CalConnect TC

Calendaring and scheduling — Consensus scheduling — iCalendar VPOLL component

Committee Draft Standard

Warning for drafts

This document is not a CalConnect Standard. It is distributed for review and comment, and is subject to change without notice and may not be referred to as a Standard. Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

The Calendaring and Scheduling Consortium, Inc. 2024

CC/CD 51006:2024

Contents

Foreword

This specification introduces a new RFC5545 iCalendar component which allows for consensus scheduling, that is, voting on a number of alternative meeting or task alternatives.

The Calendaring and Scheduling Consortium ("CalConnect") is a global non-profit organization with the aim to facilitate interoperability of collaborative technologies and tools through open standards.

CalConnect works closely with international and regional partners, of which the full list is available on our website (https://www.calconnect.org/about/liaisons-and-relationships).

The procedures used to develop this document and those intended for its further maintenance are described in the CalConnect Directives.

In particular the different approval criteria needed for the different types of CalConnect documents should be noted. This document was drafted in accordance with the editorial rules of the CalConnect Directives.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CalConnect shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be provided in the Introduction.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

This document was prepared by Technical Committee *FREEBUSY*.

Introduction

The currently existing approach to agreeing on meeting times using iTip [IETF RFC 5546](#) and/or iMip [IETF RFC 6047](#) has 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 availability. 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 approaches such 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 usually 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 for this process and allow calendar clients and servers and 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.

Calendaring and scheduling — Consensus scheduling — iCalendar VPOLL component

1. Scope

This document provides a mechanism in iCalendar for consensus scheduling.

2. Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IETF RFC 2119, S. BRADNER. *Key words for use in RFCs to Indicate Requirement Levels*. 1997. RFC Publisher. https://www.rfc-editor.org/info/rfc2119.

IETF RFC 2518, Y. GOLAND, E. WHITEHEAD, A. FAIZI, S. CARTER and D. JENSEN. *HTTP Extensions for Distributed Authoring — WEBDAV*. 1999. RFC Publisher. https://www.rfc-editor.org/info/rfc2518.

IETF RFC 3986, T. BERNERS-LEE, R. FIELDING and L. MASINTER. *Uniform Resource Identifier (URI): Generic Syntax*. 2005. RFC Publisher. https://www.rfc-editor.org/info/rfc3986.

IETF RFC 4791, C. DABOO, B. DESRUISSEAUX and L. DUSSEAULT. *Calendaring Extensions to WebDAV (CalDAV)*. 2007. RFC Publisher. https://www.rfc-editor.org/info/rfc4791.

IETF RFC 5234, P. OVERELL. *Augmented BNF for Syntax Specifications: ABNF*. 2008. RFC Publisher. https://www.rfc-editor.org/info/rfc5234.

IETF RFC 5545, B. DESRUISSEAUX (ed.). *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. 2009. RFC Publisher. https://www.rfc-editor.org/info/rfc5545.

IETF RFC 5546, C. DABOO (ed.). *iCalendar Transport-Independent Interoperability Protocol (iTIP)*. 2009. RFC Publisher. https://www.rfc-editor.org/info/rfc5546.

IETF RFC 5646, A. PHILLIPS and M. DAVIS (eds.). *Tags for Identifying Languages*. 2009. RFC Publisher. https://www.rfc-editor.org/info/rfc5646.

IETF RFC 6047, A. MELNIKOV (ed.). *iCalendar Message-Based Interoperability Protocol (iMIP)*. 2010. RFC Publisher. https://www.rfc-editor.org/info/rfc6047.

IETF RFC 6638, C. DABOO and B. DESRUISSEAUX. *Scheduling Extensions to CalDAV*. 2012. RFC Publisher. https://www.rfc-editor.org/info/rfc6638.

IETF RFC 8984, N. JENKINS and R. STEPANEK. *JSCalendar: A JSON Representation of Calendar Data*. 2021. RFC Publisher. https://www.rfc-editor.org/info/rfc8984.

IETF RFC 9073, M. DOUGLASS. *Event Publishing Extensions to iCalendar*. 2021. RFC Publisher. https://www.rfc-editor.org/info/rfc9073.

IETF RFC 9253, M. DOUGLASS. *Support for iCalendar Relationships*. 2022. RFC Publisher. https://www.rfc-editor.org/info/rfc9253.

3. Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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 IETF RFC 2119.

The notation used in this memo to (re-)define iCalendar elements is the ABNF notation of [IETF RFC 5234](#) as used by [IETF RFC 5545](#). Any syntax elements shown below that are not explicitly defined in this specification come from iCalendar [IETF RFC 5545](#).

Additionally, the following terms are used:

3.1.
consensus scheduling
The process whereby users come to some agreement on meeting or task alternatives and then book that meeting or task.

3.2.
active Vpoll
A VPoll may have a DTSTART, DTEND and DURATION which may define the start and end of the active voting period

3.3.
voter
A participant who votes on the alternatives. A voter need not be an attendee of any of the alternatives presented.

4. 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 a 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 of the meeting. In addition the voters will all be potential attendees.

This specification not only defines data structures but adds new iTip methods, one used when consensus has been reached and one to distribute the current status of the poll.

This document will show how a VPOLL object is used to inform voters of the state of a simple vote on some alternatives.

4.1. 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 and a new component 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 [IETF RFC 5545](#) property.
SEQUENCE	The usual [IETF RFC 5545](#) property. See below for SEQUENCE behavior.
UID	The usual [IETF RFC 5545](#) property.
SUMMARY	The usual [IETF RFC 5545](#) property. This optional but recommended property provides the a short title to the poll.
DESCRIPTION	The usual [IETF RFC 5545](#) property. This optional property provides more details.
DTEND	The usual [IETF RFC 5545](#) 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-COMPLETION   A new property which defines who (server or client) chooses and/or submits the winning choice. In our example the value is "SERVER-SUBMIT" which means the client chooses the winner but the server will submit the winning choice.

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.

PARTICIPANT   There is one of these components for each voter with the PARTICIPANT-TYPE set to "VOTER". The CALENDAR-ADDRESS property identifies the voter and this component will contain one VOTE component for each item being voted on.

VOTE   A new component. There is one of these for each voter and choice. It usually contains at least a POLL-ITEM-ID property to identify the choice and a RESPONSE property to provide a vote. For more complex poll modes it may contain other information such as cost or estimated duration.

