Workgroup: calext
Internet-Draft:
draft-ietf-calext-ical-tasks-08
Updates: RFC5545 (if approved)
Published: 13 April 2024
Intended Status: Standards Track
Expires: 15 October 2024
Authors: A. Apthorp M. Douglass
DHL Express Bedework Commercial Services

Task Extensions to iCalendar

Abstract

This document updates and defines extensions to the Internet Calendaring and Scheduling Core Object Specification (iCalendar) (RFC5545) to provide improved status tracking, scheduling and specification of tasks.

It also defines how Calendaring Extensions to WebDAV (CalDAV) (RFC 4791) servers can be extended to support certain automated task management behaviours.

Status of This Memo

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

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

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

This Internet-Draft will expire on 15 October 2024.

respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

This document specifies extensions to the existing Internet Calendaring and Scheduling Core Object Specification (iCalendar) [RFC5545], and associated protocols, in order to enhance the structured communication and execution of tasks. The enhancements allow for the communication, time planning and scheduling of tasks by and between automated systems (e.g. in smart power grids, business process management systems) as well as for human centered tasks.

A "task" is a representation of an item of work assigned to an individual or organization. In the iCalendar Object Model [RFC5545] the representation of tasks is by "VTODO" calendar components. Tasks can be identified in a number of situations, either informally as ad-hoc tasks in personal "to-do" lists or more formally in:

*Business processes - ranging from repetitive workflows to adaptive cases and trouble ticketing

*Project Management - whether for large scale construction projects or collaborative software development

The extensions specified here are defined in the context of an overall architecture for task calendaring and scheduling.

1.1. Terms and Definitions

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Calendaring and scheduling roles are referred to in quoted-strings of text with the first character of each word in upper case. For example, "Organizer" refers to a role of a "Calendar User" (CU) within the scheduling protocol.

Calendar components defined by [RFC5545] are referred to with capitalized, quoted-strings of text. All calendar components start with the letter "V". For example, "VEVENT" refers to the event calendar component, "VTODO" refers to the to-do calendar component, and "VJOURNAL" refers to the daily journal calendar component.

Scheduling methods are referred to with capitalized, quoted-strings of text. For example, "REQUEST" refers to the method for requesting a scheduling calendar component be created or modified; "REPLY" refers to the method a recipient of a request uses to update their status with the "Organizer" of the calendar component.

Properties defined by [RFC5545] are referred to with capitalized, quoted-strings of text, followed by the word "property". For example, "ATTENDEE" property refers to the iCalendar property used to convey the calendar address of a "Calendar User".

Property parameters defined by this specification are referred to with capitalized, quoted-strings of text, followed by the word "parameter". For example, "VALUE" parameter refers to the iCalendar property parameter used to override the default data type for a property value.

Enumerated values defined by this specification are referred to with capitalized text, either alone or followed by the word "value".

In tables, the quoted-string text is specified without quotes in order to minimize the table length.

Terms defined and used in this specification include:

Assignee A calendar user assigned to perform a given task. An assignee is equivalent to an attendee of an event.

BPMS Business Process Management Software

Calendar User (CU) A person or software system that accesses or modifies calendar information.

Calendar User Agent (CUA) This may be

1. Software with which the calendar user communicates with a calendar service or local calendar store to access calendar information.

2. Software that gathers calendar data on the Calendar User's behalf.

Candidate A calendar user who might be able to perform a given task, prior to actually being assigned the task, e.g., a

dispatcher has a list of taxi drivers (candidates) from which one will be selected to pick-up a passenger.

Organizer A calendar user who creates a calendar item, requests free/busy information, or published free/busy information. It is an Organizer who invites Attendees [RFC5545].

Observer A calendar user interested in a calendar component, e.g., a manager may have interest in all tasks that have not been completed. Often represented as an attendee with ROLE=NON-PARTICIPANT.

Resource A resource in the scheduling context is any shared entity that can be scheduled by a calendar user, but does not control its own attendance status. Resources can be of "Location", "Equipment", or "Role" type.

Task A representation of an item of work that can be assigned to one or more task actor assignees. In [RFC5545], these are "VTODO" calendar components, which are groupings of component properties and possibly "VALARM" calendar components that represent an action-item or assignment.

2. Task Architecture

A reference architecture for task calendaring and scheduling is defined in order to identify the key logical elements involved in task management and the interfaces between them to enable interoperability. The logical elements identified here establish an appropriate separation of concerns and clarify the responsibilities of different elements. However, the architecture does not prescribe a binding or packaging of elements, i.e., software systems may be developed where some elements are tightly bound and the interfaces between bound elements are not exposed. The task architecture is also described in [TARCH].

