Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 26, 2013

E. York, Ed.
C. Daboo, Ed.
Apple Inc.
M. Douglass, Ed.
RPI

January 22, 2013

VPOLL: Consensus Scheduling Component for iCalendar
draft-york-vpoll-00

Abstract

This specification introduces a new iCalendar component which allows for consensus scheduling, that is voting on a number of alternative meeting or task alternatives.

Status of this Memo

This Internet-Draft will expire on July 26, 2013.

Copyright Notice

Table of Contents

1. Introduction

The currently existing approach to agreeing on meeting times using iTip [RFC5546] and/or iMip [RFC6047] have some significant failings. There is no useful bargaining or suggestion mechanism in iTip, only the ability for a potential attendee to accept or refuse or to counter with a time of their own choosing.

Part of the problem is that for many potential attendees, their freebusy is not an accurate representation of their avalability. In fact, when trying to schedule conference calls across different organizations, attendees may not be allowed to provide freebusy information or availability as this may reveal something of the organizations internal activities.

A number of studies have shown that large amounts of time are spent trying to come to an agreement - up to and beyond 20 working hours per meeting. many organizers fall back on other approachessuch as phone calls and email to determine a suitable time.

Online services have appeared as a result and these allow participants to vote on a number of alternatives without revealing or using freebusy or availability. When agreement is reached a conventional scheduling message may be sent to the attendees. This approach appears to reach consensus fairly rapidly. Peer pressure may have some bearing on this as all voters are able to see the current state of the voting and may adjust their own meeting schedules to make themselves available for a popular choice.

The component and properties defined in this specification provide a standardized structure to this process and allow calendar clients and servers and these web based services to interact.

These structures also have uses beyond the relatively simple needs of most meeting organizers. The process of coming to consensus can also be viewed as a bidding process.

2. Conventions and Terms Used in This Document

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

Additionally we will use the following terms:

Consensus Scheduling: The process whereby we come to some agreement on meeting or task alternatives and then book that meeting or task.

Active Vpoll: A vpoll may have a DTSTART, DTEND and DURATION which may define the start and end of the active voting period.

Voter: A participant who votes on th ealternatives. A voter need not be an attendee of any of the alternatives presented.

3. Simple Consensus Scheduling

This specification defines components and properties which can be used for simple consensus scheduling but also have the generality to handle more complex cases. To provide an easy (and for many - sufficient) introduction to consensus scheduling and VPOLL we will outline the flow of information for the simple case of voting on a number of meeting alternatives which differ only in time. In addition the voters will all be potential attendees.

This specification not only defines data structures but adds a new iTip method used when consensus has been reached. we will show how a VPOLL object is used to inform voters of the state of a simple vote on some alternatives.

3.1. The VPOLL Component: An Overview

The VPOLL component acts as a wrapper for a number of alternatives to be voted on together with some properties used to maintain the state of the voting. For our simple example the following VPOLL properties and sub-components are either required or appropriate:

DTSTAMP: The usual [RFC5545] property.

SEQUENCE: The usual [RFC5545] property. See below for SEQUENCE behavior.

UID: The usual [RFC5545] property.

ORGANIZER: The usual [RFC5545] property. In general this need not be an organizer of any of the alternatives. In this simple outline we assume it is the same.

SUMMARY: The usual [RFC5545] property. This optional but recommended property provides the a short title to the poll.

DESCRIPTION: The usual [RFC5545] property. This optional property provides more details.

DTEND: The usual [RFC5545] property. This optional property provides a poll closing time and date after which the VPOLL is no longer active.

POLL-MODE: A new property which defines how the votes are used to obtain a result. For our use case it will take the value "BASIC" meaning one event will be chosen from the alternatives.

POLL-PROPERTIES: A new property which defines which icalendar properties are being voted on. For our use case it will take the value "DTSTART, LOCATION" meaning only those properties are significant for voting. Other properties in the events may differ but are not considered significant for the voting process.

VOTER: A new property. There is one of these for each voter and it is similar to the [RFC5545] ATTENDEE property. In the VPOLL component it shows who takes part in the voting.

VEVENT: In our simple use case there will be multiple VEVENT sub-components defining the alternatives. Each will have adifferent date and or time for the meeting.

Putting that together we can construct an example VPOLL with 3 voters and 3 alternative meetings:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example//Example
METHOD:REQUEST
BEGIN:VPOLL
POLL-MODE:BASIC
POLL-PROPERTIES:DTSTART,LOCATION
ORGANIZER:mailto:mike@example.com
UID:sched01-1234567890
DTSTAMP:20120101T000000Z
SUMMARY:What to do this week
DTEND:20120101T000000Z
VOTER:mailto:cyrus@exmaple.com
VOTER:mailto:eric@example.com
VEVENT.......(with a poll-item-id=1)
VEVENT.......(with a poll-item-id=2)
VEVENT.......(with a poll-item-id=3)
END:VPOLL
END:VCALENDAR
```

As can be seen in the example above, there is an iTip METHOD property with the value REQUEST. The VPOLL object will be distributed to all the voters, either through iMipor through some VPOLL enabled service.

3.2. The VPOLL Subcomponents: An Overview

Within the VPOLL component we have the alternatives to vote on. In many respects these are standard [RFC5545] components. For our simple use case they are all VEVENT components. In addition to the usual [RFC5545] properties some extra properties are used for a VPOLL.

POLL-ITEM-ID: This provides a unique reference to the sub-component within the VPOLL. It’s value SHOULD be a small integer.

VOTER: The VOTER property within a VPOLL sub-component specifies the state of the vote for that voter. The RESPONSE parameter supplies the current vote in the range 0 to 100. For many purposes this is too fine grained and recommended values are defined for voting preferences of "NO", "MAYBE" and "YES" with an additional "YES but not the preferred outcome". These properties will be added to the sub-components as responses appear.

3.3. VPOLL responses

Upon receipt of a VPOLL REQUEST the voter will reply with a VPOLL component containing their vote. In our simple case it will have the following properties:

DTSTAMP: The usual [RFC5545] property.

SEQUENCE: The usual [RFC5545] property. See below for SEQUENCE behavior.

UID: Same as the request.

ORGANIZER: Same as the request.

SUMMARY: Same as the request.

VOTER: One only - the voter replying.

POLL-ITEM-ID: One per item being voted on. There does not need to be one for each sub-component but each REPLY will set the voting state for every sub-component.

Note that a voter can send a number of REPLYs for each REQUEST sent by the organizer. Each REPLY completely replaces the voting record for that voter for all components being voted on. In our example, if Eric respondes and votre for items 1 and 2 and then responds again with a vote for only item 3 the final outcome is one vote on item 3.

Putting this together we can construct an example REPLY VPOLL from Cyrus:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example//Example
METHOD: REPLY
BEGIN:VPOLL
ORGANIZER:mailto:douglm@example.com
VOTER:mailto:cyrus@example.com
UID:sched01-1234567890
DTSTAMP:20120101T010000Z
SUMMARY:What to do this week
POLL-ITEM-ID;RESPONSE=50;PUBLIC-COMMENT=Work on iTIP:1
POLL-ITEM-ID;RESPONSE=100;COMMENT=Work on WebDAV:2
POLL-ITEM-ID;RESPONSE=0:3
END:VPOLL
END:VCALENDAR
```

3.4. VPOLL updates

When the organizer receives a response from one or more voters the current state of the poll is sent to all voters. The new iTip method POLLSTATUS is used. The VPOLL can contain a reduced set of properties but MUST contain DTSTAMP, SEQUENCE (if not 0), UID, ORGANIZER and VOTER.