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

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-COMPLETION:SERVER-SUBMIT
POLL-PROPERTIES:DTSTART,LOCATION
UID:sched01-1234567890
DTSTAMP:20120101T000000Z
SUMMARY:What to do this week
DTEND:20120101T000000Z
BEGIN:PARTICIPANT
PARTICIPANT-TYPE: VOTER
CALENDAR-ADDRESS:mailto:cyrus@example.com
UID:schedpart-7890123456
END:PARTICIPANT
BEGIN:PARTICIPANT
PARTICIPANT-TYPE: VOTER
CALENDAR-ADDRESS:mailto:eric@example.com
UID:schedpart-0987654321
END:PARTICIPANT
BEGIN:PARTICIPANT
PARTICIPANT-TYPE: VOTER,OWNER
CALENDAR-ADDRESS:mailto:mike@example.com
UID:schedpart-1234567890
```

```
      END:PARTICIPANT
      BEGIN:VEVENT.......(with a poll-item-id=1)
      END:VEVENT
      BEGIN:VEVENT.......(with a poll-item-id=2)
      END:VEVENT
      BEGIN:VEVENT.......(with a poll-item-id=3)
      END:VEVENT
      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 iMip or through some VPOLL enabled service.

4.2. The VPOLL Alternative Choices: An Overview

Within the VPOLL component we have the alternatives to vote on. In many respects these are standard [IETF RFC 5545](#) components. For our simple use case they are all VEVENT components. In addition to the usual [IETF RFC 5545](#) 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.

4.3. 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 and components:

DTSTAMP   The usual [IETF RFC 5545](#) property.

SEQUENCE   The usual [IETF RFC 5545](#) property. See below for SEQUENCE behavior.

UID   Same as the request.

SUMMARY   Same as the request.

PARTICIPANT   One only with a CALENDAR-ADDRESS identifying the voter replying.

VOTE   One per item being voted on.

POLL-ITEM-ID   One inside each VOTE component to identify the choice.

RESPONSE   One inside each VOTE component to specify the vote.

Note that a voter can send a number of REPLYs for each REQUEST sent by the organizer. in BASIC mode each REPLY completely replaces the voting record for that voter for all components being voted on. In our example, if Eric responds and votes 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.

NOTE   This is poll-mode specific behavior.

EXAMPLE

REPLY VPOLL from Cyrus:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example//Example
METHOD: REPLY
```

```
BEGIN:VPOLL
UID:sched01-1234567890
DTSTAMP:20120101T010000Z
SUMMARY:What to do this week
BEGIN:PARTICIPANT
PARTICIPANT-TYPE: VOTER
CALENDAR-ADDRESS:mailto:cyrus@example.com
UID:schedpart-7890123456
BEGIN:VOTE
POLL-ITEM-ID:1
RESPONSE:50
COMMENT:Work on iTIP
END:VOTE
BEGIN:VOTE
POLL-ITEM-ID:2
RESPONSE:100
COMMENT:Work on WebDAV
END:VOTE
BEGIN:VOTE
POLL-ITEM-ID:3
RESPONSE:0
END:VOTE
END:PARTICIPANT
END:VPOLL
END:VCALENDAR
```

4.4. VPOLL updates

When the owner 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, and one or more PARTICIPANT components each populated with zero or more VOTE components.

EXAMPLE

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example//Example
METHOD: POLLSTATUS
BEGIN:VPOLL
UID:sched01-1234567890
DTSTAMP:20120101T020000Z
SEQUENCE:0
SUMMARY:What to do this week
BEGIN:PARTICIPANT
PARTICIPANT-TYPE: VOTER
CALENDAR-ADDRESS:mailto:cyrus@example.com
UID:schedpart-7890123456
BEGIN: VOTE
POLL-ITEM-ID:1
RESPONSE:50
COMMENT:Work on iTIP
END:VOTE
BEGIN:VOTE
POLL-ITEM-ID:2
RESPONSE:100
COMMENT:Work on WebDAV
END:VOTE
BEGIN:VOTE
POLL-ITEM-ID:3
```

```
      RESPONSE:0
      END:VOTE
      END:PARTICIPANT
      BEGIN:PARTICIPANT
      PARTICIPANT-TYPE: VOTER
      CALENDAR-ADDRESS:mailto:eric@example.com
      UID:schedpart-0987654321
      BEGIN:VOTE
      POLL-ITEM-ID:1
      RESPONSE:100
      END:VOTE
      BEGIN:VOTE
      POLL-ITEM-ID:2
      RESPONSE:100
      END:VOTE
      BEGIN:VOTE
      POLL-ITEM-ID:3
      RESPONSE:0
      END:VOTE
      END:PARTICIPANT
      BEGIN:PARTICIPANT
      PARTICIPANT-TYPE: VOTER,OWNER
      CALENDAR-ADDRESS:mailto:mike@example.com
      UID:schedpart-1234567890
      BEGIN: VOTE
      POLL-ITEM-ID:1
      RESPONSE:50
      COMMENT:Work on iTIP
      END:VOTE
      BEGIN:VOTE
      POLL-ITEM-ID:2
      RESPONSE:100
      COMMENT:Work on WebDAV
      END:VOTE
      BEGIN:VOTE
      POLL-ITEM-ID:3
      RESPONSE:0
      END:VOTE
      END:PARTICIPANT
      END:VPOLL
      END:VCALENDAR
```

4.5. 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. A VPOLL can be completed (and effectively closed for voting) by sending an iTip REQUEST message with the VPOLL STATUS property set to COMPLETED.

The poll winner is confirmed by sending a final iTip REQUEST message with the VPOLL STATUS property set to CONFIRMED. In this case the VPOLL component contains all the events being voted on along with a POLL-WINNER property to identify the winning event. As the POLL- COMPLETION property is set to SERVER-SUBMIT the server will submit the winning choice and when it has done so set the STATUS to "SUBMITTED".

EXAMPLE

VPOLL confirmation:

```
BEGIN:VCALENDAR
```

```
VERSION:2.0
PRODID:-//Example//Example
METHOD: REQUEST
BEGIN:VPOLL
UID:sched01-1234567890
DTSTAMP:20120101T030000Z
COMPLETED:20120101T030000Z
POLL-COMPLETION:SERVER-SUBMIT
SEQUENCE:0
SUMMARY:What to do this week
STATUS:CONFIRMED
POLL-WINNER:3
BEGIN:PARTICIPANT
PARTICIPANT-TYPE: OWNER
CALENDAR-ADDRESS:mailto:mike@example.com
UID:schedpart-1234567890
END:PARTICIPANT
BEGIN:VEVENT.......(with a poll-item-id=1)
END:VEVENT
BEGIN:VEVENT.......(with a poll-item-id=2)
END:VEVENT
BEGIN:VEVENT.......(with a poll-item-id=3)
END:VEVENT
END:VPOLL
END:VCALENDAR
```

4.6. Other Responses

A voter being asked to choose between a number of ORGANIZER supplied alternatives may find none of them acceptable or may simply not care.