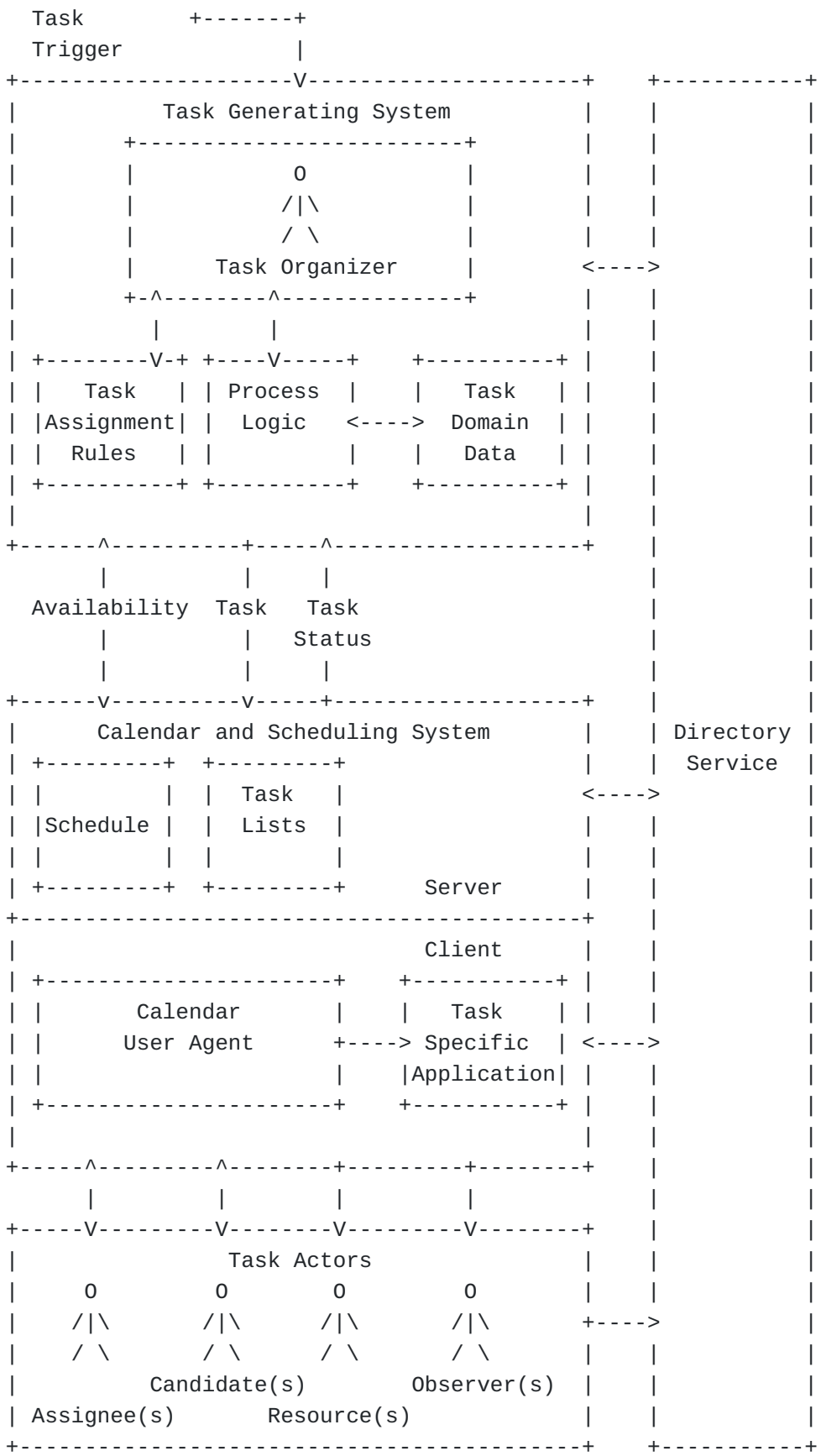

Figure 1: Task architecture diagram

3. Task Architecture Elements

The following logical elements form the task architecture that this specification is based on:

Task Actors Various calendar users that may be involved in the monitoring or performing of a task. The set of actors includes: Organizers, Observers, Resources, Assignees, and Candidates.

Task Organizer The Organizer of a task.

Task Domain Data This is any domain specific data that may be acted on or provides context to it in performing a task.

Task Specific Application A task specific application renders the data concerning the task (including task domain data) for presentation and manipulation by a task actor.

Process Logic Determines under what conditions a task (or tasks) is generated and the actions to take on completion, or some other status event occurring (or not) on the task.