In addition, for our use case, it will contain skeleton VEVENT sub-components with the POLL-ITEM-ID and VOTER properties. Clients receiving this poll status SHOULD merge it in to their own copy to give the full current status

An example:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example//Example
METHOD: POLLSTATUS
BEGIN:VPOLL
ORGANIZER:mailto:douglm@example.com
VOTER:mailto:cyrus@example.com
VOTER:mailto:eric@example.com
UID:sched01-1234567890
DTSTAMP:20120101T020000Z
SEQUENCE:0
SUMMARY:What to do this week
BEGIN:VEVENT
POLL-ITEM-ID: 1
VOTER;RESPONSE=50;COMMENT=Work on iTIP:
 mailto:cyrus@exmaple.com
VOTER;RESPONSE=100:mailto:eric@example.com
END:VEVENT
BEGIN:VEVENT
POLL-ITEM-ID: 2
VOTER;RESPONSE=100;COMMENT=Work on WebDAV:
 mailto:cyrus@exmaple.com
VOTER;RESPONSE=100:mailto:eric@example.com
END:VEVENT
BEGIN:VEVENT
POLL-ITEM-ID: 3
VOTER;RESPONSE=0:mailto:cyrus@example.com
VOTER;RESPONSE=0:mailto:eric@example.com
END:VEVENT
END:VPOLL
END:VCALENDAR
```

3.5. VPOLL Completion

After a number of REPLY messages have been received the poll will be considered complete. If there is a DTEND on the poll the system may automatically close the poll, or the organizer may, at any time, consider the poll complete.

The poll is completed by sending a final iTip message with the new CONFIRM method. In this case the VPOLL component contains ...

The VPOLL conformation example:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example//Example
METHOD: CONFIRM
BEGIN:VPOLL
ORGANIZER:mailto:douglm@example.com
UID:sched01-1234567890
DTSTAMP:20120101T030000Z
COMPLETED:20120101T030000Z
SEQUENCE:0
SUMMARY:What to do this week
BEGIN:VEVENT
UID:123
......
END:VEVENT
```

4. iCalendar Extensions

4.1. New Property Parameters

4.1.1. Public-Comment

Parameter name: PUBLIC-COMMENT

Purpose: To allow voters to provide a public comment in their reponse.

Format Definition:

This parameter is defined by the following notation:

```
public-comment-param = "PUBLIC-COMMENT=" DQUOTE text DQUOTE
```

Description: This parameter can be specified on the POLL-ITEM-ID property to allow a voter to add some commentary visible to all voters.

4.1.2. Response

Parameter name: RESPONSE

Purpose: To specify a response vote.

Format Definition:

This parameter is defined by the following notation:

```
 responseparam = "RESPONSE=" integer
       ; integer value 0..100
```

Description: This parameter can be specified on the POLL-ITEM-ID property to provide the value of the voters response. This parameter allows for fine grained responses which are appropriate to some applications. For the case of individuals voting for a choice of events, client applications SHOULD conform to the following convention:

- 0 - 39 A "NO vote".

- 40 - 79 A "MAYBE" vote

- 80 - 89 A "YES - but not preferred vote"

- 90-100 A "YES" vote.

Clients MUST preserve the responseparam value when there is no change from the user even if they have a UI with fixed states (e.g. yes/no/maybe).

4.1.3. Stay-Informed

Parameter name: STAY-INFORMED

Purpose: To specify the voter also wants to be added as an ATTENDEE when the poll is confirmed.

Format Definition:

This parameter is defined by the following notation:

```
 stayinformedparam = "STAY-INFORMED" "=" boolean
```

Description: This parameter MAY be specified on VOTER and, if the value is TRUE, indicates the voter wishes to be added to the final choice as a non participant.

4.2. New Properties

4.2.1. Accept-Response

Property name: ACCEPT-RESPONSE

Purpose: This property is used in VPOLL to indicate the types of component that may be supplied in a response.

Property Parameters: Non-standard or iana parameters can be specified on this property.

Conformance: This property MAY be specified in a VPOLL component.

Description: When used in a VPOLL this property indicates what allowable component types may be returned in a reply. Typically this would allow a voter to respond with the freebusy or availability rather than choosing one of the presented alternatives

If this property is not present voters are only allowed to respond with the same component type as in the request.

Format Definition:

This property is defined by the following notation:

```
acceptresponse = "ACCEPT-RESPONSE" "="
                     acceptresponseparams ":"
                     iana-token ("," iana-token) CRLF

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

4.2.2. Poll-Item-Id

Property name: POLL-ITEM-ID

Purpose: This property is used in VPOLL to indicate recipients of the poll and in VPOLL sub-components to give the voters response.

Value type: The value type for this property is param-value.

Property Parameters: Non-standard, response, public-comment or stayinformed parameters can be specified on this property.

Conformance: This property MAY be specified in a VPOLL component or its sub-components.

Description: In a METHOD:REQUEST each child component MUST have a POLL-ITEM-ID property. Each set of components with the same POLL-ITEM-ID value represents one overall set of items to be voted on.

POLL-ITEM-ID SHOULD be a unique ID for each component or set of components. If it remains the same between REQUESTs then the previous response for that component MAY be re-used. To force a re-vote on a component due to a significant change, the POLL-ITEM-ID MUST change.

In a METHOD:REPLY there is a single VPOLL component which MUST have: either one or more POLL-ITEM-ID properties with a RESPONSE param matching that from a REQUEST or a VFREEBUSY or VAVAILABILITY child component showing overall busy/available time.

Format Definition:

This property is defined by the following notation:

```
        pollitemid = "POLL-ITEM-ID" pollitemdparams ":"
                   param-value CRLF

        pollitemidparams = *(
                    (";" responseparam) /
                    (";" stayinformedparam) /
                    (";" public-comment-param) /
                    (";" other-param)
             )
```

4.2.3. Poll-Mode

Property name: POLL-MODE

Purpose: This property is used in VPOLL to indicate what voting mode is to be applied.

Property Parameters: Non-standard or iana parameters can be specified on this property.

Conformance: This property MAY be specified in a VPOLL component or its sub-components.

Description: The poll mode defines how the votes are applied to obtain a result. BASIC mode, the default, means that the voters are selecting one component (or group of components) with a given POLL=ITEM-ID.

Other polling modes may be defined in updates to this specification. These may allow for such modes as ranking or task assignment.

Format Definition:

This property is defined by the following notation:

```
pollmode = "POLL-MODE" pollmodeparams ":"
    ("BASIC" / iana-token / other-token) CRLF

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

Note If the client does not understand the poll mode in use it should use the LINK property to get the web-based view of the poll.

4.2.4. Poll-properties

Property name: POLL-PROPERTIES

Purpose: This property is used in VPOLL to define which icalendar properties are being voted on.

Property Parameters: Non-standard or iana parameters can be specified on this property.

Conformance: This property MAY be specified in a VPOLL component.

Description: This property defines which icalendar properties are significant in the voting process. It may not be clear to voters which properties are varying in a significant manner. Clients may use this property to highlight those listed properties.

Format Definition:

This property is defined by the following notation:

```
pollproperties = "POLL-PROPERTIES" pollpropparams ":"
    text *("," text) CRLF

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

4.2.5. Voter

Property name: VOTER

Purpose: This property is used in VPOLL to indicate recipients of the poll and in VPOLL sub-components to give the voters response.

Value type: The value type for this property is cal-address.

Property Parameters: Non-standard, cutype, member, role, rsvp, delto, delfrom, sentby, cn, dir, lang, response or stayinformed parameters can be specified on this property.

Conformance: This property MAY be specified in a VPOLL component or its sub-components.

Description: When used in a VPOLL this property indicates who the poll is sent to. When used in sub-components of VPOLL this property indicate the response of each voter as known at that time.

When the Organizer sends a full update (METHOD:REQUEST), then VOTER properties MUST be included in the child components for any voters that have responded.

Format Definition:

This property is defined by the following notation:

```
 voter = "VOTER" voterparams ":" cal-address CRLF

 voterparam   = *(
                ;
                ; The following are OPTIONAL,
                ; but MUST NOT occur more than once.
                ;
                (";" cutypeparam) / (";" memberparam) /
                (";" roleparam) /
                (";" rsvpparam) / (";" deltoparam) /
                (";" delfromparam) / (";" sentbyparam) /
                (";" cnparam) / (";" dirparam) /
                (";" languageparam) /
                (";" responseparam) /
                (";" stayinformedparam) /
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
                ;
                (";" other-param)
                ;
           )
```

Note RSVP=TRUE MAY used by the organizer to force the voter to reset their state and re-vote.

4.3. VPOLL Component

Component name: VPOLL

Purpose: This component provides a mechanism by which voters can vote on provided choices.

Format Definition:

This property is defined by the following notation:

```
pollc    = "BEGIN" ":" "VPOLL" CRLF
             pollprop
             *eventc *todoc *journalc *freebusyc
             *availabilityc *alarmc *iana-comp *x-comp
             "END" ":" "VPOLL" CRLF

pollprop = *(
           ;
           ; The following are REQUIRED,
           ; but MUST NOT occur more than once.
           ;
           dtstamp / uid / organizer /
           ;
           ; The following are OPTIONAL,
           ; but MUST NOT occur more than once.
           ;
           acceptresponse / class / created / completed /
           description / dtstart / last-mod / pollmode /
           pollproperties / priority / seq / status /
           summary / url /
           ;
           ; Either 'dtend' or 'duration' MAY appear in
           ; a 'pollprop', but 'dtend' and 'duration'
           ; MUST NOT occur in the same 'pollprop'.
           ; 'duration' MUST only occur when 'dtstart'
           ; is present
           ;
           dtend / duration /
           ;
           ; The following are OPTIONAL,
           ; and MAY occur more than once.
           ;
           attach / categories / comment /
           contact / link / pollitemid / pollwinner /
           rstatus / related /
           resources / voter / x-prop / iana-prop
           ;
           )
```

Description: This component provides a mechanism by which voters can vote on provided choices. The outcome depends upon the POLL-MODE in effect.

VOTER in VPOLL requests refers to each person who will be voting - RSVP=TRUE is used by the organizer to force the voter to reset their state and re-vote

If specified, the "DTSTART" property defines the start or opening of the poll active period. If absent the poll is presumed to have started when created.

If "DTSTART"is present "DURATION" MAY be specified and indicates the duration, and hence the ending, of the poll. The value of the property MUST be a positive duration.

"DTEND" MAY be specified with or without "DTSTART" and indicates the ending of the poll. If DTEND is specified it MUST be later than the DTSTART or CREATED property.

If one or more VALARM components are included in the VPOLL they are not components to be voted on and MUST NOT contain a POLL-ITEM-ID property. VALARM sub-components may be used to provide warnings to the user when polls are due to start or end.

Need some text to describe what relative alarms are relative to.

5. Poll Modes

The VPOLL component is intended to allow for various forms of polling. The particular form in efffect is indicated by the POLL-MODE property.

To define new poll modes create an RFC defining the new mode and provide the registration information as specified in Section 10.3.

5.1. POLL-MODE:BASIC

BASIC poll mode is the form of voting in which one possible outcome is chosen from a set of possibilities. Usually this will be represented as a number of possible event objects one of which will be selected.

5.1.1. Property restrictions

This poll mode places no special restrictions on properties beyond those specified in the description of the generic VPOLL component.

5.1.2. Outcome reporting

For a METHOD:CONFIRM there must be a single sub-component in the VPOLL object which represents the succesful candidate from the supplied choices.

Winning Non-scheduled VEVENTs or VTODOs MUST be assigned an ORGANIZER property and a list of non-participating ATTENDEEs.

6. iTip Extensions

This specification introduces a number of extensions to [RFC5546]. In group scheduling the parties involved are organizer and attendees. In VPOLL the parties are organizer and voters.

For many of the iTip processing rules the voters take the place of attendees.

6.1. Methods

There are some extensions to the behavior of iTip methods for a VPOLL object and two new methods are defined.

Method	Description
PUBLISH	No changes (yet)
REQUEST	Each child component MUST have a POLL-ITEM-ID property. Each set of components with the same POLL-ITEM-ID value represents one overall set of items to be voted on.
REPLY	There MUST be a single VPOLL component which MUST have: either one or more POLL-ITEM-ID properties with a RESPONSE param matching that from a REQUEST or a VFREEBUSY or VAVAILABILITY child component showing overall busy/available time. The VPOLL MUST have one VOTER only.
ADD	Not supported for VPOLL.
CANCEL	There MUST be a single VPOLL component with UID matching that of the poll being cancelled.

REFRESH	The organizer returns a METHOD:REQUEST with the current full state, or a METHOD:CANCEL or a METHOD:CONFIRM or an error if no matching poll is found.
COUNTER	Not supported for VPOLL.
DECLINECOUNTER	Not supported for VPOLL.
POLLSTATUS	Used to send the current state of the poll to all voters. The VPOLL can contain a reduced set of properties but MUST contain DTSTAMP, SEQUENCE (if not 0), UID, ORGANIZER and VOTER.
CONFIRM	The VPOLL MUST have all the components from the chosen POLL-ITEM-ID set.

The following table shows the above methods broken down by who can send them with VPOLL components.

Originator	Methods
Organizer	PUBLISH, REQUEST, POLLSTATUS
Voter	REPLY, REFRESH, REQUEST (only when delegating)

6.2. Interoperability Models

Most of the standard iTip specification applies with respect to organizer and voters.

6.2.1. Delegation

TBD

6.2.2. Acting on Behalf of Other Calendar Users

TBD

6.2.3. Component Revisions

Need to talk about what a change in SEQUENCE means
Sequence change forces a revote.
New voter - no sequence change
Add another poll set or change poll item ids or any change to a child

component - bump sequence

6.2.4. Message Sequencing

TBD

6.3. Application Protocol Elements

6.3.1. Methods for VPOLL Calendar Components

This section defines the property set restrictions for the method types that are applicable to the "VPOLL" calendar component. Each method is defined using a table that clarifies the property constraints that define the particular method.

The presence column uses the following values to assert whether a property is required or optional, and the number of times it may appear in the iCalendar object.

Presence Value	Description
1	One instance MUST be present.
1+	At least one instance MUST be present.
0	Instances of this property MUST NOT be present.
0+	Multiple instances MAY be present.
0 or 1	Up to 1 instance of this property MAY be present.

The following summarizes the methods that are defined for the "VPOLL" calendar component.

Method	Description
PUBLISH	Post notification of an poll. Used primarily as a method of advertising the existence of a poll.
REQUEST	Make a request for a poll. This is an explicit invitation to one or more voters. Poll requests are also used to update or change an existing poll. Clients that cannot handle REQUEST MAY degrade the poll to view it as a PUBLISH.
REPLY	Reply to a poll request. Voters may set their RESPONSE parameter to supply the current vote in the range 0 to 100.
CANCEL	Cancel a poll.
REFRESH	A request is sent to an Organizer by a Voter asking for the latest version of a poll to be resent to the requester.
POLLSTATUS	Used to send the current state of the poll to all voters. The VPOLL can contain a reduced set of properties but MUST contain DTSTAMP, SEQUENCE (if not 0), UID, ORGANIZER and VOTER.
CONFIRM	Used to signal that the poll has been closed with a choice made.

6.3.1.1. PUBLISH