An alternative response, which may be disallowed by the ORGANIZER, is to send back the respondees availability or freebusy or even one or more new, alternative choices.

This is accomplished by responding with a VOTE component which has no POLL-ITEM-ID property. In this case it MUST contain some alternative information. What form this takes depends on the poll mode in effect.

5. iCalendar Extensions

5.1. Updated Participant Type Value

Participant type property values are defined in section 11.2.1. of IETF RFC 9073. This specification updates that type to include the new participant type VOTER to provide information about the voter and to contain their votes.

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

```
partvalue       /= "VOTER"
```

Figure 1

Description The new property value indicates that the associated PARTICIPANT
            component identifies a voter in a VPOLL.

5.2. Updated Relation Type Value

Relationship parameter type values are defined in section 3.2.15. of [IETF RFC 5545](#). This specification updates that type to include the new relationship value POLL to provide a link to the VPOLL component in which the current component appears.

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

```
reltypeparam       /= "RELTYPE" "=" "POLL"
; Property value is a VPOLL uid
```

Figure 2

Description This parameter can be specified on a property that references another related calendar component. The new parameter value indicates that the associated property references a VPOLL component which contains the current component.

5.3. Updated Status Value

Status property values are defined in section 3.8.1.11. of [IETF RFC 5545](#). This specification updates that type to define valid VPOLL status values.

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

```
statvalue /= statvalue-poll
   ; Status values for "VPOLL".
statvalue-poll = "IN-PROCESS"
        / "COMPLETED"  ; Poll has closed,
                       ; nothing has been chosen yet
        / "CONFIRMED"  ; Poll has closed and
                       ; winning items confirmed
        / "SUBMITTED"  ; The winning item has been
                       ; submitted
        / "CANCELLED"
```

Figure 3

Description These values allow clients and servers to handle the choosing and submission of winning choices.

```
If the client is choosing and the server submitting then
the client should set the POLL-WINNER property, set the
status to CONFIRMED and save the poll. When the server
submits the winning choice it will set the status to
SUBMITTED.
```

Figure 4

5.4. New Property Parameters

5.4.1. Required

Parameter name REQUIRED

Purpose   To specify whether the associated property is required in the current
                     context.

Format   This parameter is defined by the following notation:
Definition
```
requirededparam = "REQUIRED"  "=" ("TRUE" / "FALSE")
 ; Default is FALSE
```

Figure 5

Description This parameter MAY be specified on REPLY-URL and, if the value is TRUE,
             indicates the organizer requires all replies to be made via the specified
             service rather than iTip replies.

5.4.2. Stay-Informed

Parameter   STAY-INFORMED
name
Purpose   To specify the voter also wants to be added as an ATTENDEE when the
                     poll is confirmed.

Format   This parameter is defined by the following notation:
Definition
```
stayinformedparam = "STAY-INFORMED"  "=" ("TRUE" / "FALSE")
                  ; Default is FALSE
```

Figure 6

Description This parameter MAY be specified on the CALENDAR-ADDRESS property in
             the PARTICIPANT component and, if the value is TRUE, indicates the voter
             wishes to be added to the final choice as a non-participant.

5.5. New Properties

5.5.1. 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   Non-standard or iana parameters can be specified on this property.
Parameters
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 their freebusy or availability rather than choosing
                     one of the presented alternatives.

                     If this property is not present voters are only allowed to respond to the
                     choices in the request.

Format   This property is defined by the following notation:
Definition
```
acceptresponse = "ACCEPT-RESPONSE" acceptresponseparams ":"
            iana-token ("," iana-token) CRLF
```

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

Figure 7

5.5.2. Poll-Completion

Property name	POLL-COMPLETION
Purpose	This property is used in VPOLL to indicate whether the client or server is responsible for choosing and/or submitting the winner(s).
Description	When a VPOLL is stored on a server which is capable of handling choosing and submission of winning choices a value of SERVER indicates that the server should close the poll, choose the winner and submit whenever it is appropriate to do so.
	For example, in BASIC poll-mode, reaching the DTEND of the poll could trigger this server side action.
	Server initiated submission requires that the submitted choice MUST be a valid calendaring component.
	POLL-COMPLETION=SERVER-SUBMIT allows the client to set the poll-winner, set the status to CONFIRMED and then store the poll on the server. The server will then submit the winning choice and set the status to SUBMITTED.
Format Definition	This property is defined by the following notation:

```
poll-completion = "POLL-COMPLETION" pcparam ":" pcvalue CRLF

pcparam = *(";" other-param)

pcvalue = "SERVER"  ; The server is responsible for both choosing and
               ; submitting the winner(s)
     / "SERVER-SUBMIT" ; The server is responsible for
               ; submitting the winner(s). The client chooses.
     / "SERVER-CHOICE"  ; The server is responsible for
               ; choosing the winner(s). The client will submit.
     / "CLIENT" ; The client is responsible for both choosing and
               ; submitting the winner(s)
     / iana-token
     / x-name
     ;Default is CLIENT
```

Figure 8

Example	The following is an example of this property:

```
POLL-COMPLETION: SERVER-SUBMIT
```

Figure 9

5.5.3. Poll-Item-Id

Property name	POLL-ITEM-ID
Purpose	This property is used in VPOLL child components as an identifier.

Value type	INTEGER
Property Parameters	Non-standard parameters can be specified on this property.
Conformance	This property MUST be specified in a VOTE component and in VPOLL choice items.
Description	In a METHOD:REQUEST each choice 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 small integer 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.
Format Definition	This property is defined by the following notation:

```
pollitemid = "POLL-ITEM-ID" pollitemdparams ":"
                integer CRLF

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

Figure 10

5.5.4. 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)
```

Figure 11

5.5.5. 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)
```

Figure 12

5.5.6. Poll-Winner

Property name	POLL-WINNER
Purpose	This property is used in a basic mode VPOLL to indicate which of the VPOLL sub-components won.
Value type	INTEGER
Property Parameters	Non-standard parameters can be specified on this property.
Conformance	This property MAY be specified in a VPOLL component.
Description	For poll confirmation each child component MUST have a POLL-ITEM-ID property. For basic mode the VPOLL component SHOULD have a POLL-WINNER property which MUST correspond to one of the POLL-ITEM-ID properties and indicates which sub-component was the winner.
Format Definition	This property is defined by the following notation:

```
pollwinner = "POLL-WINNER" pollwinnerparams ":"
                integer CRLF

pollwinnerparams = *(";" other-param)

    ; Used with a STATUS:CONFIRMED VPOLL to indicate which
    ; components have been confirmed
```

Figure 13

5.5.7. Reply-URL

Property name	REPLY-URL
Purpose	This property may be used in scheduling messages to indicate additional reply methods, for example a web-service.
Value type	URI

Property Parameters	Non-standard, required or iana parameters can be specified on this property.
Conformance	This property MAY be specified in a VPOLL component.
Description	When used in a scheduling message this property indicates additional or required services that can be used to reply. Typically this would be a web service of some form.
Format Definition	This property is defined by the following notation:

```
reply-url = "REPLY-URL" reply-urlparams ":" uri CRLF

reply-urlparams = *(
                ("\;" requiredparam) /
                ("\;" other-param)
                )
```

Figure 14

5.5.8. Response

Property name	RESPONSE
Purpose	To specify a response vote.
Value type	INTEGER
Format Definition	This property is defined by the following notation:

```
response = "RESPONSE" response-params ":" integer CRLF
                ; integer value 0..100

responseparams = *("\;" other-param)
```

Figure 15

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

5.6. New Components

5.6.1. 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
           *participantc *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 /
           ;
           ; 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 / rstatus / related /
           resources / x-prop / iana-prop
           ;
           ; The following is OPTIONAL, it SHOULD appear
           ; once for the confirmation of a BASIC mode
           ; VPOLL. Other modes may define differing
           ; requirements.
           ;
           pollwinner /
           ;
           )
```

Figure 16

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

The PARTICIPANT components in VPOLL requests provide information on each recipient who will be voting — both their identity through the CALENDAR-ADDRESS property and their votes through the VOTE components.

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.

5.6.2. VOTE Component

Component name	VOTE
Purpose	This component provides a mechanism by which voters can vote on provided choices.
Conformance	This component may be specified zero or more times in a PARTICIPANT component which identifies the voter.
Format Definition	This property is defined by the following notation:

```
votec     = "BEGIN" ":" "VOTE" CRLF
            voteprop
            *eventc *todoc *journalc *freebusyc
            *availabilityc *alarmc *iana-comp *x-comp
            "END" ":" "VOTE" CRLF

voteprop = *(
            ;
            ; The following are REQUIRED,
            ; but MUST NOT occur more than once.
            ;
            pollitemid / response /
            ;
            ; The following are OPTIONAL,
            ; and MAY occur more than once.
            ;
            comment / x-prop / iana-prop
            ;
            )
```

Figure 17

Description	This component appears inside the PARTICIPANT component with a PARTICIPANT-TYPE of VOTER to identify the voter. This component contains that participants responses.

The required and optional properties and their meanings will depend upon the POLL-MODE in effect.

For any POLL-MODE, POLL-ITEM-ID is used to associate the information to a choice supplied by the organizer. This means that each VOTE component only provides information about that choice.

If allowed by the POLL-MODE a VOTE component without a POLL-ITEM- ID may be provided in a REPLY to indicate a possible new choice or to provide information to the ORGANIZER — such as the respondees availability.

6. 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.

New poll modes can be registered by including a completed POLL-MODE Registration Template (see Clause 12.3) in a published RFC.

6.1. 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.

6.1.1. Property restrictions

This poll mode has the following property requirements:

POLL-ITEM-ID	Each contained sub-component that is being voted upon MUST contain a POLL-ITEM_ID property which is unique within the context of the POLL. The value MUST NOT be reused when events are removed and/or added to the poll.
POLL-WINNER	On confirmation of the poll this property MUST be present and identifies the winning component.

6.1.2. Outcome reporting

To confirm the winner the POLL-WINNER property MUST be present and the STATUS MUST be set to CONFIRMED.

When the winning VEVENT or VTODO is not a scheduled entity, that is, it has no ORGANIZER or ATTENDEES it MUST be assigned an ORGANIZER property and a list of non-participating ATTENDEEs. This allows the winning entity to be distributed to the participants through iTip or some other protocol.

7. New Participant Properties for iTip

The following properties are defined to be used within PARTICIPANT during scheduling and take the place of ATTENDEE and ORGANIZER properties and parameters. These are not solely for VPOLL but may be used in any future component.

7.1. Kind

Property name	KIND
Purpose	This is what kind of entity this participant is, if known.
Property Parameters	Non-standard or iana parameters can be specified on this property.

Conformance	This property MAY be specified in a PARTICIPANT component.
Description	When used in a PARTICIPANT component this property indicates the kind of entity, individual, group etc.
	It takes the va;ues hown below which are a redefinition of the CUTYPE parameter values defined in [IETF RFC 5545](#) and aligned with [IETF RFC 8984](#) .
Format Definition	This property is defined by the following notation:

```
kind = "KIND" kindparams ":"
             "INDIVIDUAL"   ; An individual
           / "GROUP"        ; A group of individuals
           / "RESOURCE"     ; A physical resource
           / "LOCATION"     ; A location resource e.g a room
known
             iana-token ("," iana-token) CRLF

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

Figure 18

7.2. Redefined participation type

This specification redefines the PARTICIPATION-TYPE property allowing it to take multiple values and extending those values to align with [IETF RFC 8984](#) roles and add a new "VOTER" role. There are also changes to the description to clarify it's use defining the roles that participant takes.

Property name	PARTICIPANT-TYPE
Purpose	This property is equivalent to the [IETF RFC 5545](#) ATTENDEE ROLE parameter but includes more values to align with [IETF RFC 8984](#).
Value Type	The value type for this property is TEXT. The allowable values are defined below.
Property Parameters	Non-standard or iana parameters can be specified on this property.
Conformance	This property MUST be specified once within a "PARTICIPANT" component.
Description	This property defines the type of participation, that is the roles the participant takes.
	It includes the values defined in [IETF RFC 8984](#).
	Note that the kind of participant, for example individual or group, is defined in the KIND property specified here.
	Some of the roles are required for the participant to be a schedulable object. These are the roles that are shown below. * Do we need a separate registry or should we extend that one?
Format Definition	This property is defined by the following notation:

```
participanttype  = "PARTICIPANT-TYPE" partvalueparam ":"
                    partvalue
                    *("," partvalue) CRLF
```

```
partvalue      = ("OWNER"
                 / "ATTENDEE"
                 / "OPTIONAL"
                 / "INFORMATIONAL"
                 / "CHAIR"
                 / "VOTER"
                 / "ACTIVE"
                 / "INACTIVE"
                 / "SPONSOR"
                 / "CONTACT"
                 / "BOOKING-CONTACT"
                 / "EMERGENCY-CONTACT"
                 / "PUBLICITY-CONTACT"
                 / "PLANNER-CONTACT"
                 / "PERFORMER"
                 / "SPEAKER"
                 / iana-token)    ; Other IANA-registered
                                  ; values

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

Figure 19

The values other than VOTER have the same meaning as defined in [IETF RFC 8984](#) and [IETF RFC 9073](#).

