURCES;LANGUAGE=fr:Nettoyeur haute pression
```

**3.8.1.11. Status**

Property Name: STATUS

Purpose: This property defines the overall status or confirmation for the calendar component.

Value Type: TEXT

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified once in "VEVENT", "VTODO", or "VJOURNAL" calendar components.

Description: In a group-scheduled calendar component, the property is used by the "Organizer" to provide a confirmation of the event to the "Attendees". For example in a "VEVENT" calendar component, the "Organizer" can indicate that a meeting is tentative, confirmed, or cancelled. In a "VTODO" calendar component, the "Organizer" can indicate that an action item needs action, is completed, is in process or being worked on, or has been cancelled. In a "VJOURNAL" calendar component, the "Organizer" can indicate that a journal entry is draft, final, or has been cancelled or removed.

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

```

status          = "STATUS" statparam ":" statvalue CRLF

statparam       = *(";" other-param)

statvalue       = (statvalue-event
                  /  statvalue-todo
                  /  statvalue-jour)

statvalue-event = "TENTATIVE"      ;Indicates event is tentative.
                  / "CONFIRMED"    ;Indicates event is definite.
                  / "CANCELLED"    ;Indicates event was cancelled.
;Status values for a "VEVENT"

statvalue-todo  = "NEEDS-ACTION"   ;Indicates to-do needs action.
                  / "COMPLETED"   ;Indicates to-do completed.
                  / "IN-PROCESS"   ;Indicates to-do in process of.
                  / "CANCELLED"    ;Indicates to-do was cancelled.
;Status values for "VTODO".
```

```
statvalue-jour = "DRAFT"           ;Indicates journal is draft.
                / "FINAL"           ;Indicates journal is final.
                / "CANCELLED"        ;Indicates journal is removed.
;Status values for "VJOURNAL".
```

Example: The following is an example of this property for a "VEVENT" calendar component:

```
STATUS:TENTATIVE
```

The following is an example of this property for a "VTOD" calendar component:

```
STATUS:NEEDS-ACTION
```

The following is an example of this property for a "VJOURNAL" calendar component:

```
STATUS:DRAFT
```

#### 3.8.1.12. Summary

Property Name: SUMMARY

Purpose: This property defines a short summary or subject for the calendar component.

Value Type: TEXT

Property Parameters: IANA, non-standard, alternate text representation, and language property parameters can be specified on this property.

Conformance: The property can be specified in "VEVENT", "VTOD", "VJOURNAL", or "VALARM" calendar components.

Description: This property is used in the "VEVENT", "VTOD", and "VJOURNAL" calendar components to capture a short, one-line summary about the activity or journal entry.

This property is used in the "VALARM" calendar component to capture the subject of an EMAIL category of alarm.

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

```

summary      = "SUMMARY" summparam ":" text CRLF
summparam    = *(
                ;
                ; The following are OPTIONAL,
                ; but MUST NOT occur more than once.
                ;
                (";" altrepparam) / (";" languageparam) /
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
                ;
                (";" other-param)
                ;
                )

```

Example: The following is an example of this property:

```
SUMMARY:Department Party
```

### 3.8.2. Date and Time Component Properties

The following properties specify date and time related information in calendar components.

#### 3.8.2.1. Date-Time Completed

Property Name: COMPLETED

Purpose: This property defines the date and time that a to-do was actually completed.

Value Type: DATE-TIME

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: The property can be specified in a "VTODO" calendar component. The value MUST be specified as a date with UTC time.

Description: This property defines the date and time that a to-do was actually completed.

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

completed = "COMPLETED" compparam ":" date-time CRLF

compparam = \*(";" other-param)

Example: The following is an example of this property:

COMPLETED:19960401T150000Z

### 3.8.2.2. Date-Time End

Property Name: DTEND

Purpose: This property specifies the date and time that a calendar component ends.

Value Type: The default value type is DATE-TIME. The value type can be set to a DATE value type.

Property Parameters: IANA, non-standard, value data type, and time zone identifier property parameters can be specified on this property.

Conformance: This property can be specified in "VEVENT" or "VFREEBUSY" calendar components.

Description: Within the "VEVENT" calendar component, this property defines the date and time by which the event ends. The value type of this property MUST be the same as the "DTSTART" property, and its value MUST be later in time than the value of the "DTSTART" property. Furthermore, this property MUST be specified as a date with local time if and only if the "DTSTART" property is also specified as a date with local time.

Within the "VFREEBUSY" calendar component, this property defines the end date and time for the free or busy time information. The time MUST be specified in the UTC time format. The value MUST be later in time than the value of the "DTSTART" property.

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

```

dtend      = "DTEND" dtendparam ":" dtendval CRLF

dtendparam = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" "VALUE" "=" ("DATE-TIME" / "DATE")) /
    (";" tzidparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
)

dtendval   = date-time / date
;Value MUST match value type

```

Example: The following is an example of this property:

```
DTEND:19960401T150000Z
```

```
DTEND;VALUE=DATE:19980704
```

### 3.8.2.3. Date-Time Due

Property Name: DUE

Purpose: This property defines the date and time that a to-do is expected to be completed.

Value Type: The default value type is DATE-TIME. The value type can be set to a DATE value type.

Property Parameters: IANA, non-standard, value data type, and time zone identifier property parameters can be specified on this property.

Conformance: The property can be specified once in a "VTODO" calendar component.

Description: This property defines the date and time before which a to-do is expected to be completed. For cases where this property is specified in a "VTODO" calendar component that also specifies a "DTSTART" property, the value type of this property MUST be the same as the "DTSTART" property, and the value of this property



MUST be later in time than the value of the "DTSTART" property. Furthermore, this property MUST be specified as a date with local time if and only if the "DTSTART" property is also specified as a date with local time.

**Format Definition:** This property is defined by the following notation:

```

due          = "DUE" dueparam ":" dueval CRLF

dueparam     = *(
                ;
                ; The following are OPTIONAL,
                ; but MUST NOT occur more than once.
                ;
                (";" "VALUE" "=" ("DATE-TIME" / "DATE")) /
                (";" tzidparam) /
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
                ;
                (";" other-param)
                ;
                )

dueval       = date-time / date
;Value MUST match value type

```

**Example:** The following is an example of this property:

```
DUE:19980430T000000Z
```

#### 3.8.2.4. Date-Time Start

**Property Name:** DTSTART

**Purpose:** This property specifies when the calendar component begins.

**Value Type:** The default value type is DATE-TIME. The time value MUST be one of the forms defined for the DATE-TIME value type. The value type can be set to a DATE value type.

**Property Parameters:** IANA, non-standard, value data type, and time zone identifier property parameters can be specified on this property.

**Conformance:** This property can be specified once in the "VEVENT", "VTODO", or "VFREEBUSY" calendar components as well as in the

"STANDARD" and "DAYLIGHT" sub-components. This property is REQUIRED in all types of recurring calendar components that specify the "RRULE" property. This property is also REQUIRED in "VEVENT" calendar components contained in iCalendar objects that don't specify the "METHOD" property.

**Description:** Within the "VEVENT" calendar component, this property defines the start date and time for the event.

Within the "VFREEBUSY" calendar component, this property defines the start date and time for the free or busy time information. The time MUST be specified in UTC time.

Within the "STANDARD" and "DAYLIGHT" sub-components, this property defines the effective start date and time for a time zone specification. This property is REQUIRED within each "STANDARD" and "DAYLIGHT" sub-components included in "VTIMEZONE" calendar components and MUST be specified as a date with local time without the "TZID" property parameter.

**Format Definition:** This property is defined by the following notation:

```
dtstart      = "DTSTART" dtstparam ":" dtstval CRLF

dtstparam    = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" "VALUE" "=" ("DATE-TIME" / "DATE")) /
    (";" tzidparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )

dtstval      = date-time / date
;Value MUST match value type
```

**Example:** The following is an example of this property:

```
DTSTART:19980118T073000Z
```

### 3.8.2.5. Duration

Property Name: DURATION

Purpose: This property specifies a positive duration of time.

Value Type: DURATION

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified in "VEVENT", "VTODO", or "VALARM" calendar components.

Description: In a "VEVENT" calendar component the property may be used to specify a duration of the event, instead of an explicit end DATE-TIME. In a "VTODO" calendar component the property may be used to specify a duration for the to-do, instead of an explicit due DATE-TIME. In a "VALARM" calendar component the property may be used to specify the delay period prior to repeating an alarm. When the "DURATION" property relates to a "DTSTART" property that is specified as a DATE value, then the "DURATION" property MUST be specified as a "dur-day" or "dur-week" value.

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

duration = "DURATION" durparam ":" dur-value CRLF  
;consisting of a positive duration of time.

durparam = \*(";" other-param)

Example: The following is an example of this property that specifies an interval of time of one hour and zero minutes and zero seconds:

DURATION:PT1H0M0S

The following is an example of this property that specifies an interval of time of 15 minutes.

DURATION:PT15M

### 3.8.2.6. Free/Busy Time

Property Name: FREEBUSY

Purpose: This property defines one or more free or busy time intervals.

Value Type: PERIOD

Property Parameters: IANA, non-standard, and free/busy time type property parameters can be specified on this property.

Conformance: The property can be specified in a "VFREEBUSY" calendar component.

Description: These time periods can be specified as either a start and end DATE-TIME or a start DATE-TIME and DURATION. The date and time MUST be a UTC time format.

"FREEBUSY" properties within the "VFREEBUSY" calendar component SHOULD be sorted in ascending order, based on start time and then end time, with the earliest periods first.

The "FREEBUSY" property can specify more than one value, separated by the COMMA character. In such cases, the "FREEBUSY" property values MUST all be of the same "FBTYPE" property parameter type (e.g., all values of a particular "FBTYPE" listed together in a single property).

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

```
freebusy    = "FREEBUSY" fbparam ":" fbvalue CRLF
fbparam     = *(
    ;
    ; The following is OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" fbtypeparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )
```

fbvalue = period \*("," period)  
 ;Time value MUST be in the UTC time format.

Example: The following are some examples of this property:

FREEBUSY;FBTYPE=BUSY-UNAVAILABLE:19970308T160000Z/PT8H30M

FREEBUSY;FBTYPE=FREE:19970308T160000Z/PT3H,19970308T200000Z/PT1H

FREEBUSY;FBTYPE=FREE:19970308T160000Z/PT3H,19970308T200000Z/PT1H  
 ,19970308T230000Z/19970309T000000Z

### 3.8.2.7. Time Transparency

Property Name: TRANSP

Purpose: This property defines whether or not an event is transparent to busy time searches.

Value Type: TEXT

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified once in a "VEVENT" calendar component.

Description: Time Transparency is the characteristic of an event that determines whether it appears to consume time on a calendar. Events that consume actual time for the individual or resource associated with the calendar SHOULD be recorded as OPAQUE, allowing them to be detected by free/busy time searches. Other events, which do not take up the individual's (or resource's) time SHOULD be recorded as TRANSPARENT, making them invisible to free/busy time searches.

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

transp = "TRANSP" transparam ":" transvalue CRLF

transparam = \*(";" other-param)

transvalue = "OPAQUE"  
 ;Blocks or opaque on busy time searches.  
 / "TRANSPARENT"  
 ;Transparent on busy time searches.  
 ;Default value is OPAQUE

**Example:** The following is an example of this property for an event that is transparent or does not block on free/busy time searches:

TRANSP:TRANSPARENT

The following is an example of this property for an event that is opaque or blocks on free/busy time searches:

TRANSP:OPAQUE

### 3.8.3. Time Zone Component Properties

The following properties specify time zone information in calendar components.

#### 3.8.3.1. Time Zone Identifier

**Property Name:** TZID

**Purpose:** This property specifies the text value that uniquely identifies the "VTIMEZONE" calendar component in the scope of an iCalendar object.

**Value Type:** TEXT

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** This property **MUST** be specified in a "VTIMEZONE" calendar component.

**Description:** This is the label by which a time zone calendar component is referenced by any iCalendar properties whose value type is either DATE-TIME or TIME and not intended to specify a UTC or a "floating" time. The presence of the SOLIDUS character as a prefix, indicates that this "TZID" represents a unique ID in a globally defined time zone registry (when such registry is defined).

**Note:** This document does not define a naming convention for time zone identifiers. Implementers may want to use the naming conventions defined in existing time zone specifications such as the public-domain TZ database [TZDB]. The specification of globally unique time zone identifiers is not addressed by this document and is left for future study.

**Format Definition:** This property is defined by the following notation:

```
tzid          = "TZID" tzidpropparam ":" [tzidprefix] text CRLF
tzidpropparam = *(";" other-param)
;tzidprefix   = "/"
; Defined previously. Just listed here for reader convenience.
```

**Example:** The following are examples of non-globally unique time zone identifiers:

TZID:America/New\_York

TZID:America/Los\_Angeles

The following is an example of a fictitious globally unique time zone identifier:

TZID:/example.org/America/New\_York

### 3.8.3.2. Time Zone Name

**Property Name:** TZNAME

**Purpose:** This property specifies the customary designation for a time zone description.

**Value Type:** TEXT

**Property Parameters:** IANA, non-standard, and language property parameters can be specified on this property.

**Conformance:** This property can be specified in "STANDARD" and "DAYLIGHT" sub-components.

**Description:** This property specifies a customary name that can be used when displaying dates that occur during the observance defined by the time zone sub-component.

**Format Definition:** This property is defined by the following notation:

```

tzname      = "TZNAME" tznparam ":" text CRLF
tznparam    = *(
    ;
    ; The following is OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" languageparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )

```

Example: The following are examples of this property:

TZNAME:EST

TZNAME;LANGUAGE=fr-CA:HNE

### 3.8.3.3. Time Zone Offset From

Property Name: TZOFFSETFROM

Purpose: This property specifies the offset that is in use prior to this time zone observance.

Value Type: UTC-OFFSET

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property MUST be specified in "STANDARD" and "DAYLIGHT" sub-components.

Description: This property specifies the offset that is in use prior to this time observance. It is used to calculate the absolute time at which the transition to a given observance takes place. This property MUST only be specified in a "VTIMEZONE" calendar component. A "VTIMEZONE" calendar component MUST include this property. The property value is a signed numeric indicating the number of hours and possibly minutes from UTC. Positive numbers represent time zones east of the prime meridian, or ahead of UTC. Negative numbers represent time zones west of the prime meridian, or behind UTC.



**Format Definition:** This property is defined by the following notation:

```
tzoffsetfrom      = "TZOFFSETFROM" frmparam ":" utc-offset
                  CRLF
```

```
frmparam         = *(";" other-param)
```

**Example:** The following are examples of this property:

```
TZOFFSETFROM:-0500
```

```
TZOFFSETFROM:+1345
```

#### **3.8.3.4. Time Zone Offset To**

**Property Name:** TZOFFSETTO

**Purpose:** This property specifies the offset that is in use in this time zone observance.

**Value Type:** UTC-OFFSET

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** This property MUST be specified in "STANDARD" and "DAYLIGHT" sub-components.

**Description:** This property specifies the offset that is in use in this time zone observance. It is used to calculate the absolute time for the new observance. The property value is a signed numeric indicating the number of hours and possibly minutes from UTC. Positive numbers represent time zones east of the prime meridian, or ahead of UTC. Negative numbers represent time zones west of the prime meridian, or behind UTC.

**Format Definition:** This property is defined by the following notation:

```
tzoffsetto = "TZOFFSETTO" toparam ":" utc-offset CRLF
```

```
toparam    = *(";" other-param)
```

Example: The following are examples of this property:

TZOFFSETTO:-0400

TZOFFSETTO:+1245

### 3.8.3.5. Time Zone URL

Property Name: TZURL

Purpose: This property provides a means for a "VTIMEZONE" component to point to a network location that can be used to retrieve an up-to-date version of itself.

Value Type: URI

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified in a "VTIMEZONE" calendar component.

Description: This property provides a means for a "VTIMEZONE" component to point to a network location that can be used to retrieve an up-to-date version of itself. This provides a hook to handle changes government bodies impose upon time zone definitions. Retrieval of this resource results in an iCalendar object containing a single "VTIMEZONE" component and a "METHOD" property set to PUBLISH.

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

tzurl = "TZURL" tzurlparam ":" uri CRLF

tzurlparam = \*(";" other-param)

Example: The following is an example of this property:

TZURL:http://timezones.example.org/tz/America-Los\_Angeles.ics

### 3.8.4. Relationship Component Properties

The following properties specify relationship information in calendar components.

### 3.8.4.1. Attendee

Property Name: ATTENDEE

Purpose: This property defines an "Attendee" within a calendar component.

Value Type: CAL-ADDRESS

Property Parameters: IANA, non-standard, language, calendar user type, group or list membership, participation role, participation status, RSVP expectation, delegatee, delegator, sent by, common name, or directory entry reference property parameters can be specified on this property.

Conformance: This property MUST be specified in an iCalendar object that specifies a group-scheduled calendar entity. This property MUST NOT be specified in an iCalendar object when publishing the calendar information (e.g., NOT in an iCalendar object that specifies the publication of a calendar user's busy time, event, to-do, or journal). This property is not specified in an iCalendar object that specifies only a time zone definition or that defines calendar components that are not group-scheduled components, but are components only on a single user's calendar.

Description: This property MUST only be specified within calendar components to specify participants, non-participants, and the chair of a group-scheduled calendar entity. The property is specified within an "EMAIL" category of the "VALARM" calendar component to specify an email address that is to receive the email type of iCalendar alarm.

The property parameter "CN" is for the common or displayable name associated with the calendar address; "ROLE", for the intended role that the attendee will have in the calendar component; "PARTSTAT", for the status of the attendee's participation; "RSVP", for indicating whether the favor of a reply is requested; "CUTYPE", to indicate the type of calendar user; "MEMBER", to indicate the groups that the attendee belongs to; "DELEGATED-TO", to indicate the calendar users that the original request was delegated to; and "DELEGATED-FROM", to indicate whom the request was delegated from; "SENT-BY", to indicate whom is acting on behalf of the "ATTENDEE"; and "DIR", to indicate the URI that points to the directory information corresponding to the attendee. These property parameters can be specified on an "ATTENDEE" property in either a "VEVENT", "VTODO", or "VJOURNAL" calendar component. They MUST NOT be specified in an "ATTENDEE" property in a "VFREEBUSY" or "VALARM" calendar component. If the

"LANGUAGE" property parameter is specified, the identified language applies to the "CN" parameter.

A recipient delegated a request MUST inherit the "RSVP" and "ROLE" values from the attendee that delegated the request to them.

Multiple attendees can be specified by including multiple "ATTENDEE" properties within the calendar component.

**Format Definition:** This property is defined by the following notation:

```
attendee    = "ATTENDEE" attparam ":" cal-address CRLF
attparam    = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" cutypeparam) / (";" memberparam) /
    (";" roleparam) / (";" partstatparam) /
    (";" rsvpparam) / (";" deltoparam) /
    (";" delfromparam) / (";" sentbyparam) /
    (";" cnparam) / (";" dirparam) /
    (";" languageparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )
```

**Example:** The following are examples of this property's use for a to-do:

```
ATTENDEE;MEMBER="mailto:DEV-GROUP@example.com":
  mailto:joecool@example.com
ATTENDEE;DELEGATED-FROM="mailto:immud@example.com":
  mailto:ildoit@example.com
```

The following is an example of this property used for specifying multiple attendees to an event:

```
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=TENTATIVE;CN=Henry
Cabot:mailto:hcabot@example.com
ATTENDEE;ROLE=REQ-PARTICIPANT;DELEGATED-FROM="mailto:bob@
example.com";PARTSTAT=ACCEPTED;CN=Jane Doe:mailto:jdoe@
example.com
```

The following is an example of this property with a URI to the directory information associated with the attendee:

```
ATTENDEE;CN=John Smith;DIR="ldap://example.com:6666/o=ABC%
20Industries,c=US???(cn=Jim%20Dolittle)":mailto:jimdo@
example.com
```

The following is an example of this property with "delegatee" and "delegator" information for an event:

```
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=TENTATIVE;DELEGATED-FROM=
"mailto:iamboss@example.com";CN=Henry Cabot:mailto:hcabot@
example.com
ATTENDEE;ROLE=NON-PARTICIPANT;PARTSTAT=DELEGATED;DELEGATED-TO=
"mailto:hcabot@example.com";CN=The Big Cheese:mailto:iamboss
@example.com
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=ACCEPTED;CN=Jane Doe
:mailto:jdoe@example.com
```

Example: The following is an example of this property's use when another calendar user is acting on behalf of the "Attendee":

```
ATTENDEE;SENT-BY=mailto:jan_doe@example.com;CN=John Smith:
mailto:jsmith@example.com
```

#### 3.8.4.2. Contact

Property Name: CONTACT

Purpose: This property is used to represent contact information or alternately a reference to contact information associated with the calendar component.

Value Type: TEXT

Property Parameters: IANA, non-standard, alternate text representation, and language property parameters can be specified on this property.

**Conformance:** This property can be specified in a "VEVENT", "VTODO", "VJOURNAL", or "VFREEBUSY" calendar component.

**Description:** The property value consists of textual contact information. An alternative representation for the property value can also be specified that refers to a URI pointing to an alternate form, such as a vCard [RFC2426], for the contact information.

**Format Definition:** This property is defined by the following notation:

```
contact      = "CONTACT" contparam ":" text CRLF
contparam    = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" altrepparam) / (";" languageparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )
```

**Example:** The following is an example of this property referencing textual contact information:

```
CONTACT:Jim Dolittle\, ABC Industries\, +1-919-555-1234
```

The following is an example of this property with an alternate representation of an LDAP URI to a directory entry containing the contact information:

```
CONTACT;ALTREP="ldap://example.com:6666/o=ABC%20Industries\,
c=US???(cn=Jim%20Dolittle)":Jim Dolittle\, ABC Industries\,
+1-919-555-1234
```

The following is an example of this property with an alternate representation of a MIME body part containing the contact information, such as a vCard [RFC2426] embedded in a text/directory media type [RFC2425]:

```
CONTACT;ALTREP="CID:part3.msg970930T083000SILVER@example.com":
Jim Dolittle\, ABC Industries\, +1-919-555-1234
```

The following is an example of this property referencing a network resource, such as a vCard [RFC2426] object containing the contact information:

```
CONTACT;ALTREP="http://example.com/pdi/jdoe.vcf":Jim  
Dolittle\, ABC Industries\, +1-919-555-1234
```

### 3.8.4.3. Organizer

Property Name: ORGANIZER

Purpose: This property defines the organizer for a calendar component.

Value Type: CAL-ADDRESS

Property Parameters: IANA, non-standard, language, common name, directory entry reference, and sent-by property parameters can be specified on this property.

Conformance: This property MUST be specified in an iCalendar object that specifies a group-scheduled calendar entity. This property MUST be specified in an iCalendar object that specifies the publication of a calendar user's busy time. This property MUST NOT be specified in an iCalendar object that specifies only a time zone definition or that defines calendar components that are not group-scheduled components, but are components only on a single user's calendar.

Description: This property is specified within the "VEVENT", "VTODO", and "VJOURNAL" calendar components to specify the organizer of a group-scheduled calendar entity. The property is specified within the "VFREEBUSY" calendar component to specify the calendar user requesting the free or busy time. When publishing a "VFREEBUSY" calendar component, the property is used to specify the calendar that the published busy time came from.

The property has the property parameters "CN", for specifying the common or display name associated with the "Organizer", "DIR", for specifying a pointer to the directory information associated with the "Organizer", "SENT-BY", for specifying another calendar user that is acting on behalf of the "Organizer". The non-standard parameters may also be specified on this property. If the "LANGUAGE" property parameter is specified, the identified language applies to the "CN" parameter value.

**Format Definition:** This property is defined by the following notation:

```

organizer = "ORGANIZER" orgparam ":"
           cal-address CRLF

orgparam  = *(
           ;
           ; The following are OPTIONAL,
           ; but MUST NOT occur more than once.
           ;
           (";" cnparam) / (";" dirparam) / (";" sentbyparam) /
           (";" languageparam) /
           ;
           ; The following is OPTIONAL,
           ; and MAY occur more than once.
           ;
           (";" other-param)
           ;
           )

```

**Example:** The following is an example of this property:

```
ORGANIZER;CN=John Smith:mailto:jsmith@example.com
```

The following is an example of this property with a pointer to the directory information associated with the organizer:

```
ORGANIZER;CN=JohnSmith;DIR="ldap://example.com:6666/o=DC%20Associates,c=US???(cn=John%20Smith)":mailto:jsmith@example.com
```

The following is an example of this property used by another calendar user who is acting on behalf of the organizer, with responses intended to be sent back to the organizer, not the other calendar user:

```
ORGANIZER;SENT-BY="mailto:jane_doe@example.com":
mailto:jsmith@example.com
```

#### 3.8.4.4. Recurrence ID

**Property Name:** RECURRENCE-ID

**Purpose:** This property is used in conjunction with the "UID" and "SEQUENCE" properties to identify a specific instance of a recurring "VEVENT", "VTODO", or "VJOURNAL" calendar component. The property value is the original value of the "DTSTART" property of the recurrence instance.



**Value Type:** The default value type is DATE-TIME. The value type can be set to a DATE value type. This property **MUST** have the same value type as the "DTSTART" property contained within the recurring component. Furthermore, this property **MUST** be specified as a date with local time if and only if the "DTSTART" property contained within the recurring component is specified as a date with local time.

**Property Parameters:** IANA, non-standard, value data type, time zone identifier, and recurrence identifier range parameters can be specified on this property.

**Conformance:** This property can be specified in an iCalendar object containing a recurring calendar component.

**Description:** The full range of calendar components specified by a recurrence set is referenced by referring to just the "UID" property value corresponding to the calendar component. The "RECURRENCE-ID" property allows the reference to an individual instance within the recurrence set.

If the value of the "DTSTART" property is a DATE type value, then the value **MUST** be the calendar date for the recurrence instance.

The DATE-TIME value is set to the time when the original recurrence instance would occur; meaning that if the intent is to change a Friday meeting to Thursday, the DATE-TIME is still set to the original Friday meeting.

The "RECURRENCE-ID" property is used in conjunction with the "UID" and "SEQUENCE" properties to identify a particular instance of a recurring event, to-do, or journal. For a given pair of "UID" and "SEQUENCE" property values, the "RECURRENCE-ID" value for a recurrence instance is fixed.

The "RANGE" parameter is used to specify the effective range of recurrence instances from the instance specified by the "RECURRENCE-ID" property value. The value for the range parameter can only be "THISANDFUTURE" to indicate a range defined by the given recurrence instance and all subsequent instances. Subsequent instances are determined by their "RECURRENCE-ID" value and not their current scheduled start time. Subsequent instances defined in separate components are not impacted by the given recurrence instance. When the given recurrence instance is rescheduled, all subsequent instances are also rescheduled by the same time difference. For instance, if the given recurrence instance is rescheduled to start 2 hours later, then all subsequent instances are also rescheduled 2 hours later.

Similarly, if the duration of the given recurrence instance is modified, then all subsequence instances are also modified to have this same duration.

Note: The "RANGE" parameter may not be appropriate to reschedule specific subsequent instances of complex recurring calendar component. Assuming an unbounded recurring calendar component scheduled to occur on Mondays and Wednesdays, the "RANGE" parameter could not be used to reschedule only the future Monday instances to occur on Tuesday instead. In such cases, the calendar application could simply truncate the unbounded recurring calendar component (i.e., with the "COUNT" or "UNTIL" rule parts), and create two new unbounded recurring calendar components for the future instances.

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

```
recurid      = "RECURRENCE-ID" ridparam ":" ridval CRLF
ridparam     = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" "VALUE" "=" ("DATE-TIME" / "DATE")) /
    (";" tzidparam) / (";" rangeparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )
ridval       = date-time / date
;Value MUST match value type
```

Example: The following are examples of this property:

```
RECURRENCE-ID;VALUE=DATE:19960401
```

```
RECURRENCE-ID;RANGE=THISANDFUTURE:19960120T120000Z
```

### 3.8.4.5. Related To

Property Name: RELATED-TO

Purpose: This property is used to represent a relationship or reference between one calendar component and another.

Value Type: TEXT

Property Parameters: IANA, non-standard, and relationship type property parameters can be specified on this property.

Conformance: This property can be specified in the "VEVENT", "VTODO", and "VJOURNAL" calendar components.

Description: The property value consists of the persistent, globally unique identifier of another calendar component. This value would be represented in a calendar component by the "UID" property.

By default, the property value points to another calendar component that has a PARENT relationship to the referencing object. The "RELTYPE" property parameter is used to either explicitly state the default PARENT relationship type to the referenced calendar component or to override the default PARENT relationship type and specify either a CHILD or SIBLING relationship. The PARENT relationship indicates that the calendar component is a subordinate of the referenced calendar component. The CHILD relationship indicates that the calendar component is a superior of the referenced calendar component. The SIBLING relationship indicates that the calendar component is a peer of the referenced calendar component.

Changes to a calendar component referenced by this property can have an implicit impact on the related calendar component. For example, if a group event changes its start or end date or time, then the related, dependent events will need to have their start and end dates changed in a corresponding way. Similarly, if a PARENT calendar component is cancelled or deleted, then there is an implied impact to the related CHILD calendar components. This property is intended only to provide information on the relationship of calendar components. It is up to the target calendar system to maintain any property implications of this relationship.

**Format Definition:** This property is defined by the following notation:

```
related      = "RELATED-TO" relparam ":" text CRLF
relparam     = *(
    ;
    ; The following is OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" reltypeparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )
```

The following is an example of this property:

```
RELATED-TO:jsmith.part7.19960817T083000.xyzMail@example.com
```

```
RELATED-TO:19960401-080045-4000F192713-0052@example.com
```

#### 3.8.4.6. Uniform Resource Locator

**Property Name:** URL

**Purpose:** This property defines a Uniform Resource Locator (URL) associated with the iCalendar object.

**Value Type:** URI

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** This property can be specified once in the "VEVENT", "VTODO", "VJOURNAL", or "VFREEBUSY" calendar components.

**Description:** This property may be used in a calendar component to convey a location where a more dynamic rendition of the calendar information associated with the calendar component can be found. This memo does not attempt to standardize the form of the URI, nor the format of the resource pointed to by the property value. If the URL property and Content-Location MIME header are both specified, they MUST point to the same resource.

**Format Definition:** This property is defined by the following notation:

url = "URL" urlparam ":" uri CRLF

urlparam = \*(";" other-param)

**Example:** The following is an example of this property:

URL:http://example.com/pub/calendars/jsmith/mytime.ics

#### 3.8.4.7. Unique Identifier

**Property Name:** UID

**Purpose:** This property defines the persistent, globally unique identifier for the calendar component.

**Value Type:** TEXT

**Property Parameters:** IANA and non-standard property parameters can be specified on this property.

**Conformance:** The property MUST be specified in the "VEVENT", "VTODO", "VJOURNAL", or "VFREEBUSY" calendar components.

**Description:** The "UID" itself MUST be a globally unique identifier. The generator of the identifier MUST guarantee that the identifier is unique. There are several algorithms that can be used to accomplish this. A good method to assure uniqueness is to put the domain name or a domain literal IP address of the host on which the identifier was created on the right-hand side of an "@", and on the left-hand side, put a combination of the current calendar date and time of day (i.e., formatted in as a DATE-TIME value) along with some other currently unique (perhaps sequential) identifier available on the system (for example, a process id number). Using a DATE-TIME value on the left-hand side and a domain name or domain literal on the right-hand side makes it possible to guarantee uniqueness since no two hosts should be using the same domain name or IP address at the same time. Though other algorithms will work, it is RECOMMENDED that the right-hand side contain some domain identifier (either of the host itself or otherwise) such that the generator of the message identifier can guarantee the uniqueness of the left-hand side within the scope of that domain.

This is the method for correlating scheduling messages with the referenced "VEVENT", "VTODO", or "VJOURNAL" calendar component.

The full range of calendar components specified by a recurrence set is referenced by referring to just the "UID" property value corresponding to the calendar component. The "RECURRENCE-ID" property allows the reference to an individual instance within the recurrence set.

This property is an important method for group-scheduling applications to match requests with later replies, modifications, or deletion requests. Calendaring and scheduling applications MUST generate this property in "VEVENT", "VTODO", and "VJOURNAL" calendar components to assure interoperability with other group-scheduling applications. This identifier is created by the calendar system that generates an iCalendar object.

Implementations MUST be able to receive and persist values of at least 255 octets for this property, but they MUST NOT truncate values in the middle of a UTF-8 multi-octet sequence.

**Format Definition:** This property is defined by the following notation:

uid = "UID" uidparam ":" text CRLF

uidparam = \*(";" other-param)

**Example:** The following is an example of this property:

UID:19960401T080045Z-4000F192713-0052@example.com

### 3.8.5. Recurrence Component Properties

The following properties specify recurrence information in calendar components.

#### 3.8.5.1. Exception Date-Times

**Property Name:** EXDATE

**Purpose:** This property defines the list of DATE-TIME exceptions for recurring events, to-dos, journal entries, or time zone definitions.

**Value Type:** The default value type for this property is DATE-TIME. The value type can be set to DATE.

**Property Parameters:** IANA, non-standard, value data type, and time zone identifier property parameters can be specified on this property.

**Conformance:** This property can be specified in recurring "VEVENT", "VTODO", and "VJOURNAL" calendar components as well as in the "STANDARD" and "DAYLIGHT" sub-components of the "VTIMEZONE" calendar component.

**Description:** The exception dates, if specified, are used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is generated by considering the initial "DTSTART" property along with the "RRULE", "RDATE", and "EXDATE" properties contained within the recurring component. The "DTSTART" property defines the first instance in the recurrence set. The "DTSTART" property value SHOULD match the pattern of the recurrence rule, if specified. The recurrence set generated with a "DTSTART" property value that doesn't match the pattern of the rule is undefined. The final recurrence set is generated by gathering all of the start DATE-TIME values generated by any of the specified "RRULE" and "RDATE" properties, and then excluding any start DATE-TIME values specified by "EXDATE" properties. This implies that start DATE-TIME values specified by "EXDATE" properties take precedence over those specified by inclusion properties (i.e., "RDATE" and "RRULE"). When duplicate instances are generated by the "RRULE" and "RDATE" properties, only one recurrence is considered. Duplicate instances are ignored.

The "EXDATE" property can be used to exclude the value specified in "DTSTART". However, in such cases, the original "DTSTART" date MUST still be maintained by the calendaring and scheduling system because the original "DTSTART" value has inherent usage dependencies by other properties such as the "RECURRENCE-ID".

**Format Definition:** This property is defined by the following notation:

```

exdate      = "EXDATE" exdtparam ":" exdtval *("," exdtval) CRLF
exdtparam   = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" "VALUE" "=" ("DATE-TIME" / "DATE")) /
    ;
    (";" tzidparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )

exdtval     = date-time / date
;Value MUST match value type

```

Example: The following is an example of this property:

```
EXDATE:19960402T010000Z,19960403T010000Z,19960404T010000Z
```

### 3.8.5.2. Recurrence Date-Times

Property Name: RDATE

Purpose: This property defines the list of DATE-TIME values for recurring events, to-dos, journal entries, or time zone definitions.

Value Type: The default value type for this property is DATE-TIME. The value type can be set to DATE or PERIOD.

Property Parameters: IANA, non-standard, value data type, and time zone identifier property parameters can be specified on this property.

Conformance: This property can be specified in recurring "VEVENT", "VTODO", and "VJOURNAL" calendar components as well as in the "STANDARD" and "DAYLIGHT" sub-components of the "VTIMEZONE" calendar component.

Description: This property can appear along with the "RRULE" property to define an aggregate set of repeating occurrences. When they both appear in a recurring component, the recurrence



instances are defined by the union of occurrences defined by both the "RDATE" and "RRULE".

The recurrence dates, if specified, are used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is generated by considering the initial "DTSTART" property along with the "RRULE", "RDATE", and "EXDATE" properties contained within the recurring component. The "DTSTART" property defines the first instance in the recurrence set. The "DTSTART" property value SHOULD match the pattern of the recurrence rule, if specified. The recurrence set generated with a "DTSTART" property value that doesn't match the pattern of the rule is undefined. The final recurrence set is generated by gathering all of the start DATE-TIME values generated by any of the specified "RRULE" and "RDATE" properties, and then excluding any start DATE-TIME values specified by "EXDATE" properties. This implies that start DATE-TIME values specified by "EXDATE" properties take precedence over those specified by inclusion properties (i.e., "RDATE" and "RRULE"). Where duplicate instances are generated by the "RRULE" and "RDATE" properties, only one recurrence is considered. Duplicate instances are ignored.

**Format Definition:** This property is defined by the following notation:

```

rdate      = "RDATE" rdtparam ":" rdtval *("," rdtval) CRLF

rdtparam   = *(
    ;
    ; The following are OPTIONAL,
    ; but MUST NOT occur more than once.
    ;
    (";" "VALUE" "=" ("DATE-TIME" / "DATE" / "PERIOD")) /
    (";" tzidparam) /
    ;
    ; The following is OPTIONAL,
    ; and MAY occur more than once.
    ;
    (";" other-param)
    ;
    )

rdtval     = date-time / date / period
;Value MUST match value type

```

Example: The following are examples of this property:

```
RDATE:19970714T123000Z
```

```
RDATE;TZID=America/New_York:19970714T083000
```

```
RDATE;VALUE=PERIOD:19960403T020000Z/19960403T040000Z,  
19960404T010000Z/PT3H
```

```
RDATE;VALUE=DATE:19970101,19970120,19970217,19970421  
19970526,19970704,19970901,19971014,19971128,19971129,19971225
```

### 3.8.5.3. Recurrence Rule

Property Name: RRULE

Purpose: This property defines a rule or repeating pattern for recurring events, to-dos, journal entries, or time zone definitions.

Value Type: RECUR

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified in recurring "VEVENT", "VTODO", and "VJOURNAL" calendar components as well as in the "STANDARD" and "DAYLIGHT" sub-components of the "VTIMEZONE" calendar component, but it SHOULD NOT be specified more than once. The recurrence set generated with multiple "RRULE" properties is undefined.

Description: The recurrence rule, if specified, is used in computing the recurrence set. The recurrence set is the complete set of recurrence instances for a calendar component. The recurrence set is generated by considering the initial "DTSTART" property along with the "RRULE", "RDATE", and "EXDATE" properties contained within the recurring component. The "DTSTART" property defines the first instance in the recurrence set. The "DTSTART" property value SHOULD be synchronized with the recurrence rule, if specified. The recurrence set generated with a "DTSTART" property value not synchronized with the recurrence rule is undefined. The final recurrence set is generated by gathering all of the start DATE-TIME values generated by any of the specified "RRULE" and "RDATE" properties, and then excluding any start DATE-TIME values specified by "EXDATE" properties. This implies that start DATE-TIME values specified by "EXDATE" properties take precedence over those specified by inclusion properties (i.e., "RDATE" and "RRULE"). Where duplicate instances are generated by the "RRULE"

and "RDATE" properties, only one recurrence is considered. Duplicate instances are ignored.

The "DTSTART" property specified within the iCalendar object defines the first instance of the recurrence. In most cases, a "DTSTART" property of DATE-TIME value type used with a recurrence rule, should be specified as a date with local time and time zone reference to make sure all the recurrence instances start at the same local time regardless of time zone changes.

If the duration of the recurring component is specified with the "DTEND" or "DUE" property, then the same exact duration will apply to all the members of the generated recurrence set. Else, if the duration of the recurring component is specified with the "DURATION" property, then the same nominal duration will apply to all the members of the generated recurrence set and the exact duration of each recurrence instance will depend on its specific start time. For example, recurrence instances of a nominal duration of one day will have an exact duration of more or less than 24 hours on a day where a time zone shift occurs. The duration of a specific recurrence may be modified in an exception component or simply by using an "RDATE" property of PERIOD value type.

**Format Definition:** This property is defined by the following notation:

```
rrule      = "RRULE" rrulparam ":" recur CRLF
rrulparam  = *(";" other-param)
```

**Example:** All examples assume the Eastern United States time zone.

Daily for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=DAILY;COUNT=10
```

==> (1997 9:00 AM EDT) September 2-11

Daily until December 24, 1997:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=DAILY;UNTIL=19971224T000000Z
```

==> (1997 9:00 AM EDT) September 2-30;October 1-25  
 (1997 9:00 AM EST) October 26-31;November 1-30;December 1-23

Every other day - forever:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=DAILY;INTERVAL=2
```

```
=> (1997 9:00 AM EDT) September 2,4,6,8...24,26,28,30;
      October 2,4,6...20,22,24
      (1997 9:00 AM EST) October 26,28,30;
      November 1,3,5,7...25,27,29;
      December 1,3,...
```

Every 10 days, 5 occurrences:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=DAILY;INTERVAL=10;COUNT=5
```

```
=> (1997 9:00 AM EDT) September 2,12,22;
      October 2,12
```

Every day in January, for 3 years:

```
DTSTART;TZID=America/New_York:19980101T090000

RRULE:FREQ=YEARLY;UNTIL=20000131T140000Z;
  BYMONTH=1;BYDAY=SU,MO,TU,WE,TH,FR,SA
or
RRULE:FREQ=DAILY;UNTIL=20000131T140000Z;BYMONTH=1
```

```
=> (1998 9:00 AM EST)January 1-31
      (1999 9:00 AM EST)January 1-31
      (2000 9:00 AM EST)January 1-31
```

Weekly for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=WEEKLY;COUNT=10
```

```
=> (1997 9:00 AM EDT) September 2,9,16,23,30;October 7,14,21
      (1997 9:00 AM EST) October 28;November 4
```

Weekly until December 24, 1997:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=WEEKLY;UNTIL=19971224T000000Z
```

```
=> (1997 9:00 AM EDT) September 2,9,16,23,30;
    October 7,14,21
    (1997 9:00 AM EST) October 28;
    November 4,11,18,25;
    December 2,9,16,23
```

Every other week - forever:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=WEEKLY;INTERVAL=2;WKST=SU
```

```
=> (1997 9:00 AM EDT) September 2,16,30;
    October 14
    (1997 9:00 AM EST) October 28;
    November 11,25;
    December 9,23
    (1998 9:00 AM EST) January 6,20;
    February 3, 17
    ...
```

Weekly on Tuesday and Thursday for five weeks:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=WEEKLY;UNTIL=19971007T000000Z;WKST=SU;BYDAY=TU,TH
```

or

```
RRULE:FREQ=WEEKLY;COUNT=10;WKST=SU;BYDAY=TU,TH
```

```
=> (1997 9:00 AM EDT) September 2,4,9,11,16,18,23,25,30;
    October 2
```

Every other week on Monday, Wednesday, and Friday until December 24, 1997, starting on Monday, September 1, 1997:

```
DTSTART;TZID=America/New_York:19970901T090000
RRULE:FREQ=WEEKLY;INTERVAL=2;UNTIL=19971224T000000Z;WKST=SU;
  BYDAY=MO,WE,FR
```

```
=> (1997 9:00 AM EDT) September 1,3,5,15,17,19,29;
    October 1,3,13,15,17
    (1997 9:00 AM EST) October 27,29,31;
    November 10,12,14,24,26,28;
```

December 8,10,12,22

Every other week on Tuesday and Thursday, for 8 occurrences:

DTSTART;TZID=America/New\_York:19970902T090000  
RRULE:FREQ=WEEKLY;INTERVAL=2;COUNT=8;WKST=SU;BYDAY=TU,TH

==> (1997 9:00 AM EDT) September 2,4,16,18,30;  
October 2,14,16

Monthly on the first Friday for 10 occurrences:

DTSTART;TZID=America/New\_York:19970905T090000  
RRULE:FREQ=MONTHLY;COUNT=10;BYDAY=1FR

==> (1997 9:00 AM EDT) September 5;October 3  
(1997 9:00 AM EST) November 7;December 5  
(1998 9:00 AM EST) January 2;February 6;March 6;April 3  
(1998 9:00 AM EDT) May 1;June 5

Monthly on the first Friday until December 24, 1997:

DTSTART;TZID=America/New\_York:19970905T090000  
RRULE:FREQ=MONTHLY;UNTIL=19971224T000000Z;BYDAY=1FR

==> (1997 9:00 AM EDT) September 5; October 3  
(1997 9:00 AM EST) November 7; December 5

Every other month on the first and last Sunday of the month for 10 occurrences:

DTSTART;TZID=America/New\_York:19970907T090000  
RRULE:FREQ=MONTHLY;INTERVAL=2;COUNT=10;BYDAY=1SU,-1SU

==> (1997 9:00 AM EDT) September 7,28  
(1997 9:00 AM EST) November 2,30  
(1998 9:00 AM EST) January 4,25;March 1,29  
(1998 9:00 AM EDT) May 3,31

Monthly on the second-to-last Monday of the month for 6 months:

DTSTART;TZID=America/New\_York:19970922T090000  
RRULE:FREQ=MONTHLY;COUNT=6;BYDAY=-2MO

==> (1997 9:00 AM EDT) September 22;October 20  
(1997 9:00 AM EST) November 17;December 22  
(1998 9:00 AM EST) January 19;February 16

Monthly on the third-to-the-last day of the month, forever:

```
DTSTART;TZID=America/New_York:19970928T090000
RRULE:FREQ=MONTHLY;BYMONTHDAY=-3
```

```
=> (1997 9:00 AM EDT) September 28
    (1997 9:00 AM EST) October 29;November 28;December 29
    (1998 9:00 AM EST) January 29;February 26
    ...
```

Monthly on the 2nd and 15th of the month for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=MONTHLY;COUNT=10;BYMONTHDAY=2,15
```

```
=> (1997 9:00 AM EDT) September 2,15;October 2,15
    (1997 9:00 AM EST) November 2,15;December 2,15
    (1998 9:00 AM EST) January 2,15
```

Monthly on the first and last day of the month for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970930T090000
RRULE:FREQ=MONTHLY;COUNT=10;BYMONTHDAY=1,-1
```

```
=> (1997 9:00 AM EDT) September 30;October 1
    (1997 9:00 AM EST) October 31;November 1,30;December 1,31
    (1998 9:00 AM EST) January 1,31;February 1
```

Every 18 months on the 10th thru 15th of the month for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970910T090000
RRULE:FREQ=MONTHLY;INTERVAL=18;COUNT=10;BYMONTHDAY=10,11,12,
    13,14,15
```

```
=> (1997 9:00 AM EDT) September 10,11,12,13,14,15
    (1999 9:00 AM EST) March 10,11,12,13
```

Every Tuesday, every other month:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=MONTHLY;INTERVAL=2;BYDAY=TU
```

```
=> (1997 9:00 AM EDT) September 2,9,16,23,30
    (1997 9:00 AM EST) November 4,11,18,25
    (1998 9:00 AM EST) January 6,13,20,27;March 3,10,17,24,31
    ...
```

Yearly in June and July for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970610T090000
RRULE:FREQ=YEARLY;COUNT=10;BYMONTH=6,7
```

```
=> (1997 9:00 AM EDT) June 10;July 10
    (1998 9:00 AM EDT) June 10;July 10
    (1999 9:00 AM EDT) June 10;July 10
    (2000 9:00 AM EDT) June 10;July 10
    (2001 9:00 AM EDT) June 10;July 10
```

Note: Since none of the BYDAY, BYMONTHDAY, or BYYEARDAY components are specified, the day is gotten from "DTSTART".

Every other year on January, February, and March for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970310T090000
RRULE:FREQ=YEARLY;INTERVAL=2;COUNT=10;BYMONTH=1,2,3
```

```
=> (1997 9:00 AM EST) March 10
    (1999 9:00 AM EST) January 10;February 10;March 10
    (2001 9:00 AM EST) January 10;February 10;March 10
    (2003 9:00 AM EST) January 10;February 10;March 10
```

Every third year on the 1st, 100th, and 200th day for 10 occurrences:

```
DTSTART;TZID=America/New_York:19970101T090000
RRULE:FREQ=YEARLY;INTERVAL=3;COUNT=10;BYYEARDAY=1,100,200
```

```
=> (1997 9:00 AM EST) January 1
    (1997 9:00 AM EDT) April 10;July 19
    (2000 9:00 AM EST) January 1
    (2000 9:00 AM EDT) April 9;July 18
    (2003 9:00 AM EST) January 1
    (2003 9:00 AM EDT) April 10;July 19
    (2006 9:00 AM EST) January 1
```

Every 20th Monday of the year, forever:

```
DTSTART;TZID=America/New_York:19970519T090000
RRULE:FREQ=YEARLY;BYDAY=20MO
```

```
=> (1997 9:00 AM EDT) May 19
    (1998 9:00 AM EDT) May 18
    (1999 9:00 AM EDT) May 17
    ...
```



Monday of week number 20 (where the default start of the week is Monday), forever:

```
DTSTART;TZID=America/New_York:19970512T090000
RRULE:FREQ=YEARLY;BYWEEKNO=20;BYDAY=MO
```

```
==> (1997 9:00 AM EDT) May 12
      (1998 9:00 AM EDT) May 11
      (1999 9:00 AM EDT) May 17
      ...
```

Every Thursday in March, forever:

```
DTSTART;TZID=America/New_York:19970313T090000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=TH
```

```
==> (1997 9:00 AM EST) March 13,20,27
      (1998 9:00 AM EST) March 5,12,19,26
      (1999 9:00 AM EST) March 4,11,18,25
      ...
```

Every Thursday, but only during June, July, and August, forever:

```
DTSTART;TZID=America/New_York:19970605T090000
RRULE:FREQ=YEARLY;BYDAY=TH;BYMONTH=6,7,8
```

```
==> (1997 9:00 AM EDT) June 5,12,19,26;July 3,10,17,24,31;
      August 7,14,21,28
      (1998 9:00 AM EDT) June 4,11,18,25;July 2,9,16,23,30;
      August 6,13,20,27
      (1999 9:00 AM EDT) June 3,10,17,24;July 1,8,15,22,29;
      August 5,12,19,26
      ...
```

Every Friday the 13th, forever:

```
DTSTART;TZID=America/New_York:19970902T090000
EXDATE;TZID=America/New_York:19970902T090000
RRULE:FREQ=MONTHLY;BYDAY=FR;BYMONTHDAY=13
```

```
==> (1998 9:00 AM EST) February 13;March 13;November 13
      (1999 9:00 AM EDT) August 13
      (2000 9:00 AM EDT) October 13
      ...
```

The first Saturday that follows the first Sunday of the month,  
forever:

```
DTSTART;TZID=America/New_York:19970913T090000
RRULE:FREQ=MONTHLY;BYDAY=SA;BYMONTHDAY=7,8,9,10,11,12,13
```

```
=> (1997 9:00 AM EDT) September 13;October 11
    (1997 9:00 AM EST) November 8;December 13
    (1998 9:00 AM EST) January 10;February 7;March 7
    (1998 9:00 AM EDT) April 11;May 9;June 13...
    ...
```

Every 4 years, the first Tuesday after a Monday in November,  
forever (U.S. Presidential Election day):

```
DTSTART;TZID=America/New_York:19961105T090000
RRULE:FREQ=YEARLY;INTERVAL=4;BYMONTH=11;BYDAY=TU;
      BYMONTHDAY=2,3,4,5,6,7,8
```

```
=> (1996 9:00 AM EST) November 5
    (2000 9:00 AM EST) November 7
    (2004 9:00 AM EST) November 2
    ...
```

The third instance into the month of one of Tuesday, Wednesday, or  
Thursday, for the next 3 months:

```
DTSTART;TZID=America/New_York:19970904T090000
RRULE:FREQ=MONTHLY;COUNT=3;BYDAY=TU,WE,TH;BYSETPOS=3
```

```
=> (1997 9:00 AM EDT) September 4;October 7
    (1997 9:00 AM EST) November 6
```

The second-to-last weekday of the month:

```
DTSTART;TZID=America/New_York:19970929T090000
RRULE:FREQ=MONTHLY;BYDAY=MO,TU,WE,TH,FR;BYSETPOS=-2
```

```
=> (1997 9:00 AM EDT) September 29
    (1997 9:00 AM EST) October 30;November 27;December 30
    (1998 9:00 AM EST) January 29;February 26;March 30
    ...
```

Every 3 hours from 9:00 AM to 5:00 PM on a specific day:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=HOURLY;INTERVAL=3;UNTIL=19970902T170000Z
```

=> (September 2, 1997 EDT) 09:00,12:00,15:00

Every 15 minutes for 6 occurrences:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=MINUTELY;INTERVAL=15;COUNT=6
```

=> (September 2, 1997 EDT) 09:00,09:15,09:30,09:45,10:00,10:15

Every hour and a half for 4 occurrences:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=MINUTELY;INTERVAL=90;COUNT=4
```

=> (September 2, 1997 EDT) 09:00,10:30,12:00,13:30

Every 20 minutes from 9:00 AM to 4:40 PM every day:

```
DTSTART;TZID=America/New_York:19970902T090000
RRULE:FREQ=DAILY;BYHOUR=9,10,11,12,13,14,15,16;BYMINUTE=0,20,40
or
RRULE:FREQ=MINUTELY;INTERVAL=20;BYHOUR=9,10,11,12,13,14,15,16
```

```
=> (September 2, 1997 EDT) 9:00,9:20,9:40,10:00,10:20,
    ... 16:00,16:20,16:40
    (September 3, 1997 EDT) 9:00,9:20,9:40,10:00,10:20,
    ... 16:00,16:20,16:40
    ...
```

An example where the days generated makes a difference because of WKST:

```
DTSTART;TZID=America/New_York:19970805T090000
RRULE:FREQ=WEEKLY;INTERVAL=2;COUNT=4;BYDAY=TU,SU;WKST=MO
```

=> (1997 EDT) August 5,10,19,24

changing only WKST from MO to SU, yields different results...

```
DTSTART;TZID=America/New_York:19970805T090000
RRULE:FREQ=WEEKLY;INTERVAL=2;COUNT=4;BYDAY=TU,SU;WKST=SU
```

=> (1997 EDT) August 5,17,19,31

An example where an invalid date (i.e., February 30) is ignored.

```
DTSTART;TZID=America/New_York:20070115T090000
RRULE:FREQ=MONTHLY;BYMONTHDAY=15,30;COUNT=5
```

```
==> (2007 EST) January 15,30
      (2007 EST) February 15
      (2007 EDT) March 15,30
```

### 3.8.6. Alarm Component Properties

The following properties specify alarm information in calendar components.

#### 3.8.6.1. Action

Property Name: ACTION

Purpose: This property defines the action to be invoked when an alarm is triggered.

Value Type: TEXT

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property MUST be specified once in a "VALARM" calendar component.

Description: Each "VALARM" calendar component has a particular type of action with which it is associated. This property specifies the type of action. Applications MUST ignore alarms with x-name and iana-token values they don't recognize.

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

```
action      = "ACTION" actionparam ":" actionvalue CRLF
```

```
actionparam = *(";" other-param)
```

```
actionvalue = "AUDIO" / "DISPLAY" / "EMAIL"
              / iana-token / x-name
```

Example: The following are examples of this property in a "VALARM" calendar component:

ACTION:AUDIO

ACTION:DISPLAY

### 3.8.6.2. Repeat Count

Property Name: REPEAT

Purpose: This property defines the number of times the alarm should be repeated, after the initial trigger.

Value Type: INTEGER

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified in a "VALARM" calendar component.

Description: This property defines the number of times an alarm should be repeated after its initial trigger. If the alarm triggers more than once, then this property MUST be specified along with the "DURATION" property.

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

```
repeat = "REPEAT" repparam ":" integer CRLF
;Default is "0", zero.
```

```
repparam = *(";" other-param)
```

Example: The following is an example of this property for an alarm that repeats 4 additional times with a 5-minute delay after the initial triggering of the alarm:

```
REPEAT:4
DURATION:PT5M
```

### 3.8.6.3. Trigger

Property Name: TRIGGER

Purpose: This property specifies when an alarm will trigger.

Value Type: The default value type is DURATION. The value type can be set to a DATE-TIME value type, in which case the value MUST specify a UTC-formatted DATE-TIME value.

**Property Parameters:** IANA, non-standard, value data type, time zone identifier, or trigger relationship property parameters can be specified on this property. The trigger relationship property parameter **MUST** only be specified when the value type is "DURATION".

**Conformance:** This property **MUST** be specified in the "VALARM" calendar component.

**Description:** This property defines when an alarm will trigger. The default value type is DURATION, specifying a relative time for the trigger of the alarm. The default duration is relative to the start of an event or to-do with which the alarm is associated. The duration can be explicitly set to trigger from either the end or the start of the associated event or to-do with the "RELATED" parameter. A value of START will set the alarm to trigger off the start of the associated event or to-do. A value of END will set the alarm to trigger off the end of the associated event or to-do.

Either a positive or negative duration may be specified for the "TRIGGER" property. An alarm with a positive duration is triggered after the associated start or end of the event or to-do. An alarm with a negative duration is triggered before the associated start or end of the event or to-do.

The "RELATED" property parameter is not valid if the value type of the property is set to DATE-TIME (i.e., for an absolute date and time alarm trigger). If a value type of DATE-TIME is specified, then the property value **MUST** be specified in the UTC time format. If an absolute trigger is specified on an alarm for a recurring event or to-do, then the alarm will only trigger for the specified absolute DATE-TIME, along with any specified repeating instances.

If the trigger is set relative to START, then the "DTSTART" property **MUST** be present in the associated "VEVENT" or "VTODO" calendar component. If an alarm is specified for an event with the trigger set relative to the END, then the "DTEND" property or the "DTSTART" and "DURATION" properties **MUST** be present in the associated "VEVENT" calendar component. If the alarm is specified for a to-do with a trigger set relative to the END, then either the "DUE" property or the "DTSTART" and "DURATION" properties **MUST** be present in the associated "VTODO" calendar component.

Alarms specified in an event or to-do that is defined in terms of a DATE value type will be triggered relative to 00:00:00 of the user's configured time zone on the specified date, or relative to 00:00:00 UTC on the specified date if no configured time zone can be found for the user. For example, if "DTSTART" is a DATE value

set to 19980205 then the duration trigger will be relative to 19980205T000000 America/New\_York for a user configured with the America/New\_York time zone.

**Format Definition:** This property is defined by the following notation:

```
trigger      = "TRIGGER" (trigrel / trigabs) CRLF
trigrel      = *(
                ;
                ; The following are OPTIONAL,
                ; but MUST NOT occur more than once.
                ;
                (";" "VALUE" "=" "DURATION") /
                (";" trigrelparam) /
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
                ;
                (";" other-param)
                ;
                ) ":" dur-value
trigabs      = *(
                ;
                ; The following is REQUIRED,
                ; but MUST NOT occur more than once.
                ;
                (";" "VALUE" "=" "DATE-TIME") /
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
                ;
                (";" other-param)
                ;
                ) ":" date-time
```

**Example:** A trigger set 15 minutes prior to the start of the event or to-do.

```
TRIGGER:-PT15M
```

A trigger set five minutes after the end of an event or the due date of a to-do.

```
TRIGGER;RELATED=END:PT5M
```

A trigger set to an absolute DATE-TIME.

TRIGGER;VALUE=DATE-TIME:19980101T050000Z

### 3.8.7. Change Management Component Properties

The following properties specify change management information in calendar components.

#### 3.8.7.1. Date-Time Created

Property Name: CREATED

Purpose: This property specifies the date and time that the calendar information was created by the calendar user agent in the calendar store.

Note: This is analogous to the creation date and time for a file in the file system.

Value Type: DATE-TIME

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: The property can be specified once in "VEVENT", "VTODO", or "VJOURNAL" calendar components. The value MUST be specified as a date with UTC time.

Description: This property specifies the date and time that the calendar information was created by the calendar user agent in the calendar store.

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

created = "CREATED" creaparam ":" date-time CRLF

creaparam = \*(";" other-param)

Example: The following is an example of this property:

CREATED:19960329T133000Z



### 3.8.7.2. Date-Time Stamp

Property Name: DTSTAMP

Purpose: In the case of an iCalendar object that specifies a "METHOD" property, this property specifies the date and time that the instance of the iCalendar object was created. In the case of an iCalendar object that doesn't specify a "METHOD" property, this property specifies the date and time that the information associated with the calendar component was last revised in the calendar store.

Value Type: DATE-TIME

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property MUST be included in the "VEVENT", "VTODO", "VJOURNAL", or "VFREEBUSY" calendar components.

Description: The value MUST be specified in the UTC time format.

This property is also useful to protocols such as [2447bis] that have inherent latency issues with the delivery of content. This property will assist in the proper sequencing of messages containing iCalendar objects.

In the case of an iCalendar object that specifies a "METHOD" property, this property differs from the "CREATED" and "LAST-MODIFIED" properties. These two properties are used to specify when the particular calendar data in the calendar store was created and last modified. This is different than when the iCalendar object representation of the calendar service information was created or last modified.

In the case of an iCalendar object that doesn't specify a "METHOD" property, this property is equivalent to the "LAST-MODIFIED" property.

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

dtstamp = "DTSTAMP" stmparam ":" date-time CRLF

stmparam = \*(";" other-param)

Example:

DTSTAMP:19971210T080000Z

### 3.8.7.3. Last Modified

Property Name: LAST-MODIFIED

Purpose: This property specifies the date and time that the information associated with the calendar component was last revised in the calendar store.

Note: This is analogous to the modification date and time for a file in the file system.

Value Type: DATE-TIME

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified in the "VEVENT", "VTODO", "VJOURNAL", or "VTIMEZONE" calendar components.

Description: The property value MUST be specified in the UTC time format.

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

last-mod = "LAST-MODIFIED" lstparam ":" date-time CRLF

lstparam = \*(";" other-param)

Example: The following is an example of this property:

LAST-MODIFIED:19960817T133000Z

### 3.8.7.4. Sequence Number

Property Name: SEQUENCE

Purpose: This property defines the revision sequence number of the calendar component within a sequence of revisions.