The "PUBLISH" method in a "VPOLL" calendar component is an unsolicited posting of an iCalendar object. Any CU may add published components to their calendar. The "Organizer" MUST be present in a published iCalendar component. "Voters" MUST NOT be present. Its expected usage is for encapsulating an arbitrary poll as an iCalendar object. The "Organizer" may subsequently update (with another "PUBLISH" method) or cancel (with a "CANCEL" method) a previously published "VPOLL" calendar component.

This method type is an iCalendar object that conforms to the following property constraints:

Constraints for a METHOD:PUBLISH of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST equal PUBLISH.
VPOLL	1+	
DTSTAMP	1	
DTSTART	1	
ORGANIZER	1	
SUMMARY	1	Can be null.
UID	1	
SEQUENCE	0 or 1	MUST be present if value is greater than 0; MAY be present if 0.
ACCEPT-RESPONSE	0 or 1	
ATTACH	0+	
CATEGORIES	0+	
CLASS	0 or 1	
COMMENT	0+	
COMPLETED	0 or 1	
CONTACT	0 or 1	
CREATED	0 or 1	
DESCRIPTION	0 or 1	Can be null.
DTEND	0 or 1	If present, DURATION MUST NOT be present.
DURATION	0 or 1	If present, DTEND MUST NOT be present.
LAST-MODIFIED	0 or 1	
POLL-ITEM-ID	0	
POLL-MODE	0 or 1	
POLL-PROPERTIES	0 or 1	
PRIORITY	0 or 1	
RELATED-TO	0+	
RESOURCES	0+	
STATUS	0 or 1	MAY be one of TENTATIVE/CONFIRMED/CANCELLED.
URL	0 or 1	
IANA-PROPERTY	0+	
X-PROPERTY	0+	
VOTER	0	
REQUEST-STATUS	0	
VALARM	0+	

VEVENT	0+	Depending upon the poll mode in effect there MAY be candidate components included in the poll component. If voting has already taken place, these components MUST include the VOTER property to indicate each voters current response.
VFREEBUSY	0	
VJOURNAL	0+	Depending upon the poll mode in effect there MAY be candidate components included in the poll component. If voting has already taken place, these components MUST include the VOTER property to indicate each voters current response.
VTODO	0+	Depending upon the poll mode in effect there MAY be candidate components included in the poll component. If voting has already taken place, these components MUST include the VOTER property to indicate each voters current response.
VTIMEZONE	0+	MUST be present if any date/time refers to a timezone.
IANA-COMPONENT	0+	
X-COMPONENT	0+	

6.3.1.2. REQUEST

The "REQUEST" method in a "VPOLL" component provides the following scheduling functions:

- Invite "Voters" to respond to the poll.

- Change the items being voted upon.

- Response to a "REFRESH" request.

- Update the details of an existing vpoll.

- Update the status of "Voters".

- Forward a "VPOLL" to another uninvited CU.

- For an existing "VPOLL" calendar component, delegate the role of "Voter" to another CU.

- For an existing "VPOLL" calendar component, change the role of "Organizer" to another CU.

The "Organizer" originates the "REQUEST". The recipients of the "REQUEST" method are the CUs voting in th epoll, the "Voters". "Voters" use the "REPLY" method to convey votesto the "Organizer".

The "UID" and "SEQUENCE" properties are used to distinguish the various uses of the "REQUEST" method. If the "UID" property value in the "REQUEST" is not found on the recipient’s calendar, then the "REQUEST" is for a new "VPOLL" calendar component. If the "UID" property value is found on the recipient’s calendar, then the "REQUEST" is for an update, or a reconfirmation of the "VPOLL" calendar component.

For the "REQUEST" method only a single iCalendar object is permitted.

This method type is an iCalendar object that conforms to the following property constraints:

Constraints for a METHOD:REQUEST of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST be REQUEST.
VPOLL	1	
VOTER	1+	
DTSTAMP	1	
DTSTART	1	
ORGANIZER	1	
SEQUENCE	0 or 1	MUST be present if value is greater than 0; MAY be present if 0.
SUMMARY	1	Can be null.
UID	1	

ACCEPT-RESPONSE	0 or 1	
ATTACH	0+	
CATEGORIES	0+	
CLASS	0 or 1	
COMMENT	0+	
COMPLETED	0 or 1	
CONTACT	0+	
CREATED	0 or 1	
DESCRIPTION	0 or 1	Can be null.
DTEND	0 or 1	If present, DURATION MUST NOT be present.
DURATION	0 or 1	If present, DTEND MUST NOT be present.
GEO	0 or 1	
LAST-MODIFIED	0 or 1	
LOCATION	0 or 1	
POLL-ITEM-ID	0	
POLL-MODE	0 or 1	
POLL-PROPERTIES	0 or 1	
PRIORITY	0 or 1	
RELATED-TO	0+	
REQUEST-STATUS	0	
RESOURCES	0+	
STATUS	0 or 1	MAY be one of TENTATIVE/CONFIRMED.
TRANSP	0 or 1	
URL	0 or 1	
IANA-PROPERTY	0+	
X-PROPERTY	0+	
VALARM	0+	
VTIMEZONE	0+	MUST be present if any date/time refers to a timezone.
IANA-COMPONENT	0+	
X-COMPONENT	0+	
VEVENT	0+	Depending upon the poll mode in effect there MAY be candidate components included in the poll component. If voting has already taken place, these components MUST include the VOTER property to indicate each voters current response.
VFREEBUSY	0	

VJOURNAL	0+	Depending upon the poll mode in effect there MAY be candidate components included in the poll component. If voting has already taken place, these components MUST include the VOTER property to indicate each voters current response.
VTODO	0+	Depending upon the poll mode in effect there MAY be candidate components included in the poll component. If voting has already taken place, these components MUST include the VOTER property to indicate each voters current response.

6.3.1.2.1. Rescheduling a poll

The "REQUEST" method may be used to reschedule a poll, that is force a revote. A rescheduled poll involves a change to the existing poll in terms of its time the components being voted on may have changed. If the recipient CUA of a "REQUEST" method finds that the "UID" property value already exists on the calendar but that the "SEQUENCE" (or "DTSTAMP") property value in the "REQUEST" method is greater than the value for the existing poll, then the "REQUEST" method describes a rescheduling of the poll.

6.3.1.2.2. Updating or Reconfirmation of an Poll

The "REQUEST" method may be used to update or reconfirm a poll. An update to an existing poll does not involve changes to the time or candidates, and might not involve a change to the location or description for the poll. If the recipient CUA of a "REQUEST" method finds that the "UID" property value already exists on the calendar and that the "SEQUENCE" property value in the "REQUEST" is the same as the value for the existing poll, then the "REQUEST" method describes an update of the poll details, but not a rescheduling of the POLL.

The update "REQUEST" method is the appropriate response to a "REFRESH" method sent from an "Voter" to the "Organizer" of a poll.

The "Organizer" of a poll may also send unsolicited "REQUEST" methods. The unsolicited "REQUEST" methods may be used to update the details of the poll without rescheduling it, to update the "RESPONSE" parameter of "Voters", or to reconfirm the poll.

6.3.1.2.3. Delegating a POll to Another CU

Some calendar and scheduling systems allow "Voters" to delegate the vote to another "Calendar User". iTIP supports this concept using the following workflow. Any "Voter" may delegate their right to vote in a poll to another CU. The implication is that the delegate participates in lieu of the original "Voter", NOT in addition to the "Voter". The delegator MUST notify the "Organizer" of this action using the steps outlined below. Implementations may support or restrict delegation as they see fit. For instance, some implementations may restrict a delegate from delegating a "REQUEST" to another CU.