The VOTER roles value is used for a PARTICIPANT within a VPOLL to indicate a voting participant.

Other roles defined here have no direct equivalent in [IETF RFC 5545](#)

To map [IETF RFC 5545](#) ATTENDEE ROLE values to PARTICIPANT-TYPE or jsCalendar values use the following.

Table 1

RFC5545 ROLE	PARTICIPANT-TYPE	jsCalendar
CHAIR	CHAIR	chair
REQ-PARTICIPANT	ATTENDEE	attendee
OPT-PARTICIPANT	OPTIONAL	optional
NON-PARTICIPANT	INFORMATIONAL	information

The following table shows those roles that MUST appear in the PARTICIPANT-TYPE for group-scheduling. Additionally, the mapping PARTICIPANT-TYPE or jscalendar roles to [IETF RFC 5545](#) ATTENDEE and ORGANIZER values are shown.

Table 2

PARTICIPANT-TYPE	jsCalendar	RFC5545 ROLE
OWNER	owner	Create ORGANIZER
ATTENDEE	attendee	REQ-PARTICIPANT
OPTIONAL	optional	OPT-PARTICIPANT
INFORMATIONAL	informational	NON-PARTICIPANT
CHAIR	chair	CHAIR

PARTICIPANT-TYPE   **jsCalendar**   **RFC5545 ROLE**

Subsequent values have no [IETF RFC 5545](#) equivalent

CONTACT   contact

VOTER

ACTIVE

INACTIVE

SPONSOR

BOOKING-CONTACT

EMERGENCY-CONTACT

PUBLICITY-CONTACT

PLANNER-CONTACT

PERFORMER

SPEAKER

Examples — PARTICIPANT-TYPE=OWNER,... equivalent to an ORGANIZER
             — PARTICIPANT-TYPE=ATTENDEE,... equivalent to the ATTENDEE property

7.3. Participation-status

Property name   PARTICIPATION-STATUS

Purpose   This property is used in the PARTICIPANT component to indicate the participation status — if any.

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

Conformance   This property MAY be specified in a PARTICIPANT component.

Description   When used in a PARTICIPANT component this property indicates what status, if any, the participant has.

                       It takes the same values as the PARTSTAT parameter defined in [IETF RFC 5545](#).

Format Definition   This property is defined by the following notation:

```
participation-status = "PARTICIPATION-STATUS"
                  participation-statusparams ":"
          NEEDS-ACTION / ; No status
               ; has yet been set by the participant.

          ACCEPTED / ; The invited
               ; participant will participate.
          DECLINED / ; The invited
               ; participant will not participate.
           TENTATIVE / ; The invited participant
                ; may participate.
             DELEGATED / ; The invited participant
```



```
                         ; has delegated their attendance to
                         ; another participant, as specified
                          ; in the PARTICIPATION-DELEGATED-TO property.
                    iana-token ("," iana-token) CRLF

participation-statusparams = *(";" other-param)
```

Figure 20

7.4. Participation delegated from

Property name   PARTICIPATION-DELEGATED-FROM

Purpose   This property is used in the PARTICIPANT component to indicate who
                 has delegated their participation to this participant.

Property   Non-standard or iana parameters can be specified on this property.
Parameters
Conformance   This property MAY be specified in a PARTICIPANT component.

Description   This property specifies those calendar users that have delegated their
                 participation in a group-scheduled component to the calendar user
                 specified by the component.

Format   This property is defined by the following notation:
Definition
```
participation-delfrom = "PARTICIPATION-DELEGATED-FROM"
                    participation-delfromparams ":"
            CAL-ADDRESS
            *("," CAL-ADDRESS)
            iana-token ("," iana-token) CRLF

participation-delfromparams = *(";" other-param)
```

Figure 21

7.5. Participation delegated to

Property name   PARTICIPATION-DELEGATED-TO

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

Property   Non-standard or iana parameters can be specified on this property.
Parameters
Conformance   This property MAY be specified in a PARTICIPANT component.

Description   This property specifies those calendar users that have been delegated
                 participation in a group-scheduled component by the calendar user
                 specified by the component.

Format   This property is defined by the following notation:
Definition
```
participation-delto = "PARTICIPATION-DELEGATED-TO"
                    participation-deltoparams ":"
            CAL-ADDRESS
            *("," CAL-ADDRESS)
            iana-token ("," iana-token) CRLF
```

```
participation-deltoparams = *(";" other-param)
```
Figure 22

7.6. Member of

Property name   MEMBER-OF

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

Property   Non-standard or iana parameters can be specified on this property.
Parameters
Conformance   This property MAY be specified in a PARTICIPANT component.

Description   This property identifies the groups or list membership for the calendar user specified by the component. The value is one or more calendar addresses.

Format   This property is defined by the following notation:
Definition

```
member-of = "MEMBER-OF" member-ogparams ":"
            CAL-ADDRESS
            *("," CAL-ADDRESS)
            iana-token ("," iana-token) CRLF

memberofparams = *(";" other-param)
```
Figure 23

7.7. Lang

Property name   LANG

Purpose   This is the language tag, as defined in [IETF RFC 5646](#), that best describes the participant's preferred language, if known.

Property   Non-standard or iana parameters can be specified on this property.
Parameters
Conformance   This property MAY be specified in any appropriate component.

Format   This property is defined by the following notation:
Definition

```
lang = "LANG" langparams ":" TEXT CRLF

langparams = *(";" other-param)
```
Figure 24

7.8. Expect reply

Property name   EXPECT-REPLY

Purpose   If true, the organizer is expecting the participant to notify them of their participation status.

Property   Non-standard or iana parameters can be specified on this property.
Parameters
Conformance   This property MAY be specified once in the PARTICIPANT component.