Value Type: INTEGER

Property Parameters: IANA and non-standard property parameters can be specified on this property.

**Conformance:** The property can be specified in "VEVENT", "VTODO", or "VJOURNAL" calendar component.

**Description:** When a calendar component is created, its sequence number is 0. It is monotonically incremented by the "Organizer's" CUA each time the "Organizer" makes a significant revision to the calendar component.

The "Organizer" includes this property in an iCalendar object that it sends to an "Attendee" to specify the current version of the calendar component.

The "Attendee" includes this property in an iCalendar object that it sends to the "Organizer" to specify the version of the calendar component to which the "Attendee" is referring.

A change to the sequence number is not the mechanism that an "Organizer" uses to request a response from the "Attendees". The "RSVP" parameter on the "ATTENDEE" property is used by the "Organizer" to indicate that a response from the "Attendees" is requested.

Recurrence instances of a recurring component MAY have different sequence numbers.

**Format Definition:** This property is defined by the following notation:

```
seq = "SEQUENCE" seqparam ":" integer CRLF
; Default is "0"
```

```
seqparam = *(";" other-param)
```

**Example:** The following is an example of this property for a calendar component that was just created by the "Organizer":

```
SEQUENCE:0
```

The following is an example of this property for a calendar component that has been revised two different times by the "Organizer":

```
SEQUENCE:2
```

### 3.8.8. Miscellaneous Component Properties

The following properties specify information about a number of miscellaneous features of calendar components.

### 3.8.8.1. IANA Properties

**Property Name:** An IANA-registered property name

**Value Type:** The default value type is TEXT. The value type can be set to any value type.

**Property Parameters:** Any parameter can be specified on this property.

**Description:** This specification allows other properties registered with IANA to be specified in any calendar components. Compliant applications are expected to be able to parse these other IANA-registered properties but can ignore them.

**Format Definition:** This property is defined by the following notation:

iana-prop = iana-token \*(";" icalparameter) ":" value CRLF

**Example:** The following are examples of properties that might be registered to IANA:

DRESSCODE:CASUAL

NON-SMOKING;VALUE=BOOLEAN:TRUE

### 3.8.8.2. Non-Standard Properties

**Property Name:** Any property name with a "X-" prefix

**Purpose:** This class of property provides a framework for defining non-standard properties.

**Value Type:** The default value type is TEXT. The value type can be set to any value type.

**Property Parameters:** IANA, non-standard, and language property parameters can be specified on this property.

**Conformance:** This property can be specified in any calendar component.

**Description:** The MIME Calendaring and Scheduling Content Type provides a "standard mechanism for doing non-standard things". This extension support is provided for implementers to "push the envelope" on the existing version of the memo. Extension properties are specified by property and/or property parameter

names that have the prefix text of "X-" (the two-character sequence: LATIN CAPITAL LETTER X character followed by the HYPHEN-MINUS character). It is recommended that vendors concatenate onto this sentinel another short prefix text to identify the vendor. This will facilitate readability of the extensions and minimize possible collision of names between different vendors. User agents that support this content type are expected to be able to parse the extension properties and property parameters but can ignore them.

At present, there is no registration authority for names of extension properties and property parameters. The value type for this property is TEXT. Optionally, the value type can be any of the other valid value types.

**Format Definition:** This property is defined by the following notation:

x-prop = x-name \*(";" icalparameter) ":" value CRLF

**Example:** The following might be the ABC vendor's extension for an audio-clip form of subject property:

X-ABC-MMSUBJ;VALUE=URI;FMTPROPERTY=audio/basic:http://www.example.org/mysubj.au

### 3.8.8.3. Request Status

**Property Name:** REQUEST-STATUS

**Purpose:** This property defines the status code returned for a scheduling request.

**Value Type:** TEXT

**Property Parameters:** IANA, non-standard, and language property parameters can be specified on this property.

**Conformance:** The property can be specified in the "VEVENT", "VTODO", "VJOURNAL", or "VFREEBUSY" calendar component.

**Description:** This property is used to return status code information related to the processing of an associated iCalendar object. The value type for this property is TEXT.

The value consists of a short return status component, a longer return status description component, and optionally a status-specific data component. The components of the value are separated by the SEMICOLON character.

The short return status is a PERIOD character separated pair or 3-tuple of integers. For example, "3.1" or "3.1.1". The successive levels of integers provide for a successive level of status code granularity.

The following are initial classes for the return status code. Individual iCalendar object methods will define specific return status codes for these classes. In addition, other classes for the return status code may be defined using the registration process defined later in this memo.

Short Return Status Code	Longer Return Status Description
1.xx	Preliminary success. This class of status code indicates that the request has been initially processed but that completion is pending.
2.xx	Successful. This class of status code indicates that the request was completed successfully. However, the exact status code can indicate that a fallback has been taken.
3.xx	Client Error. This class of status code indicates that the request was not successful. The error is the result of either a syntax or a semantic error in the client-formatted request. Request should not be retried until the condition in the request is corrected.
4.xx	Scheduling Error. This class of status code indicates that the request was not successful. Some sort of error occurred within the calendaring and scheduling service, not directly related to the request itself.

**Format Definition:** This property is defined by the following notation:

```

rstatus      = "REQUEST-STATUS" rstatparam ":"
               statcode ";" statdesc [";" extdata]

rstatparam = *(
               ;
               ; The following is OPTIONAL,
               ; but MUST NOT occur more than once.
               ;
               (";" languageparam) /
               ;
               ; The following is OPTIONAL,
               ; and MAY occur more than once.
               ;
               (";" other-param)
               ;
               )

statcode     = 1*DIGIT 1*2("." 1*DIGIT)
               ;Hierarchical, numeric return status code

statdesc     = text
               ;Textual status description

extdata      = text
               ;Textual exception data.  For example, the offending property
               ;name and value or complete property line.

```

**Example:** The following are some possible examples of this property.

The COMMA and SEMICOLON separator characters in the property value are BACKSLASH character escaped because they appear in a text value.

```
REQUEST-STATUS:2.0;Success
```

```
REQUEST-STATUS:3.1;Invalid property value;DTSTART:96-Apr-01
```

```
REQUEST-STATUS:2.8; Success\, repeating event ignored. Scheduled
as a single event.;RRULE:FREQ=WEEKLY\;INTERVAL=2
```

```
REQUEST-STATUS:4.1;Event conflict. Date-time is busy.
```

```
REQUEST-STATUS:3.7;Invalid calendar user;ATTENDEE:
mailto:jsmith@example.com
```

#### 4. iCalendar Object Examples

The following examples are provided as an informational source of illustrative iCalendar objects consistent with this content type.

The following example specifies a three-day conference that begins at 2:30 P.M. UTC, September 18, 1996 and ends at 10:00 P.M. UTC, September 20, 1996.

```
BEGIN:VCALENDAR
PRODID:-//xyz Corp//NONSGML PDA Calendar Version 1.0//EN
VERSION:2.0
BEGIN:VEVENT
DTSTAMP:19960704T120000Z
UID:uid1@example.com
ORGANIZER:mailto:jsmith@example.com
DTSTART:19960918T143000Z
DTEND:19960920T220000Z
STATUS:CONFIRMED
CATEGORIES:CONFERENCE
SUMMARY:Networld+Interop Conference
DESCRIPTION:Networld+Interop Conference
    and Exhibit\nAtlanta World Congress Center\n
    Atlanta\, Georgia
END:VEVENT
END:VCALENDAR
```

The following example specifies a group-scheduled meeting that begins at 8:30 AM EST on March 12, 1998 and ends at 9:30 AM EST on March 12, 1998. The "Organizer" has scheduled the meeting with one or more calendar users in a group. A time zone specification for Eastern United States has been specified.

```
BEGIN:VCALENDAR
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VTIMEZONE
TZID:America/New_York
BEGIN:STANDARD
DTSTART:19981025T020000
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:EST
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19990404T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
```



```
TZNAME:EDT
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTAMP:19980309T231000Z
UID:guid-1.example.com
ORGANIZER:mailto:mrbig@example.com
ATTENDEE;RSVP=TRUE;ROLE=REQ-PARTICIPANT;CUTYPE=GROUP:
  mailto:employee-A@example.com
DESCRIPTION:Project XYZ Review Meeting
CATEGORIES:MEETING
CLASS:PUBLIC
CREATED:19980309T130000Z
SUMMARY:XYZ Project Review
DTSTART;TZID=America/New_York:19980312T083000
DTEND;TZID=America/New_York:19980312T093000
LOCATION:1CP Conference Room 4350
END:VEVENT
END:VCALENDAR
```

The following is an example of an iCalendar object passed in a MIME message with a single body part consisting of a "text/calendar" Content Type.

```
TO:jsmith@example.com
FROM:jdoe@example.com
MIME-VERSION:1.0
MESSAGE-ID:<id3@example.com>
CONTENT-TYPE:text/calendar; method="xyz"; component="VEVENT"

BEGIN:VCALENDAR
METHOD:xyz
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT
DTSTAMP:19970324T120000Z
SEQUENCE:0
UID:uid3@example.com
ORGANIZER:mailto:jdoe@example.com
ATTENDEE;RSVP=TRUE:mailto:jsmith@example.com
DTSTART:19970324T123000Z
DTEND:19970324T210000Z
CATEGORIES:MEETING,PROJECT
CLASS:PUBLIC
SUMMARY:Calendaring Interoperability Planning Meeting
DESCRIPTION:Discuss how we can test c&s interoperability\n
  using iCalendar and other IETF standards.
LOCATION:LDB Lobby
```

```
ATTACH;FMTTYPE=application/postscript:ftp://example.com/pub/
  conf/bkgrnd.ps
END:VEVENT
END:VCALENDAR
```

The following is an example of a to-do due on April 15, 1998. An audio alarm has been specified to remind the calendar user at noon, the day before the to-do is expected to be completed and repeat hourly, four additional times. The to-do definition has been modified twice since it was initially created.

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VTODO
DTSTAMP:19980130T134500Z
SEQUENCE:2
UID:uid4@example.com
ORGANIZER:mailto:unclesam@example.com
ATTENDEE;PARTSTAT=ACCEPTED:mailto:jqppublic@example.com
DUE:19980415T000000
STATUS:NEEDS-ACTION
SUMMARY:Submit Income Taxes
BEGIN:VALARM
ACTION:AUDIO
TRIGGER:19980403T120000Z
ATTACH;FMTTYPE=audio/basic:http://example.com/pub/audio-
  files/ssbanner.aud
REPEAT:4
DURATION:PT1H
END:VALARM
END:VTODO
END:VCALENDAR
```

The following is an example of a journal entry:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VJOURNAL
DTSTAMP:19970324T120000Z
UID:uid5@example.com
ORGANIZER:mailto:jsmith@example.com
STATUS:DRAFT
CLASS:PUBLIC
CATEGORIES:Project Report,XYZ,Weekly Meeting
DESCRIPTION:Project xyz Review Meeting Minutes\n
  Agenda\n1. Review of project version 1.0 requirements.\n2.
```

```

Definition
of project processes.\n3. Review of project schedule.\n
Participants: John Smith\, Jane Doe\, Jim Dandy\n-It was
decided that the requirements need to be signed off by
product marketing.\n-Project processes were accepted.\n
-Project schedule needs to account for scheduled holidays
and employee vacation time. Check with HR for specific
dates.\n-New schedule will be distributed by Friday.\n-
Next weeks meeting is cancelled. No meeting until 3/23.
END:VJOURNAL
END:VCALENDAR

```

The following is an example of published busy time information. The iCalendar object might be placed in the network resource <http://www.example.com/calendar/busytime/jsmith.ifb>.

```

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//RDU Software//NONSGML HandCal//EN
BEGIN:VFREEBUSY
ORGANIZER:mailto:jsmith@example.com
DTSTART:19980313T141711Z
DTEND:19980410T141711Z
FREEBUSY:19980314T233000Z/19980315T003000Z
FREEBUSY:19980316T153000Z/19980316T163000Z
FREEBUSY:19980318T030000Z/19980318T040000Z
URL:http://www.example.com/calendar/busytime/jsmith.ifb
END:VFREEBUSY
END:VCALENDAR

```

## 5. Recommended Practices

These recommended practices should be followed in order to assure consistent handling of the following cases for an iCalendar object.