The "Delegator" of a poll forwards the existing "REQUEST" to the "Delegate". The "REQUEST" method MUST include a "Voter" property with the calendar address of the "Delegate". The "Delegator" MUST also send a "REPLY" method to the "Organizer" with the "Delegator’s" "Voter" property "DELEGATED-TO" parameter set to the calendar address of the "Delegate". Also, a new "Voter" property for the "Delegate" MUST be included and must specify the calendar user address set in the "DELEGATED-TO" parameter, as above.

In response to the request, the "Delegate" MUST send a "REPLY" method to the "Organizer", and optionally to the "Delegator". The "REPLY" method SHOULD include the "Voter" property with the "DELEGATED-FROM" parameter value of the "Delegator’s" calendar address.

The "Delegator" may continue to receive updates to the poll even though they will not be attending. This is accomplished by the "Delegator" setting their "role" attribute to "NON-PARTICIPANT" in the "REPLY" to the "Organizer".

6.3.1.2.4. Changing the Organizer

The situation may arise where the "Organizer" of a "VPOLL" is no longer able to perform the "Organizer" role and abdicates without passing on the "Organizer" role to someone else. When this occurs, the "Voters" of the "VPOLL" may use out-of-band mechanisms to communicate the situation and agree upon a new "Organizer". The new "Organizer" should then send out a new "REQUEST" with a modified version of the "VPOLL" in which the "SEQUENCE" number has been incremented and the "ORGANIZER" property has been changed to the new "Organizer".

6.3.1.2.5. Sending on Behalf of the Organizer

There are a number of scenarios that support the need for a "Calendar User" to act on behalf of the "Organizer" without explicit role changing. This might be the case if the CU designated as "Organizer" is sick or unable to perform duties associated with that function. In these cases, iTIP supports the notion of one CU acting on behalf of another. Using the "SENT-BY" parameter, a "Calendar User" could send an updated "VPOLL" "REQUEST". In the case where one CU sends on behalf of another CU, the "Voter" responses are still directed back towards the CU designated as "Organizer".

6.3.1.2.6. Forwarding to an Uninvited CU

A "Voter" invited to a "VPOLL" calendar component may send the "VPOLL" calendar component to another new CU not previously associated with the "VPOLL" calendar component. The current "Voter" participating in the "VPOLL" calendar component does this by forwarding the original "REQUEST" method to the new CU. The new CU can send a "REPLY" to the "Organizer" of the "VPOLL" calendar component. The reply contains an "Voter" property for the new CU.

The "Organizer" ultimately decides whether or not the new CU becomes part of the poll and is not obligated to do anything with a "REPLY" from a new (uninvited) CU. If the "Organizer" does not want the new CU to be part of the poll, the new "Voter" property is not added to the "VPOLL" calendar component. The "Organizer" MAY send the CU a "CANCEL" message to indicate that they will not be added to the poll. If the "Organizer" decides to add the new CU, the new "Voter" property is added to the "VPOLL" calendar component. Furthermore, the "Organizer" is free to change any "Voter" property parameter from the values supplied by the new CU to something the "Organizer" considers appropriate. The "Organizer" SHOULD send the new CU a "REQUEST" message to inform them that they have been added.

When forwarding a "REQUEST" to another CU, the forwarding "Voter" MUST NOT make changes to the original message.

6.3.1.2.7. Updating Voter Status

The "Organizer" of an poll may also request updated status from one or more "Voters". The "Organizer" sends a "REQUEST" method to the "Voter" and sets the "VOTER;RSVP=TRUE" property parameter. The "SEQUENCE" property for the poll is not changed from its previous value. A recipient will determine that the only change in the "REQUEST" is that their "RSVP" property parameter indicates a request for updated status. The recipient SHOULD respond with a "REPLY" method indicating their current vote with respect to the "REQUEST".

6.3.1.3. REPLY

The "REPLY" method in a "VPOLL" calendar component is used to respond (e.g., accept or decline) to a "REQUEST" or to reply to a delegation "REQUEST". When used to provide a delegation response, the "Delegator" SHOULD include the calendar address of the "Delegate" on the "DELEGATED-TO" property parameter of the "Delegator’s" "Voter" property. The "Delegate" SHOULD include the calendar address of the "Delegator" on the "DELEGATED-FROM" property parameter of the "Delegate’s" "Voter" property.

The "REPLY" method is also used when processing of a "REQUEST" fails. Depending on the value of the "REQUEST-STATUS" property, no action may have been performed.

The "Organizer" of a poll may receive the "REPLY" method from a CU not in the original "REQUEST". For example, a "REPLY" may be received from a "Delegate" to a poll. In addition, the "REPLY" method may be received from an unknown CU (a "Party Crasher"). This uninvited "Voter" may be accepted, or the "Organizer" may cancel the poll for the uninvited "Voter" by sending a "CANCEL" method to the uninvited "Voter".

An "Voter" MAY include a message to the "Organizer" using the "COMMENT" property. For example, if the user indicates a low interest and wants to let the "Organizer" know why, the reason can be expressed in the "COMMENT" property value.

The "Organizer" may also receive a "REPLY" from one CU on behalf of another. Like the scenario enumerated above for the "Organizer", "Voters" may have another CU respond on their behalf. This is done using the "SENT-BY" parameter.

The optional properties listed in the table below (those listed as "0+" or "0 or 1") MUST NOT be changed from those of the original request.

This method type is an iCalendar object that conforms to the following property constraints:

Constraints for a METHOD:REPLY of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST be REPLY.

VPOLL	1+	All components MUST have the same UID.
VOTER	1	MUST be the address of the Voter replying.
DTSTAMP	1	
ORGANIZER	1	
UID	1	MUST be the UID of the original REQUEST.
SEQUENCE	0 or 1	If non-zero, MUST be the sequence number of the original REQUEST. MAY be present if 0.
ACCEPT-RESPONSE	0 or 1	
ATTACH	0+	
CATEGORIES	0+	
CLASS	0 or 1	
COMMENT	0+	
COMPLETED	0 or 1	
CONTACT	0+	
CREATED	0 or 1	
DESCRIPTION	0 or 1	
DTEND	0 or 1	If present, DURATION MUST NOT be present.
DTSTART	0 or 1	
DURATION	0 or 1	If present, DTEND MUST NOT be present.
GEO	0 or 1	
LAST-MODIFIED	0 or 1	
LOCATION	0 or 1	
POLL-ITEM-ID	1+	One per item being voted on.
POLL-MODE	0	
POLL-PROPERTIES	0	
PRIORITY	0 or 1	
RELATED-TO	0+	
RESOURCES	0+	
REQUEST-STATUS	0+	
STATUS	0 or 1	
SUMMARY	0 or 1	
TRANSP	0 or 1	
URL	0 or 1	
IANA-PROPERTY	0+	
X-PROPERTY	0+	
VALARM	0	
VTIMEZONE	0 or 1	MUST be present if any date/time refers to a timezone.

IANA-COMPONENT	0+	
X-COMPONENT	0+	
VEVENT	0	
VFREEBUSY	0	
VJOURNAL	0	
VTODO	0	

6.3.1.4. CANCEL

The "CANCEL" method in a "VPOLL" calendar component is used to send a cancellation notice of an existing poll request to the affected "Voters". The message is sent by the "Organizer" of the poll.

The "Organizer" MUST send a "CANCEL" message to each "Voter" affected by the cancellation. This can be done using a single "CANCEL" message for all "Voters" or by using multiple messages with different subsets of the affected "Voters" in each.

When a "VPOLL" is cancelled, the "SEQUENCE" property value MUST be incremented as described in Section 6.2.3.