Format Definition   This property is defined by the following notation:

```
expect-reply = "EXPECT-REPLY"
               expect-replyparams ":"
               ( "TRUE" / "FALSE") CRLF

expect-replyparams = *(";" other-param)
```

Figure 25

8. iTip With Participants

The PARTICIPANT component introduced in IETF RFC 9073 with the addition of some properties defined in this specification mirrors the participant object in IETF RFC 8984.

For all existing schedulable and publishable components, VEVENT, VTODO, VFREEBUSY, and VAVAILABILITY; the ORGANIZER and ATTENDEE properties MUST be supplied as appropriate. For new components such as VPOLL, defined here, only PARTICIPANT components MUST be used.

Note that extensions to the IETF RFC 5546 specification for VPOLL will be dealt with in later sections.

A participant object that takes part in group scheduling MUST have the following characteristics:

— It MUST have a calendar address (IETF RFC 5545 CALENDAR-ADDRESS, IETF RFC 8984 calendarAddress).
— It must have one or more scheduling role defined. (PARTICIPANT-TYPE or IETF RFC 8984 role)

Scheduling with PARTICIPANT components behaves exactly as with ATTENDEE and ORGANIZER properties. When iTip specifies the setting of ATTENDEE or ORGANIZER parameters then the appropriate PARTICIPANT property will be set.

8.1. Attendee parameters mapping

Table 3

Parameter	iCalendar PARTICIPANT	jscalendar participant
CN	NAME (defined 7986)	name
CUTYPE	KIND (defined here)	kind
DELEGATED-FROM	PARTICIPATION-DELEGATED-FROM Clause 7.4	delegatedFrom
DELEGATED-TO	PARTICIPATION-DELEGATED-TO (Defined here)	delegatedTo
DIR	LINK IETF RFC 9253	links IETF RFC 8984
LANGUAGE	LANG (defined here)	language
MEMBER	MEMBER-OF (defined here)	memberOf
PARTSTAT	PARTICIPATION-STATUS (defined here)	MEMBER
ROLE	PARTICIPATION-TYPE (Updated here)	roles
RSVP	EXPECT-REPLY (Defined here)	expectReply
SENT-BY		

9. iTIP Extensions

This specification introduces a number of extensions to [IETF RFC 5546](#). In group scheduling the parties involved are organizer and attendees. In VPOLL the parties are owner and voter participants.

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

9.1. Methods

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

Table 4

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 owner returns a METHOD:REQUEST with the current full state, or a METHOD: CANCEL 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 and PARTICIPANTS. One PARTICIPANT MUST be the owner.

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

Table 5

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

9.2. Interoperability Models

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

9.2.1. Delegation

TBD

9.2.2. Acting on Behalf of Other Calendar Users

TBD

9.2.3. 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

9.2.4. Message Sequencing

TBD

9.3. Application Protocol Elements

9.3.1. 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.

Table 6

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.

Table 7

Method	Description
PUBLISH	Post notification of an poll. Used primarily as a method of advertising the existence of a poll.
REQUEST	To make a request for a poll. This is an explicit invitation to one or more voters. Poll requests are also used to update, change or confirm an existing poll. Clients that cannot handle REQUEST MAY degrade the poll to view it as a PUBLISH. REQUEST SHOULD NOT be used just to set the status of the poll - POLLSTATUS provides a more compact approach.

Method	Description
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 a poll owner 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, and PARTICIPANT.

9.3.2. Method: 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. An owner participant 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 "Owner" may subsequently update (with another "PUBLISH" method) or cancel (with a "CANCEL" method) a previously published "VPOLL" calendar component.

> Note Not clear how useful this is but needs some work on transmitting the current vote without any voter identification.

This method type has a "METHOD" property with the value "PUBLISH" and one or more complete VPOLL objects that conform to the property constraints defined in section Clause 5.6.1.

9.3.3. Method: 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.
— Complete or confirm the poll.
— 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 "Owner" to another CU.

The "Owner" originates the "REQUEST". The recipients of the "REQUEST" method are the CUs voting in the poll, the "Voters". "Voters" use the "REPLY" method to convey votes to the "Owner".

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 has a "METHOD" property with the value "REQUEST" and a single complete VPOLL object that conforms to the property constraints defined in section Clause 5.6.1.

9.3.3.1. 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.

9.3.3.2. Updating or Reconfirmation of a 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 a "Voter" to the "Owner" of a poll.

The "Owner" 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.

9.3.3.3. Confirmation of a Poll

The "REQUEST" method may be used to confirm a poll, that is announce the winner in BASIC mode. The STATUS MUST be set to CONFIRMED and for BASIC mode a VPOLL POLL-WINNER property must be provided with the poll-id of the winning component.

9.3.3.4. Closing a Poll

The "REQUEST" method may be used to close a poll, that is indicate voting is completed. The STATUS MUST be set to COMPLETED.

9.3.3.5. 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 "Owner" 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 "Owner" 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 "Owner", and optionally to the "Delegator". The "REPLY"

method SHOULD include the "Voter" participant with the "PARTICIPANT-DELEGATED-FROM" property 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 "INFORMATIONAL" in the "REPLY" to the "Owner".

9.3.3.6. Changing the Owner

The situation may arise where the "Owner" of a "VPOLL" is no longer able to perform the "Owner" role and abdicates without passing on the "Owner" 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 "Owner". The new "Owner" should then send out a new "REQUEST" with a modified version of the "VPOLL" in which the "SEQUENCE" number has been incremented and the owner role assigned to the appropriate "PARTICIPANT".

9.3.3.7. Sending on Behalf of the Owner

There are a number of scenarios that support the need for a "Calendar User" to act on behalf of the "Owner" without explicit role changing. This might be the case if the CU designated as "Owner" 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. In the case where one CU sends on behalf of another CU, the "Voter" responses are still directed back towards the CU designated as "Owner".

9.3.3.8. 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 "Owner" of the "VPOLL" calendar component. The reply contains a "Voter" participant component for the new CU.

The "Owner" ultimately decides whether 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 "Owner" does not want the new CU to be part of the poll, the new "Voter" is not added to the "VPOLL" calendar component. The "Owner" MAY send the CU a "CANCEL" message to indicate that they will not be added to the poll.

If the "Owner" decides to add the new CU, a new participant for the "Voter" is added to the "VPOLL" calendar component. Furthermore, the "Owner" is free to change any "Voter" participant property values from the values supplied by the new CU to something the "Owner" considers appropriate. The "Owner" 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.

9.3.3.9. Updating Voter Status

The "Owner" of a poll may also request updated status from one or more "Voters". The "Owner" sends a "REQUEST" method to the "Voter" and sets the "EXPECT-REPLY" property value to TRUE. 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 "EXPECT-REPLY" property indicates a request for updated status. The recipient SHOULD respond with a "REPLY" method indicating their current vote with respect to the "REQUEST".