1. Content lines longer than 75 octets SHOULD be folded.
2. When the combination of the "RRULE" and "RDATE" properties in a recurring component produces multiple instances having the same start DATE-TIME value, they should be collapsed to, and considered as, a single instance. If the "RDATE" property is specified as a PERIOD value the duration of the recurrence instance will be the one specified by the "RDATE" property, and not the duration of the recurrence instance defined by the "DTSTART" property.
3. When a calendar user receives multiple requests for the same calendar component (e.g., REQUEST for a "VEVENT" calendar

component) as a result of being on multiple mailing lists specified by "ATTENDEE" properties in the request, they SHOULD respond to only one of the requests. The calendar user SHOULD also specify (using the "MEMBER" parameter of the "ATTENDEE" property) of which mailing list they are a member.

4. An implementation can truncate a "SUMMARY" property value to 255 octets, but it MUST NOT truncate the value in the middle of a UTF-8 multi-octet sequence.
5. If seconds of the minute are not supported by an implementation, then a value of "00" SHOULD be specified for the seconds component in a time value.
6. "TZURL" values SHOULD NOT be specified as a file URI type. This URI form can be useful within an organization, but is problematic in the Internet.
7. Some possible English values for "CATEGORIES" property include: "ANNIVERSARY", "APPOINTMENT", "BUSINESS", "EDUCATION", "HOLIDAY", "MEETING", "MISCELLANEOUS", "NON-WORKING HOURS", "NOT IN OFFICE", "PERSONAL", "PHONE CALL", "SICK DAY", "SPECIAL OCCASION", "TRAVEL", "VACATION". Categories can be specified in any registered language.
8. Some possible English values for the "RESOURCES" property include: "CATERING", "CHAIRS", "COMPUTER PROJECTOR", "EASEL", "OVERHEAD PROJECTOR", "SPEAKER PHONE", "TABLE", "TV", "VCR", "VIDEO PHONE", "VEHICLE". Resources can be specified in any registered language.

## 6. Internationalization Considerations

Applications MUST generate iCalendar streams in the UTF-8 charset and MUST accept an iCalendar stream in the UTF-8 or US-ASCII charset.

## 7. Security Considerations

Because calendaring and scheduling information is very privacy-sensitive, the protocol used for the transmission of calendaring and scheduling information should have capabilities to protect the information from possible threats, such as eavesdropping, replay, message insertion, deletion, modification, and man-in-the-middle attacks.

As this document only defines the data format and media type of text/calendar that is independent of any calendar service or protocol, it is up to the actual protocol specifications such as iTIP [2446bis],

iMIP [2447bis], and "Calendaring Extensions to WebDAV (CalDAV)" [RFC4791] to describe the threats that the above attacks present, as well as ways in which to mitigate them.

## 8. IANA Considerations

### 8.1. iCalendar Media Type Registration

The Calendaring and Scheduling Core Object Specification is intended for use as a MIME content type.

To: ietf-types@iana.org

Subject: Registration of media type text/calendar

Type name: text

Subtype name: calendar

Required parameters: none

Optional parameters: charset, method, component, and optinfo

The "charset" parameter is defined in [RFC2046] for subtypes of the "text" media type. It is used to indicate the charset used in the body part. The charset supported by this revision of iCalendar is UTF-8. The use of any other charset is deprecated by this revision of iCalendar; however, note that this revision requires that compliant applications **MUST** accept iCalendar streams using either the UTF-8 or US-ASCII charset.

The "method" parameter is used to convey the iCalendar object method or transaction semantics for the calendaring and scheduling information. It also is an identifier for the restricted set of properties and values of which the iCalendar object consists. The parameter is to be used as a guide for applications interpreting the information contained within the body part. It **SHOULD NOT** be used to exclude or require particular pieces of information unless the identified method definition specifically calls for this behavior. Unless specifically forbidden by a particular method definition, a text/calendar content type can contain any set of properties permitted by the Calendaring and Scheduling Core Object Specification. The "method" parameter **MUST** be specified and **MUST** be set to the same value as the "METHOD" component property of the iCalendar objects of the iCalendar stream if and only if the iCalendar objects in the iCalendar stream all have a "METHOD" component property set to the same value.

The value for the "method" parameter is defined as follows:

```
method = 1*(ALPHA / DIGIT / "-")  
; IANA-registered iCalendar object method
```

The "component" parameter conveys the type of iCalendar calendar component within the body part. If the iCalendar object contains more than one calendar component type, then multiple component parameters MUST be specified.

The value for the "component" parameter is defined as follows:

```
component = "VEVENT"  
           / "VTODO"  
           / "VJOURNAL"  
           / "VFREEBUSY"  
           / "VTIMEZONE"  
           / iana-token  
           / x-name
```

The "optinfo" parameter conveys optional information about the iCalendar object within the body part. This parameter can only specify semantics already specified by the iCalendar object and that can be otherwise determined by parsing the body part. In addition, the optional information specified by this parameter MUST be consistent with that information specified by the iCalendar object. For example, it can be used to convey the "Attendee" response status to a meeting request. The parameter value consists of a string value.

The parameter can be specified multiple times.

The value for the "optinfo" parameter is defined as follows:

```
optinfo    = infovalue / qinfovalue  
  
infovalue  = iana-token / x-name  
  
qinfovalue = DQUOTE (infovalue) DQUOTE
```

Encoding considerations: This media type can contain 8bit characters, so the use of quoted-printable or base64 MIME Content-Transfer-Encodings might be necessary when iCalendar objects are transferred across protocols restricted to the 7bit repertoire. Note that a text valued property in the content entity can also have content encoding of special characters using a BACKSLASH character escapement technique. This means that content values can end up being encoded twice.

Security considerations: See Section 7.

Interoperability considerations: This media type is intended to define a common format for conveying calendaring and scheduling information between different systems. It is heavily based on the earlier [VCAL] industry specification.

Published specification: This specification.

Applications that use this media type: This media type is designed for widespread use by Internet calendaring and scheduling applications. In addition, applications in the workflow and document management area might find this content-type applicable. The iTIP [2446bis], iMIP [2447bis], and CalDAV [RFC4791] Internet protocols directly use this media type also.

Additional information:

Magic number(s): None.

File extension(s): The file extension of "ics" is to be used to designate a file containing (an arbitrary set of) calendaring and scheduling information consistent with this MIME content type.

The file extension of "ifb" is to be used to designate a file containing free or busy time information consistent with this MIME content type.

Macintosh file type code(s): The file type code of "iCal" is to be used in Apple MacIntosh operating system environments to designate a file containing calendaring and scheduling information consistent with this MIME media type.

The file type code of "iFBf" is to be used in Apple MacIntosh operating system environments to designate a file containing free or busy time information consistent with this MIME media type.

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

## 8.2. New iCalendar Elements Registration

This section defines the process to register new or modified iCalendar elements, that is, components, properties, parameters, value data types, and values, with IANA.

### 8.2.1. iCalendar Elements Registration Procedure

The IETF will create a mailing list, `icalendar@ietf.org`, which can be used for public discussion of iCalendar elements proposals prior to registration. Use of the mailing list is strongly encouraged. The IESG will appoint a designated expert who will monitor the `icalendar@ietf.org` mailing list and review registrations.

Registration of new iCalendar elements MUST be reviewed by the designated expert and published in an RFC. A Standards Track RFC is REQUIRED for the registration of new value data types that modify existing properties, as well as for the registration of participation status values to be used in "VEVENT" calendar components. A Standards Track RFC is also REQUIRED for registration of iCalendar elements that modify iCalendar elements previously documented in a Standards Track RFC.

The registration procedure begins when a completed registration template, defined in the sections below, is sent to `icalendar@ietf.org` and `iana@iana.org`. The designated expert is expected to tell IANA and the submitter of the registration within two weeks whether the registration is approved, approved with minor changes, or rejected with cause. When a registration is rejected with cause, it can be re-submitted if the concerns listed in the cause are addressed. Decisions made by the designated expert can be appealed to the IESG Applications Area Director, then to the IESG. They follow the normal appeals procedure for IESG decisions.

### 8.2.2. Registration Template for Components

A component is defined by completing the following template.

Component name: The name of the component.

Purpose: The purpose of the component. Give a short but clear description.

Format definition: The ABNF for the component definition needs to be specified.



**Description:** Any special notes about the component, how it is to be used, etc.

**Example(s):** One or more examples of instances of the component need to be specified.

### **8.2.3. Registration Template for Properties**

A property is defined by completing the following template.

**Property name:** The name of the property.

**Purpose:** The purpose of the property. Give a short but clear description.

**Value type:** Any of the valid value types for the property value need to be specified. The default value type also needs to be specified.

**Property parameters:** Any of the valid property parameters for the property **MUST** be specified.

**Conformance:** The calendar components in which the property can appear **MUST** be specified.

**Description:** Any special notes about the property, how it is to be used, etc.

**Format definition:** The ABNF for the property definition needs to be specified.

**Example(s):** One or more examples of instances of the property need to be specified.

### **8.2.4. Registration Template for Parameters**

A parameter is defined by completing the following template.

**Parameter name:** The name of the parameter.

**Purpose:** The purpose of the parameter. Give a short but clear description.

**Format definition:** The ABNF for the parameter definition needs to be specified.

**Description:** Any special notes about the parameter, how it is to be used, etc.

Example(s): One or more examples of instances of the parameter need to be specified.

#### **8.2.5. Registration Template for Value Data Types**

A value data type is defined by completing the following template.

Value name: The name of the value type.

Purpose: The purpose of the value type. Give a short but clear description.

Format definition: The ABNF for the value type definition needs to be specified.

Description: Any special notes about the value type, how it is to be used, etc.

Example(s): One or more examples of instances of the value type need to be specified.

#### **8.2.6. Registration Template for Values**

A value is defined by completing the following template.

Value: The value literal.

Purpose: The purpose of the value. Give a short but clear description.

Conformance: The calendar properties and/or parameters that can take this value need to be specified.

Example(s): One or more examples of instances of the value need to be specified.

The following is a fictitious example of a registration of an iCalendar value:

Value: TOP-SECRET

Purpose: This value is used to specify the access classification of top-secret calendar components.

Conformance: This value can be used with the "CLASS" property.

Example(s): The following is an example of this value used with the "CLASS" property:

CLASS:TOP-SECRET

### 8.3. Initial iCalendar Elements Registries

The IANA created and maintains the following registries for iCalendar elements with pointers to appropriate reference documents.

#### 8.3.1. Components Registry

The following table has been used to initialize the components registry.

Component	Status	Reference
VCALENDAR	Current	RFC 5545, Section 3.4
VEVENT	Current	RFC 5545, Section 3.6.1
VTODO	Current	RFC 5545, Section 3.6.2
VJOURNAL	Current	RFC 5545, Section 3.6.3
VFREEBUSY	Current	RFC 5545, Section 3.6.4
VTIMEZONE	Current	RFC 5545, Section 3.6.5
VALARM	Current	RFC 5545, Section 3.6.6
STANDARD	Current	RFC 5545, Section 3.6.5
DAYLIGHT	Current	RFC 5545, Section 3.6.5

### 8.3.2. Properties Registry

The following table is has been used to initialize the properties registry.