Once a CANCEL message has been sent to all voters no further voting may take place. The poll is considered closed.

This method type is an iCalendar object that conforms to the following property constraints:

Constraints for a METHOD:CANCEL of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST be CANCEL.
VPOLL	1+	All must have the same UID.
VOTER	0+	MUST include some or all Voters being removed from the poll. MUST include some or all Voters if the entire poll is cancelled.
UID	1	MUST be the UID of the original REQUEST.

DTSTAMP	1	
ORGANIZER	1	
SEQUENCE	1	
ATTACH	0+	ACCEPT-RESPONSE
0 or 1		
COMMENT	0+	
COMPLETED	0 or 1	
CATEGORIES	0+	
CLASS	0 or 1	
CONTACT	0+	
CREATED	0 or 1	
DESCRIPTION	0 or 1	
DTEND	0 or 1	If present, DURATION MUST NOT be present.
DTSTART	0 or 1	
DURATION	0 or 1	If present, DTEND MUST NOT be present.
GEO	0 or 1	
LAST-MODIFIED	0 or 1	
LOCATION	0 or 1	
POLL-ITEM-ID	0	
POLL-MODE	0	
POLL-PROPERTIES	0	
PRIORITY	0 or 1	
RELATED-TO	0+	
RESOURCES	0+	
STATUS	0 or 1	MUST be set to CANCELLED to cancel the entire event. If uninviting specific Attendees, then MUST NOT be included.
SUMMARY	0 or 1	
TRANSP	0 or 1	
URL	0 or 1	
IANA-PROPERTY	0+	
X-PROPERTY	0+	
REQUEST-STATUS	0	
VALARM	0	
VTIMEZONE	0+	MUST be present if any date/time refers to a timezone.
IANA-COMPONENT	0+	
X-COMPONENT	0+	
VTODO	0	
VJOURNAL	0	

VEVENT	0	
VFREEBUSY	0	

6.3.1.5. REFRESH

The "REFRESH" method in a "VPOLL" calendar component is used by "Voters" of an existing event to request an updated description from the poll "Organizer". The "REFRESH" method must specify the "UID" property of the poll to update. The "Organizer" responds with the latest description and version of the poll.

This method type is an iCalendar object that conforms to the following property constraints:

Constraints for a METHOD:REFRESH of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST be REFRESH.
VPOLL	1	
VOTER	1	MUST be the address of requester.
DTSTAMP	1	
ORGANIZER	1	
UID	1	MUST be the UID associated with original REQUEST.
COMMENT	0+	
COMPLETED	0	
IANA-PROPERTY	0+	
X-PROPERTY	0+	
ACCEPT-RESPONSE	0	
ATTACH	0	
CATEGORIES	0	
CLASS	0	
CONTACT	0	
CREATED	0	
DESCRIPTION	0	
DTEND	0	
DTSTART	0	
DURATION	0	
GEO	0	
LAST-MODIFIED	0	

LOCATION	0	
POLL-ITEM-ID	0	
POLL-MODE	0	
POLL-PROPERTIES	0	
PRIORITY	0	
RELATED-TO	0	
REQUEST-STATUS	0	
RESOURCES	0	
SEQUENCE	0	
STATUS	0	
SUMMARY	0	
URL	0	
VALARM	0	
VTIMEZONE	0+	
IANA-COMPONENT	0+	
X-COMPONENT	0+	
VTODO	0	
VJOURNAL	0	
VEVENT	0	
VFREEBUSY	0	

6.3.1.6. CONFIRM

The "CONFIRM" method in a "VPOLL" calendar component is used to inform recipients that the poll has completed with a result. The "Organizer" MUST be present in the confirmed poll component. "Voters" MUST NOT be present. The selected component(s) according to the poll mode MUST also be present in the poll component. Clients receiving this message may store the confirmed items in their calendars.

Once a CONFIRM message has been sent no further voting may take place. The poll is considered closed.

This method type is an iCalendar object that conforms to the following property constraints:

Constraints for a METHOD:CONFIRM of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST equal CONFIRM.
VPOLL	1+	
COMPLETED	1	
DTSTAMP	1	
DTSTART	1	
ORGANIZER	1	
SUMMARY	1	Can be null.
VOTER	0	
UID	1	
SEQUENCE	0 or 1	MUST be present if value is greater than 0; MAY be present if 0.
ACCEPT-RESPONSE	0	
ATTACH	0	
CATEGORIES	0	
CLASS	0	
COMMENT	0+	
CONTACT	0 or 1	
CREATED	0 or 1	
DESCRIPTION	0 or 1	Can be null.
DTEND	0 or 1	If present, DURATION MUST NOT be present.
DURATION	0 or 1	If present, DTEND MUST NOT be present.
LAST-MODIFIED	0 or 1	
POLL-ITEM-ID	0	
POLL-MODE	0 or 1	
POLL-PROPERTIES	0	
PRIORITY	0 or 1	
RELATED-TO	0+	
RESOURCES	0+	
STATUS	0 or 1	MAY be one of TENTATIVE/CONFIRMED/CANCELLED.
URL	0 or 1	
IANA-PROPERTY	0+	
X-PROPERTY	0+	
VOTER	0	
REQUEST-STATUS	0	
VALARM	0+	
VEVENT	0+	All selected components MAY be present.

VFREEBUSY	0	
VJOURNAL	0+	All selected components MAY be present.
VTODO	0+	All selected components MAY be present.
VTIMEZONE	0+	MUST be present if any date/time refers to a timezone.
IANA-COMPONENT	0+	
X-COMPONENT	0+	

6.3.1.7. POLLSTATUS

The "POLLSTATUS" method in a "VPOLL" calendar component is used to inform recipients of the current status of the poll in a compact manner. The "Organizer" MUST be present in the confirmed poll component. "Voters" MUST NOT be present. The selected component(s) according to the poll mode MUST also be present in the poll component. Clients receiving this message may store the confirmed items in their calendars.

This method type is an iCalendar object that conforms to the following property constraints:

Constraints for a METHOD:CONFIRM of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST equal POLLSTATUS.
VPOLL	1+	
COMPLETED	0 or 1	Only present for a completed poll
DTSTAMP	1	
DTSTART	0 or 1	
ORGANIZER	1	
SUMMARY	1	Can be null.
VOTER	1+	
UID	1	
SEQUENCE	0 or 1	MUST be present if value is greater than 0; MAY be present if 0.

ACCEPT-RESPONSE	0	
ATTACH	0	
CATEGORIES	0	
CLASS	0	
COMMENT	0+	
CONTACT	0	
CREATED	0 or 1	
DESCRIPTION	0 or 1	Can be null.
DTEND	0 or 1	If present, DURATION MUST NOT be present.
DURATION	0 or 1	If present, DTEND MUST NOT be present.
LAST-MODIFIED	0 or 1	
POLL-ITEM-ID	0	
POLL-MODE	0 or 1	
POLL-PROPERTIES	0	
PRIORITY	0 or 1	
RELATED-TO	0+	
RESOURCES	0+	
STATUS	0 or 1	MAY be one of TENTATIVE/CONFIRMED/CANCELLED.
URL	0 or 1	
IANA-PROPERTY	0+	
X-PROPERTY	0+	
VOTER	0	
REQUEST-STATUS	0	
VALARM	0+	
VEVENT	0+	All candidate components MUST be present but in a reduced form sufficient to provide the voting status.
VFREEBUSY	0	
VJOURNAL	0+	All candidate components MUST be present but in a reduced form sufficient to provide the voting status.
VTODO	0+	All candidate components MUST be present but in a reduced form sufficient to provide the voting status.
VTIMEZONE	0+	MUST be present if any date/time refers to a timezone.

IANA-COMPONENT	0+	
X-COMPONENT	0+	

7. CalDAV Extensions

This specification extends [RFC4791] in that it defines a new component and new iCalendar properties to be supported and requires extra definitions related to time-ranges and reports.

7.1. Calendar Collection Properties

This section defines new CalDAV properties for calendar collections.

7.1.1. CALDAV:vpoll-supported-component-set

Name: vpoll-supported-component-set

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

Purpose: Specifies the calendar component types (e.g., VEVENT, VTODO, etc.) that may be included in a VPOLL component.

Conformance: This property MAY be defined on any calendar collection. If defined, it MUST be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]).

Description: The CALDAV:vpoll-supported-component-set property is used to specify restrictions on the calendar component types that VPOLL components may contain in a calendar collection. Any attempt by the client to store VPOLL components with component types not listed in this property, if it exists, MUST result in an error, with the CALDAV:vpoll-supported-component precondition Section 7.2 being violated. Since this property is protected, it cannot be changed by clients using a PROPPATCH request. However, clients can initialize the value of this property when creating a new calendar collection with MKCALENDAR. In the absence of this property, the server MUST accept all component types, and the client can assume that all component types are accepted.

Definition:

```
<!ELEMENT vpoll-supported-component-set (comp+)>
```

Example:

```
<C:vpoll-supported-component-set
    xmlns:C="urn:ietf:params:xml:ns:caldav">
  <C:comp name="VEVENT"/>
  <C:comp name="VTODO"/>
</C:vpoll-supported-component-set>
```

7.1.2. CALDAV:vpoll-max-items

Name: vpoll-max-items

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

Purpose: Provides a numeric value indicating the maximum number of items that may be contained in any instance of a VPOLL calendar object resource stored in the calendar collection.

Conformance: This property MAY be defined on any calendar collection. If defined, it MUST be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]).

Description: The CALDAV:vpoll-max-items is used to specify a numeric value that indicates the maximum number of iCalendar components in any one instance of a VPOLL calendar object resource stored in a calendar collection. Any attempt to store a calendar object resource with more components per instance than this value MUST result in an error, with the CALDAV: vpoll-max-items precondition Section 7.2 being violated. In the absence of this property, the client can assume that the server can handle any number of items in a VPOLL calendar component.

Definition:

```
<!ELEMENT vpoll-max-items (#PCDATA)>
PCDATA value: a numeric value (integer greater than zero)
```

Example:

```
<C:vpoll-max-items
     xmlns:C="urn:ietf:params:xml:ns:caldav"
>25</C:vpoll-max-items>
```

7.1.3. CALDAV:vpoll-max-active

Name: vpoll-max-active

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

Purpose: Provides a numeric value indicating the maximum number of active vpolls at any one time.

Conformance: This property MAY be defined on any calendar collection. If defined, it MUST be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]).

Description: The CALDAV:vpoll-max-active is used to specify a numeric value that indicates the maximum number of active VPOLLs at any one time. Any attempt to store a new active VPOLL calendar object resource which results in exceeding this limit MUST result in an error, with the CALDAV: vpoll-max-active precondition Section 7.2 being violated. In the absence of this property, the client can assume that the server can handle any number of active VPOLLs.

Definition:

```
         <!ELEMENT vpoll-max-active (#PCDATA)>
         PCDATA value: a numeric value (integer greater than zero)
```

Example:

```
         <C:vpoll-max-active
              xmlns:C="urn:ietf:params:xml:ns:caldav"
         >25</C:vpoll-max-active>
```

7.1.4. CALDAV:vpoll-max-voters

Name: vpoll-max-voters

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

Purpose: Provides a numeric value indicating the maximum number of voters for any instance of a VPOLL calendar object resource stored in the calendar collection.

Conformance: This property MAY be defined on any calendar collection. If defined, it MUST be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]).

Description: The CALDAV:vpoll-max-voters is used to specify a numeric value that indicates the maximum number of VOTER properties for any one instance of a VPOLL calendar object resource stored in a calendar collection. Any attempt to store a calendar object resource with more VOTER properties per instance than this value MUST result in an error, with the CALDAV: vpoll-max-voters precondition Section 7.2 being violated. In the absence of this property, the client can assume that the server can handle any number of voters in a VPOLL calendar component.

Definition:

```
<!ELEMENT vpoll-max-voters (#PCDATA)>
PCDATA value: a numeric value (integer greater than zero)
```

Example:

```
<C:vpoll-max-voters
     xmlns:C="urn:ietf:params:xml:ns:caldav"
>25</C:vpoll-max-voters>
```

7.1.5. CalDAV:even-more-properties

1. List of poll modes

2. Combination of components - For a caldav server, a capability dav property similar to support component sets which specifies what sets of icalendar components are supports for voting, for example, only support VEVENTs.

3. Limits:Size of individual resources

4. Limits:Size of total vpoll

5. Limits:Max voting period

6. Limits:itip/˜itip

7. Limits:remote attendee/voter capable

8. poll options - e.g "vpoll-basic"

7.2. Additional Preconditions for PUT, COPY, and MOVE

This specification creates additional Preconditions for PUT, COPY, and MOVE methods. These preconditions apply when a PUT operation of a VPOLL calendar object resource into a calendar collection occurs, or when a COPY or MOVE operation of a calendar object resource into a calendar collection occurs, or when a COPY or MOVE operation occurs on a calendar collection.

The new preconditions are:

(CALDAV:vpoll-supported-component): The VPOLL resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST contain a type of calendar component that is supported in the targeted calendar collection;

(CALDAV:vpoll-max-items): The VPOLL resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have a number of sub-components (excluding VTIMEZONE) less than or equal to the value of the CALDAV:vpoll-max-items property value Section 7.1.2 on the calendar collection where the resource will be stored;;

(CALDAV:vpoll-max-active): The PUT request, or COPY or MOVE request, MUST not result in the number of active VPOLLsbeing greater than the value of the CALDAV:vpoll-max-active property value Section 7.1.3 on the calendar collection where the resource will be stored;;

(CALDAV:vpoll-max-voters): The VPOLL resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have a number of VOTER properties less than or equal to the value of the CALDAV: vpoll-max-voters property value Section 7.1.4 on the calendar collection where the resource will be stored;;

7.3. CalDAV:calendar-query Report

This allows the retrieval of VPOLLs and their included components. The query specification allows queries to be directed at the contained sub-components

7.3.1. Example: Partial Retrieval of VPOLL

In this example, the client requests the server to return specific components and properties of the VPOLL components that overlap the time range from December 4, 2012, at 00:00:00 A.M. UTC to December 5, 2012, at 00:00:00 A.M. UTC. In addition, the DAV:getetag property is also requested and returned as part of the response. Note that due to the CALDAV: calendar-data element restrictions, the DTSTAMP property in VPOLL components has not been returned, and the only property returned in the VCALENDAR object is VERSION.

```
>> Request <<

REPORT /cyrus/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:D="DAV:"
              xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <D:getetag/>
    <C:calendar-data>
      <C:comp name="VCALENDAR">
        <C:prop name="VERSION"/>
        <C:comp name="VPOLL">
          <C:prop name="SUMMARY"/>
          <C:prop name="UID"/>
          <C:prop name="DTSTART"/>
          <C:prop name="DTEND"/>
          <C:prop name="DURATION"/>
        </C:comp>
      </C:comp>
    </C:calendar-data>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VPOLL">
        <C:time-range start="20121204T000000Z"
                      end="20121205T000000Z"/>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>

>> Response <<

HTTP/1.1 207 Multi-Status
Date: Sat, 11 Nov 2012 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx
```

```
      <?xml version="1.0" encoding="utf-8" ?>
      <D:multistatus xmlns:D="DAV:"
                 xmlns:C="urn:ietf:params:xml:ns:caldav">
        <D:response>
          <D:href>http://cal.example.com/cyrus/work/poll2.ics</D:href>
          <D:propstat>
            <D:prop>
              <D:getetag>"fffff-abcd2"</D:getetag>
              <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      BEGIN:VEVENT
      DTSTART;TZID=US/Eastern:20121202T120000
      DURATION:PT4D
      SUMMARY:Poll #2
      UID:00959BC664CA650E933C892C@example.com
      END:VPOLL
      END:VCALENDAR
      </C:calendar-data>
            </D:prop>
            <D:status>HTTP/1.1 200 OK</D:status>
          </D:propstat>
        </D:response>
        <D:response>
          <D:href>http://cal.example.com/cyrus/work/poll3.ics</D:href>
          <D:propstat>
            <D:prop>
              <D:getetag>"fffff-abcd3"</D:getetag>
              <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VPOLL
      DTSTART;TZID=US/Eastern:20121204T100000
      DURATION:PT4D
      SUMMARY:Poll #3
      UID:DC6C50A017428C5216A2F1CD@example.com
      END:VPOLL
      END:VCALENDAR
      </C:calendar-data>
            </D:prop>
            <D:status>HTTP/1.1 200 OK</D:status>
          </D:propstat>
        </D:response>
      </D:multistatus>
```

7.3.2. Example: Retrieval of Vpolls by sub-component Time Range

In this example, the client requests the server to return the VPOLL components that have an alarm trigger scheduled in the specified time range.

```
>> Request <<

REPORT /cyrus/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop xmlns:D="DAV:">
    <D:getetag/>
    <C:calendar-data/>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VPOLL">
        <C:comp-filter name="*">
          <C:time-range start="20121206T100000Z"
                        end="20121207T100000Z"/>
        </C:comp-filter>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>

>> Response <<

HTTP/1.1 207 Multi-Status
Date: Sat, 11 Nov 2012 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:"
               xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:response>
    <D:href>http://cal.example.com/cyrus/work/poll4.ics</D:href>
    <D:propstat>
      <D:prop>
        <D:getetag>"fffff-abcd4"</D:getetag>
        <C:calendar-data>BEGIN:VCALENDAR
```

```
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VPOLL
DTSTAMP:20121205T235300Z
DTEND;TZID=US/Eastern:20121206T120000
LAST-MODIFIED:20121205T235308Z
SEQUENCE:1
SUMMARY:Poll #4
UID:E10BA47467C5C69BB74E8720@example.com
BEGIN:VEVENT
...
END:VEVENT
BEGIN:VEVENT
...
END:VEVENT
BEGIN:VEVENT
...
DTSTART;TZID=US/Eastern:20120606T150000
...
END:VEVENT
END:VPOLL
END:VCALENDAR
</C:calendar-data>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

7.4. CalDAV time ranges

Section 9.9 "CALDAV:time-range XML Element" in [RFC4791] describes how to specify time ranges to limit the set of calendar components returned by the server. This specification extends [RFC4791] to describe the meaning of time ranges for VPOLL

A VPOLL component is said to overlap a given time range if the condition for the corresponding component state specified in the table below is satisfied. The conditions depend on the presence of the DTSTART, DURATION, DTEND, COMPLETED and CREATED properties in the VPOLL component. Note that, as specified above, the DTEND value MUST be a DATE-TIME value equal to or after the DTSTART value if specified.