9.3.4. Method: 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" "CALENDAR-ADDRESS" property. The "Delegate" SHOULD include the calendar address of the "Delegator" on the "DELEGATED-FROM" property parameter of the "Delegate's" "CALENDAR-ADDRESS" 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 "Owner" 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 "Owner" may cancel the poll for the uninvited "Voter" by sending a "CANCEL" method to the uninvited "Voter".

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

The "Owner" may also receive a "REPLY" from one CU on behalf of another. Like the scenario enumerated above for the "Owner", "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. (But see comments on VFREEBUSY and VAVAILABILITY)

This method type has a "METHOD" property with the value "REPLY" and a single VPOLL object. That object MUST contain the properties shown below. All other properties or components SHOULD NOT be present and MUST be ignored by the recipient if present.

Table 8 — 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.
PARTICIPANT	1	Identifies the Voter replying.
DTSTAMP	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	
POLL-ITEM-ID	1+	One per item being voted on.
VFREEBUSY	0 or 1	A voter may respond with a VFREEBUSY component indicating that the "Owner" may select some other time which is not marked as busy.
VAVAILABILITY	0	A voter may respond with a VAVAILABILITY component indicating that the "Owner" may select some other time which is shown as available.

9.3.5. Method: 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 "Owner" of the poll.

The "Owner" 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 Clause 9.2.3.

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

This method type has a "METHOD" property with the value "CANCEL" and one or more VPOLL objects. Those objects MUST contain the properties shown below. All other properties or components SHOULD NOT be present and MUST be ignored by the recipient if present.

Table 9 — 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.
PARTICIPANT	0+	Any included participents are being removed from the poll. Otherwise the entire poll is cancelled.
UID	1	MUST be the UID of the original REQUEST.
DTSTAMP	1	
SEQUENCE	1	

9.3.6. Method: REFRESH

The "REFRESH" method in a "VPOLL" calendar component is used by "Voters" of an existing event to request an updated vpoll status from the poll "Owner". The "REFRESH" method MUST specify the "UID" property of the poll to update. The "Owner" responds with a METHOD=REQUEST giving the latest status and version of the poll.

This method type has a "METHOD" property with the value "REFRESH" and a single VPOLL object. That object MUST contain the properties shown below and no others.

Table 10 — Constraints for a METHOD:REFRESH of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST be REFRESH.
VPOLL	1	
PARTICIPANT	1	MUST identify the requester as a voter.
DTSTAMP	1	
UID	1	MUST be the UID associated with original REQUEST.

9.3.7. Method: 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 "Owner" participant MUST be present in the confirmed poll component. All "Voters" MUST be present. The selected component(s) according to the poll mode SHOULD NOT be present in the poll component. Clients receiving this message may store the confirmed items in their calendars.

This method type has a "METHOD" property with the value "POLLSTATUS" and one or more VPOLL objects. Those objects MUST contain the properties shown below and no others.

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

Table 11 — Constraints for a METHOD:POLLSTATUS of a VPOLL

Component/Property	Presence	Comment
METHOD	1	MUST equal POLLSTATUS.
VPOLL	1+	
PARTICIPANT	1+	The voters containing their current vote
COMPLETED	0 or 1	Only present for a completed poll
DTSTAMP	1	
DTSTART	0 or 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.

10. CalDAV Extensions

This specification extends IETF RFC 4791 in that it defines a new component and new iCalendar properties to be supported and requires extra definitions related to time-ranges and reports.

Additionally, it extends IETF RFC 6638 as a VPOLL component is a schedulable entity.

10.1. Calendar Collection Properties

This section defines new CalDAV properties for calendar collections.

10.1.1. CALDAV:supported-vpoll-component-sets

Name	supported-vpoll-component-sets
Namespace	urn:ietf:params:xml:ns:caldav
Purpose	Specifies the calendar component types (e.g., VEVENT, VTODO, etc.) and combination of types 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 IETF RFC 2518, Section 12.14.1).
Description	The CALDAV:supported-vpoll-component-sets property is used to specify restrictions on the calendar component types that VPOLL components may contain in a calendar collection. It also specifies the combination of allowed component types. Any attempt by the client to store VPOLL components with component types or combinations of types not listed in this property, if it exists, MUST result in an error, with the CALDAV:supported-vpoll-component-sets precondition Clause 10.2 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 supported-vpoll-component-sets
     (supported-vpoll-component-set*) >

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

Figure 26

```
<C:supported-vpoll-component-sets
     xmlns:C="urn:ietf:params:xml:ns:caldav">

  <!-- VPOLLs with VEVENT, VFREEBUSY or VTODO -->
  <C:supported-vpoll-component-set>
    <C:comp name="VEVENT" />
    <C:comp name="VFREEBUSY" />
    <C:comp name="VTODO" />
  </C:supported-vpoll-component-set>

  <!-- VPOLLs with just VEVENT or VFREEBUSY -->
  <C:supported-vpoll-component-set>
    <C:comp name="VEVENT" />
    <C:comp name="VFREEBUSY" />
  </C:supported-vpoll-component-set>

  <!-- VPOLLs with just VEVENT -->
  <C:supported-vpoll-component-set>
    <C:comp name="VEVENT" />
  </C:supported-vpoll-component-set>

  <!-- VPOLLs with just VTODO -->
  <C:supported-vpoll-component-set>
    <C:comp name="VTODO" />
  </C:supported-vpoll-component-set>
</C:supported-vpoll-component-sets>
```

Figure 27

10.1.2. 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 IETF RFC 2518, Section 12.14.1).