Property	Status	Reference
CALSCALE	Current	RFC 5545, Section 3.7.1
METHOD	Current	RFC 5545, Section 3.7.2
PRODID	Current	RFC 5545, Section 3.7.3
VERSION	Current	RFC 5545, Section 3.7.4
ATTACH	Current	RFC 5545, Section 3.8.1.1
CATEGORIES	Current	RFC 5545, Section 3.8.1.2
CLASS	Current	RFC 5545, Section 3.8.1.3
COMMENT	Current	RFC 5545, Section 3.8.1.4
DESCRIPTION	Current	RFC 5545, Section 3.8.1.5
GEO	Current	RFC 5545, Section 3.8.1.6
LOCATION	Current	RFC 5545, Section 3.8.1.7
PERCENT-COMPLETE	Current	RFC 5545, Section 3.8.1.8
PRIORITY	Current	RFC 5545, Section 3.8.1.9
RESOURCES	Current	RFC 5545, Section 3.8.1.10
STATUS	Current	RFC 5545, Section 3.8.1.11
SUMMARY	Current	RFC 5545, Section 3.8.1.12
COMPLETED	Current	RFC 5545, Section 3.8.2.1
DTEND	Current	RFC 5545, Section 3.8.2.2
DUE	Current	RFC 5545, Section 3.8.2.3
DTSTART	Current	RFC 5545, Section 3.8.2.4
DURATION	Current	RFC 5545, Section 3.8.2.5

FREEBUSY	Current	RFC 5545, Section 3.8.2.6
TRANSP	Current	RFC 5545, Section 3.8.2.7
TZID	Current	RFC 5545, Section 3.8.3.1
TZNAME	Current	RFC 5545, Section 3.8.3.2
TZOFFSETFROM	Current	RFC 5545, Section 3.8.3.3
TZOFFSETTO	Current	RFC 5545, Section 3.8.3.4
TZURL	Current	RFC 5545, Section 3.8.3.5
ATTENDEE	Current	RFC 5545, Section 3.8.4.1
CONTACT	Current	RFC 5545, Section 3.8.4.2
ORGANIZER	Current	RFC 5545, Section 3.8.4.3
RECURRENCE-ID	Current	RFC 5545, Section 3.8.4.4
RELATED-TO	Current	RFC 5545, Section 3.8.4.5
URL	Current	RFC 5545, Section 3.8.4.6
UID	Current	RFC 5545, Section 3.8.4.7
EXDATE	Current	RFC 5545, Section 3.8.5.1
EXRULE	Deprecated	[RFC2445], Section 4.8.5.2
RDATE	Current	RFC 5545, Section 3.8.5.2
RRULE	Current	RFC 5545, Section 3.8.5.3
ACTION	Current	RFC 5545, Section 3.8.6.1
REPEAT	Current	RFC 5545, Section 3.8.6.2
TRIGGER	Current	RFC 5545, Section 3.8.6.3
CREATED	Current	RFC 5545, Section 3.8.7.1
DTSTAMP	Current	RFC 5545, Section 3.8.7.2
LAST-MODIFIED	Current	RFC 5545, Section 3.8.7.3

SEQUENCE	Current	RFC 5545, Section 3.8.7.4
REQUEST-STATUS	Current	RFC 5545, Section 3.8.8.3

### 8.3.3. Parameters Registry

The following table has been used to initialize the parameters registry.

Parameter	Status	Reference
ALTREP	Current	RFC 5545, Section 3.2.1
CN	Current	RFC 5545, Section 3.2.2
CUTYPE	Current	RFC 5545, Section 3.2.3
DELEGATED-FROM	Current	RFC 5545, Section 3.2.4
DELEGATED-TO	Current	RFC 5545, Section 3.2.5
DIR	Current	RFC 5545, Section 3.2.6
ENCODING	Current	RFC 5545, Section 3.2.7
FMTTYPE	Current	RFC 5545, Section 3.2.8
FBTYPE	Current	RFC 5545, Section 3.2.9
LANGUAGE	Current	RFC 5545, Section 3.2.10
MEMBER	Current	RFC 5545, Section 3.2.11
PARTSTAT	Current	RFC 5545, Section 3.2.12
RANGE	Current	RFC 5545, Section 3.2.13
RELATED	Current	RFC 5545, Section 3.2.14
RELTYPE	Current	RFC 5545, Section 3.2.15
ROLE	Current	RFC 5545, Section 3.2.16
RSVP	Current	RFC 5545, Section 3.2.17

SENT-BY	Current	RFC 5545, Section 3.2.18
TZID	Current	RFC 5545, Section 3.2.19
VALUE	Current	RFC 5545, Section 3.2.20

#### 8.3.4. Value Data Types Registry

The following table has been used to initialize the value data types registry.

Value Data Type	Status	Reference
BINARY	Current	RFC 5545, Section 3.3.1
BOOLEAN	Current	RFC 5545, Section 3.3.2
CAL-ADDRESS	Current	RFC 5545, Section 3.3.3
DATE	Current	RFC 5545, Section 3.3.4
DATE-TIME	Current	RFC 5545, Section 3.3.5
DURATION	Current	RFC 5545, Section 3.3.6
FLOAT	Current	RFC 5545, Section 3.3.7
INTEGER	Current	RFC 5545, Section 3.3.8
PERIOD	Current	RFC 5545, Section 3.3.9
RECUR	Current	RFC 5545, Section 3.3.10
TEXT	Current	RFC 5545, Section 3.3.11
TIME	Current	RFC 5545, Section 3.3.12
URI	Current	RFC 5545, Section 3.3.13
UTC-OFFSET	Current	RFC 5545, Section 3.3.14

### 8.3.5. Calendar User Types Registry

The following table has been used to initialize the calendar user types registry.

Calendar User Type	Status	Reference
INDIVIDUAL	Current	RFC 5545, Section 3.2.3
GROUP	Current	RFC 5545, Section 3.2.3
RESOURCE	Current	RFC 5545, Section 3.2.3
ROOM	Current	RFC 5545, Section 3.2.3
UNKNOWN	Current	RFC 5545, Section 3.2.3

### 8.3.6. Free/Busy Time Types Registry

The following table has been used to initialize the free/busy time types registry.

Free/Busy Time Type	Status	Reference
FREE	Current	RFC 5545, Section 3.2.9
BUSY	Current	RFC 5545, Section 3.2.9
BUSY-UNAVAILABLE	Current	RFC 5545, Section 3.2.9
BUSY-TENTATIVE	Current	RFC 5545, Section 3.2.9



### 8.3.7. Participation Statuses Registry

The following table has been used to initialize the participation statuses registry.

Participant Status	Status	Reference
NEEDS-ACTION	Current	RFC 5545, Section 3.2.12
ACCEPTED	Current	RFC 5545, Section 3.2.12
DECLINED	Current	RFC 5545, Section 3.2.12
TENTATIVE	Current	RFC 5545, Section 3.2.12
DELEGATED	Current	RFC 5545, Section 3.2.12
COMPLETED	Current	RFC 5545, Section 3.2.12
IN-PROCESS	Current	RFC 5545, Section 3.2.12

### 8.3.8. Relationship Types Registry

The following table has been used to initialize the relationship types registry.

Relationship Type	Status	Reference
CHILD	Current	RFC 5545, Section 3.2.15
PARENT	Current	RFC 5545, Section 3.2.15
SIBLING	Current	RFC 5545, Section 3.2.15

### 8.3.9. Participation Roles Registry

The following table has been used to initialize the participation roles registry.

Role Type	Status	Reference
CHAIR	Current	RFC 5545, Section 3.2.16
REQ-PARTICIPANT	Current	RFC 5545, Section 3.2.16
OPT-PARTICIPANT	Current	RFC 5545, Section 3.2.16
NON-PARTICIPANT	Current	RFC 5545, Section 3.2.16

### 8.3.10. Actions Registry

The following table has been used to initialize the actions registry.

Action	Status	Reference
AUDIO	Current	RFC 5545, Section 3.8.6.1
DISPLAY	Current	RFC 5545, Section 3.8.6.1
EMAIL	Current	RFC 5545, Section 3.8.6.1
PROCEDURE	Deprecated	[RFC2445], Section 4.8.6.1

### 8.3.11. Classifications Registry

The following table has been used to initialize the classifications registry.

Classification	Status	Reference
PUBLIC	Current	RFC 5545, Section 3.8.1.3
PRIVATE	Current	RFC 5545, Section 3.8.1.3
CONFIDENTIAL	Current	RFC 5545, Section 3.8.1.3

### 8.3.12. Methods Registry

No values are defined in this document for the "METHOD" property.

## 9. Acknowledgments

The editor of this document wishes to thank Frank Dawson and Derik Stenerson, the original authors of RFC 2445, as well as the following individuals who have participated in the drafting, review, and discussion of this memo:

Joe Abley, Hervey Allen, Steve Allen, Jay Batson, Oliver Block, Stephane Bortzmeyer, Chris Bryant, Tantek Celik, Mark Crispin, Cyrus Daboo, Mike Douglass, Andrew N. Dowden, Lisa Dusseault, Lars Eggert, Gren Eliot, Pasi Eronen, Ben Fortuna, Ned Freed, Neal Gafter, Ted Hardie, Tim Hare, Jeffrey Harris, Helge Hess, Paul B. Hill, Thomas Hnetila, Russ Housley, Leif Johansson, Ciny Joy, Bruce Kahn, Reinhold Kainhofer, Martin Kiff, Patrice Lapierre, Michiel van Leeuwen, Jonathan Lennox, Jeff McCullough, Bill McQuillan, Alexey Melnikov, John W. Noerenberg II, Chuck Norris, Mark Paterson, Simon Pilette, Arnaud Quillaud, Robert Ransdell, Julian F. Reschke, Caleb Richardson, Sam Roberts, Dan Romascanu, Mike Samuel, George Sexton, Nigel Swinson, Clint Talbert, Simon Vaillancourt, Magnus Westerlund, and Sandy Wills.

A special thanks to the working group chairs Aki Niemi and Eliot Lear for their support and guidance.

The editor would also like to thank the Calendaring and Scheduling Consortium for advice with this specification, and for organizing interoperability testing events to help refine it.

## 10. References

### 10.1. Normative References

- [ISO.8601.2004] International Organization for Standardization, "Data elements and interchange formats -- Information interchange -- Representation of dates and times", 2004.
- [ISO.9070.1991] International Organization for Standardization, "Information Technology\_SGML Support Facilities -- Registration Procedures for Public Text Owner Identifiers, Second Edition", April 1991.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2368] Hoffman, P., Masinter, L., and J. Zawinski, "The mailto URL scheme", RFC 2368, July 1998.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

RFC 5545

iCalendar

September 2009

- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5646] Phillips, A., Ed., and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.
- [US-ASCII] American National Standards Institute, "Coded Character Set - 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.

## 10.2. Informative References

- [2446bis] Daboo, C., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", Work in Progress, April 2009.
- [2447bis] Melnikov, A., "iCalendar Message-Based Interoperability Protocol (iMIP)", Work in Progress, June 2008.
- [ANSI INCITS 61-1986] International Committee for Information Technology, "Representation of Geographic Point Locations for Information Interchange (formerly ANSI X3.61-1986 (R1997))", ANSI INCITS 61-1986 (R2007), 2007.
- [RFC1738] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, August 1998.
- [RFC2397] Masinter, L., "The "data" URL scheme", RFC 2397, August 1998.
- [RFC2425] Howes, T., Smith, M., and F. Dawson, "A MIME Content-Type for Directory Information", RFC 2425, September 1998.
- [RFC2426] Dawson, F. and T. Howes, "vCard MIME Directory Profile", RFC 2426, September 1998.

RFC 5545

iCalendar

September 2009

- [RFC2445] Dawson, F. and Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.
- [RFC4516] Smith, M. and T. Howes, "Lightweight Directory Access Protocol (LDAP): Uniform Resource Locator", RFC 4516, June 2006.
- [RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault, "Calendaring Extensions to WebDAV (CalDAV)", RFC 4791, March 2007.
- [TZDB] Eggert, P. and A.D. Olson, "Sources for Time Zone and Daylight Saving Time Data", July 2009, <<http://www.twinsun.com/tz/tz-link.htm>>.
- [VCAL] Internet Mail Consortium, "vCalendar: The Electronic Calendaring and Scheduling Exchange Format", September 1996, <<http://www.imc.org/pdi/vcal-10.txt>>.

## **Appendix A. Differences from RFC 2445**

This appendix contains a list of changes that have been made in the Internet Calendaring and Scheduling Core Object Specification from RFC 2445.

### **A.1. New Restrictions**

1. The "DTSTART" property SHOULD be synchronized with the recurrence rule, if specified.
2. The "RRULE" property SHOULD NOT occur more than once in a component.
3. The BYHOUR, BYMINUTE, and BYSECOND rule parts MUST NOT be specified in the "RRULE" property when the "DTSTART" property is specified as a DATE value.
4. The value type of the "DTEND" or "DUE" properties MUST match the value type of "DTSTART" property.
5. The "DURATION" property can no longer appear in "VFREEBUSY" components.

### **A.2. Restrictions Removed**

1. The "DTSTART" and "DTEND" properties are no longer required to be specified as date with local time and time zone reference when used with a recurrence rule.

### **A.3. Deprecated Features**

1. The "EXRULE" property can no longer be specified in a component.
2. The "THISANDPRIOR" value can no longer be used with the "RANGE" parameter.
3. The "PROCEDURE" value can no longer be used with the "ACTION" property.
4. The value type RECUR no longer allows multiple values to be specified by a COMMA-separated list of values.
5. x-name rule parts can no longer be specified in properties of RECUR value type (e.g., "RRULE"). x-param can be used on RECUR value type properties instead.

RFC 5545

iCalendar

September 2009

## Author's Address

Bernard Desruisseaux (editor)  
Oracle Corporation  
600 blvd. de Maisonneuve West  
Suite 1900  
Montreal, QC H3A 3J2  
CANADA

EMail: [bernard.desruisseaux@oracle.com](mailto:bernard.desruisseaux@oracle.com)  
URI: <http://www.oracle.com/>