VPOLL has the DTSTART property?	VPOLL has the DURATION property?	VPOLL has the DTEND property?	VPOLL has the COMPLETED property?	VPOLL has the CREATED property?	Condition to evaluate
Y	Y	N	*	*	(start <= DTSTART+DURATION) AND ((end > DTSTART) OR (end >= DTSTART+DURATION))
Y	N	Y	*	*	((start < DTEND) OR (start <= DTSTART)) AND ((end > DTSTART) OR (end >= DTEND))
Y	N	N	*	*	(start <= DTSTART) AND (end > DTSTART)
N	N	Y	*	*	(start < DTEND) AND (end >= DTEND)
N	N	N	Y	Y	((start <= CREATED) OR (start <= COMPLETED)) AND ((end >= CREATED) OR (end >= COMPLETED))
N	N	N	Y	N	(start <= COMPLETED) AND (end >= COMPLETED)
N	N	N	N	Y	(end > CREATED)
N	N	N	N	N	TRUE

7.5. freebusy reports

How do outstanding VPOLLs affect freebusy requests? Still an outstanding issue. Transparency is client controlled - server/client implementations may provide defaults. We previously ruled out TRANSP in VPOLL. Following are some notes.

It seems appropriate to allow end-users to specify that they wish time to be reserved when voting in favor of a given event.

The default for VPOLL should be that all contained events are marked transparent - note the default for VEVENT is OPAQUE so should all contained VEVENTS have the TRANSP added when created?

Should we create temporary related events to block time if the user requests? Or just treat the VPOLL as a psuedo-collection and add it to the freebusy set?

8. Notes

Various collected notes to be fitted in somewhere...

1. What about POLL-WINNER?

```
pollwinner = "POLL-WINNER" pollwinnerparams ":" text CRLF
pollwinnerparams = *(";" other-param)
; Used with a STATUS:CONFIRMED VPOLL to indicate which
; components have been confirmed
```

2. Need to add example of freebusy in response

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//BedeworkCaldavTest//BedeworkCaldavTest
METHOD: REPLY
BEGIN:VPOLL
ORGANIZER:mailto:douglm@mysite.edu
VOTER:mailto:eric@example.com
UID:sched01-1234567890
DTSTAMP:20120101T010000Z
SEQUENCE:0
SUMMARY:What to do this week
BEGIN:VFREEBUSY
.......
END:VFREEBUSY
END:VPOLL
END:VCALENDAR
```

3. Define LINK types - n

4. Keep polls around after confirm or cancel. Use STATUS parameter inside VPOLL with states: NEEDS-ACTION, CANCELLED, CLOSED, CONFIRMED. Use POLL-ITEM-ID to identify the confirmed component.

5. A good client will remember the DTSTAMP of each VOTER and not allow an older DTSTAMP to override a newer DTSTAMP.

6. Fill out VOTER with the rest of the ATTENDEE properties (SCHEDULE-STATUS, SCHEDULE-AGENT)

7. Use STATUS to mark active, canceled, completed.

8. Client updates their VOTER property as in the POLLSTATUS example.

9. LINK or URL in POLL-MODE?

10. Use RELATED-TO in a icalendar component such as VEVENT or VTODO with a new reltypeparam to specify the relationship to a VPOLL. May need a reltypeparam for both an active and confirmed poll, or may only need one reltypeparam. Need to pick a new reltypeparam name.

11. Like POLL-PROPERTIES, but for a caldav server, a dav property that specify what icalender properties this server supports for voting.

12. Need to do a section on what Notifications to support.
    A. VPOLL is about to end and you haven’t voted on it yet.

13. Need to pick: SCHEDULED-AGENT=NONE or STATUS:DRAFT or a non-calendar collection for drafts. Each one has it’s pluses and minuses.

14. What to do with changes to STATUS:CONFIRMED? Allow them or not? What do to that poll had a winning event or todo.

15. Need to write down what isn’t valid in a VPOLL.
    a. Can’t change POLL-MODE

16. Can a user create a poll with scheduled events where that user’s isn’t the organizer of the poll? So is there a requierment that the account that poll is on is able to create each one of the resources in the poll? i.e. I can’t create a poll with a set of events where I am just the attendee of the polls. Are there any other restrictions for components in a VPOLL?

17. Ability to reject VPOLL.

18. Limits on number of outstanding VPOLL.

19. Add a dav property which will specify when a poll will expire and therfore be deleted from the server.

20. Guide for ATTENDEE roles chair, NON-PARTICIPANT etc

21. Shared Calendars
    Only voters (or delegates) can manipulate votes.

22. When voting on existing event - winning properties ONLY are merged in to the real event.

23. On confirm - send itip if appropriate (PUBLISH) - all non-participating - shared - feeds
    Organizer can specify where result is?
    Confirm can specify that itip is sent - ITIP / NONE - parameter ? on POLL-WINNER

9. Security Considerations

Applications using these property need to be aware of the risks entailed in using the URIs provided as values. See [RFC3986] for a discussion of the security considerations relating to URIs.

10. IANA Considerations

10.1. Parameter Registrations

This document defines the following new iCalendar property parameters to be added to the registry defined in Section 8.2.4 of [RFC5545]:

Property Parameter	Status	Reference
PUBLIC-COMMENT	Current	RFCXXXX, Section 4.1.1
RESPONSE	Current	RFCXXXX, Section 4.1.2
STAY-INFORMED	Current	RFCXXXX, Section 4.1.3

10.2. Property Registrations

This document defines the following new iCalendar properties to be added to the registry defined in Section 8.2.3 of [RFC5545]:

Property	Status	Reference
ACCEPT-RESPONSE	Current	RFCXXXX, Section 4.2.1
POLL-ITEM-ID	Current	RFCXXXX, Section 4.2.2
POLL-MODE	Current	RFCXXXX, Section 4.2.3
VOTER	Current	RFCXXXX, Section 4.2.5

10.3. POLL-MODE Registration Template

A poll mode is defined by completing the following template.

Poll mode name: The name of the poll mode.

Purpose: The purpose of the poll mode. Give a short but clear description.

Reference: A reference to the RFC in which the poll mode is defined

10.4. POLL-MODE Registrations

This document defines the following registered poll modes.

Poll mode name	Purpose	Reference
BASIC	To provide simple voting for a single outcome from a number of candidates.	Current

11. Acknowledgements

The authors would like to thank the members of the Calendaring and Scheduling Consortium Freebusy technical committee and the following individuals for contributing their ideas and support:

...

The authors would also like to thank the Calendaring and Scheduling Consortium for advice with this specification.

12. Normative References

[I-D.daboo-icalendar-extensions]
Daboo, C., "New Properties for iCalendar", draft-daboo-icalendar-extensions-05 (work in progress), June 2012.

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S., and D. Jensen, "HTTP Extensions for Distributed Authoring -- WEBDAV", RFC 2518, February 1999.

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

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4589] Schulzrinne, H. and H. Tschofenig, "Location Types Registry", RFC 4589, July 2006.

[RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault, "Calendaring Extensions to WebDAV (CalDAV)", RFC 4791, March 2007.

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

[RFC5546] Daboo, C., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", RFC 5546, December 2009.

[RFC6047] Melnikov, A., "iCalendar Message-Based Interoperability Protocol (iMIP)", RFC 6047, December 2010.

[W3C.REC-xml-20060816]
Yergeau, F., Paoli, J., Sperberg-McQueen, C., Maler, E., and T. Bray, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", World Wide Web Consortium FirstEdition REC-xml-20060816, August 2006,

<http://www.w3.org/TR/2006/REC-xml-20060816>.

Appendix A. Open issues

Issue 1: What’s the issue with that?

Appendix B. Change log

2012-11-02 MD Initial version

Authors’ Addresses

Eric York (editor)
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
USA

Email: eyork@apple.com
URI: http://www.apple.com/

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

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

Michael Douglass (editor)
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
USA

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