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 Clause 10.2 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)
```

Figure 28

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

Figure 29

10.1.3. 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 IETF RFC 2518, Section 12.14.1).

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 Clause 10.2 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)
```

Figure 30

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

```
>25</C:vpoll-max-active>
```

Figure 31

10.1.4. 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 [IETF RFC 2518, Section 12.14.1](#)).
Description	The `CALDAV:vpoll-max-voters` is used to specify a numeric value that indicates the maximum number of voters 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 voters per instance than this value MUST result in an error, with the `CALDAV:vpoll-max-voters` precondition [Clause 10.2](#) 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)
```

Figure 32

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

Figure 33

10.1.5. CalDAV:even-more-properties

10.1.6. Extensions to CalDAV scheduling

This specification extends [IETF RFC 6638](#).

Each section of Appendix A "Scheduling Privileges Summary" is extended to include VPOLL.

Any reference to the ATTENDEE property should be read to include the CALENDAR-ADDRESS property contained in the PARTICIPANT compoents. That is, for scheduling purposes the CALENDAR-ADDRESS property is handled in exactly the same manner as the ATTENDEE property.

10.2. 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: supported-vpoll-component-sets)	The VPOLL resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST contain a type or combination 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 [Clause 10.1.2](#) 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 VPOLLs being greater than the value of the `CALDAV:vpoll-max-active` property value [Clause 10.1.3](#) 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 voters represented by PARTICIPANT components less than or equal to the value of the `CALDAV:vpoll-max-voters` property value [Clause 10.1.4](#) on the calendar collection where the resource will be stored;

10.3. 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. For VPOLL queries this feature is disallowed. Time-range queries can only target the vpoll component itself.

10.3.1. 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:VPOLL
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>
```

Figure 34

10.4. CalDAV time ranges

"CALDAV:time-range XML Element" in [IETF RFC 4791, Section 9.9](#) describes how to specify time ranges to limit the set of calendar components returned by the server. This specification extends [IETF RFC 4791](#) 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                                          |
+---+---+---+---+---+-------------------------------------------------+
```

Figure 35

11. Security Considerations

Applications using these property need to be aware of the risks entailed in using the URIs provided as values. See [IETF RFC 3986](#) for a discussion of the security considerations relating to URIs.

12. IANA Considerations

12.1. Parameter Registrations

This document defines the following new iCalendar property parameters to be added to the registry defined in [IETF RFC 5545, Section 8.2.4](#):

Table 12

Property Parameter	Status	Reference
REQUIRED	Current	[Clause 5.4.1](#)
STAY-INFORMED	Current	[Clause 5.4.2](#)

12.2. Property Registrations

This document defines the following new iCalendar properties to be added to the registry defined in [IETF RFC 5545, Section 8.2.3](#):

Table 13

Property	Status	Reference
ACCEPT-RESPONSE	Current	[Clause 5.5.1](#)
POLL-ITEM-ID	Current	[Clause 5.5.3](#)
POLL-MODE	Current	[Clause 5.5.4](#)
POLL-PROPERTIES	Current	[Clause 5.5.5](#)
POLL-WINNER	Current	[Clause 5.5.6](#)
REPLY-URL	Current	[Clause 5.5.7](#)
RESPONSE	Current	[Clause 5.5.8](#)

12.3. 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

12.4. POLL-MODE Registrations

This document defines the following registered poll modes.

Table 14

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

Appendix A
(informative)
Open issues

public-comment: Not documented and was a parameter on something. Really sounds like a PARTICIPANT or VOTE property

Notifications: Need to do a section on what Notifications to support.
A. VPOLL is about to end and you haven't voted on it yet. Instead reuse VALARMS to notify the user?

Future: Restarting a confirmed/completed VPOLL What to do with changes to STATUS:CONFIRMED? Allow them or not? What do to that poll had a winning event or todo. Stress VPOLL UID MUST be unique Changing status back from CONFIRMED MUST adjust status of any events booked as a result of confirmation. MUST winning event be cancelled for POLL-MODE basic? No — voter has indicated now unable to attend — want to revote

Future: Voting on a confirmed/completed VPOLL Can a voter vote after completion? May be unable to attend and wants to indicate. Requires retention of VPOLL retention period Removed status

ORGANIZER/ATTENDEE validity Can a user create a poll with scheduled events where that user's isn't the organizer of the poll? So is there a requirement 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 events. Are there any other restrictions for components in a VPOLL? Add to security consideration

Update to existing event after poll confirm When voting on existing event — winning properties ONLY are merged in to the real event.

A.1. Advertising tasks

Use VPOLL for advertising a task to a pool of possible ATTENDEEs and then select the respondent to assign one or more assignees.

Introduce POLL-MODE:ASSIGNMENT

Need to indicate number of assignees required.

Potentially different types of response e.g. ACCEPT or DECLINE, or a weighting e.g. 0 — 100

Take into FREEBUSY discussion.

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

Guide for ATTENDEE roles chair, NON-PARTICIPANT etc

? — some iTip notes 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

Need to add example of freebusy in response

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//BedeworkCaldavTest//BedeworkCaldavTest
METHOD: REPLY
BEGIN:VPOLL
BEGIN:PARTICIPANT
```

```
PARTICIPANT-TYPE: OWNER
CALENDAR-ADDRESS:mailto:douglm@mysite.edu
UID:schedpart-1234567890
DTSTAMP:20120101T010000Z
SEQUENCE:0
SUMMARY:What to do this week
BEGIN:VFREEBUSY
.......
END:VFREEBUSY
END:PARTICIPANT
BEGIN:PARTICIPANT
PARTICIPANT-TYPE: VOTER
CALENDAR-ADDRESS:mailto:eric@example.com
UID:schedpart-0987654321
DTSTAMP:20120101T010000Z
SEQUENCE:0
SUMMARY:What to do this week
BEGIN:VFREEBUSY
.......
END:VFREEBUSY
END:PARTICIPANT
END:VPOLL
END:VCALENDAR
```

Figure A.1

Appendix B
(informative)
Change log

Calext V01: 2019-10-17 MD	Replace VVOTER and VOTER with PARTICIPANT.
Calext V00: 2019-05-17 MD	First calext version. Moved source to metanorma. No changes to specification.
V03: 2014-10-28 MD	— Add VVOTER and VOTE components. — Add RESPONSE property. — Remove RESPONSE parameter from VOTER.
V03: 2014-05-12 MD	— Add reply-url property and required parameter. — Fix ACCEPT-RESPONSE definition.
V02: 2014-05-12 MD	— Typos fixed, clarifications made. — Removed spurious COMMENT param. Switched some to PUBLIC-COMMENT — Changed STAY-INFORMED to remove boolean value type and state explicit TRUE/FALSE values. — iTip: Allow VPOLL DTSTART to be optional and allow VAVAILABILITY as subcomponent — iTip: fix broken table cells — Add POLL-PROPERTIES, POLL-WINNER to 5545 extensions table — Added Caldav scheduling section
V01: 2013-08-07 MD	— Removed method CONFIRM — Removed pollitemid from VPOLL abnf. Added text for pollwinner — Added POLL-WINNER and verbiage — Added STATUS values — Added RELTYPE=POLL — Added supported-vpoll-component-sets — Added CalDAV related parameters to VOTER — Removed bad CalDAV query example. State that queries cannot target the sub-components.
Initial version: 2012-11-02 MD	

Bibliography