Task Trigger This is some event that gives rise to the generation of a task according to Process Logic. Task triggers can come from many sources including, for example; a task being requested through the calendaring system, a status change in the progression of a business process being managed by a business process management or Enterprise resource planning (ERP) system.

Task Assignment Rules Govern how actors are assigned to a task. A range of different assignment patterns [WfRP] may be considered, including the two general cases:

a. Delegation to a named actor or group of actors

b. Advertising to a pool of actors for self-selection

In either case the assignment may be made based on a variety of criteria including, name, availability, skills, capacity, etc.

Task Generating System A system that creates and assigns tasks in response to some initiating event (task trigger). Task creation is according to Process Logic with task assignment determined by Task Assignment Rules. This system also tracks the status of tasks and will initiate further actions based upon the status. A task generating system can take many forms, for example; Business Process Management System, Project Management System, Bug Tracking System, Building Control System. A Task Generating System may also be a human. In iCalendar terms the Task Generating System is the organizer.

Human Task Generation
: Task creation, assignment and tracking coordinated by a human organizer is a special case of a task generating system. In this case Task Assignment Rules and Process Logic may be either explicit or tacit.

Directory Service
: A software system that stores and provides access to information providing details of task actors that may participate or be interested in a task.

Calendar and Scheduling System
: A software system that stores, publishes and synchronizes calendar data such as events, tasks and journal entries for actors. In the context of tasks this includes schedules (i.e. allocated time and availability to perform tasks) and task lists. A calendar and scheduling system typically consists of server and client software components.

It is not within the scope of this document to specify how Process Logic or Task Assignment Rules are codified. Such logic and rules may be codified in a variety of ways, including traditional programming languages (e.g. C++, Java) or process modelling languages (e.g. BPMN [BPMN]).

4. Architecture Foundations

The key standards that enable interoperability between the logical elements of the architecture are the Internet Calendaring and Scheduling Core Object Specification (iCalendar) [RFC5545] and associated protocols. Task and task status are represented by the iCalendar "VTODO" component. Protocols include, in particular, the iCalendar Transport-Independent Interoperability Protocol (iTIP) [RFC5546] for task assignment and scheduling, and Calendaring Extensions to WebDAV (CalDAV) [RFC4791] for client server communication.

Additionally, this specification uses definitions from Support for iCalendar Relationships [RFC9253]. The LINK, REFID, RELATED-TO and CONCEPT properties enable context and a rich set of relationships between tasks and other iCalendar components to be specified.

5. Task Extensions

In order to support the task architecture described in Section 2, this document defines a number of extensions to the current iCalendar standards in the areas of:

Task Specification
improved ability to specify domain specific tasks

Task Deadlines, Milestones and Time Planning clarification of deadlines and extension for task duration to support task time planning

Task Scheduling and Assignment ensure support for common pattens of scheduling and assigning tasks

Task Status Tracking improved granularity in status tracking information and alerting task actors to pending or actual task status changes

These extensions are supported mainly by additions to the properties and parameters used within the "VTODO" component.

6. Task Specification

The specification of tasks must be semantically explicit in order for them to be managed within the context of a business process or project, and be understood by both humans and IT systems. The current VTODO component only provides for simple ad-hoc tasks or 'to do' lists, and is therefore extended by this specification as follows:

Task type explicitly what type of task is to be performed is identified.

Task context and relationships how a specific task relates to other tasks and other objects that need to be understood for the effective execution of a task.

Task specific data the form and content of domain data provided as input to a task and/or that may be output from a task.

Organizer and attendee recognizes that a task organizer or attendee can be an automated system.

6.1. Task type

The [RFC9253] CONCEPT property is used to identify the type of task, for example;

CONCEPT:http://example.com/task/delivery

6.2. Task Context and Relationships

The [RFC9253] LINK property specifies a link to external information, which may be context to the task. For example:

```
LINK;LINKREL=SOURCE:http://example.com/package/1234567890

LINK;LINKREL=describedby:mid:752142.141482.307E5@mx123.example.com
```

The external information may be data to be manipulated in performing the task. See Section 6.3.

The [RFC9253] REFID property is used to identify a key used to group tasks by that key.

```
REFID:Manhattan

REFID:1234567890
```

Extensions to the RELATED-TO property defined in [RFC9253] allow temporal relationships between tasks as found in project management to be specified as well as parent/child relationships and dependencies (DEPENDS-ON). Tasks (VTODOs) may also be related to other calendar components; for example to a VEVENT to block time to perform a task.

6.3. Task Specific Data

The LINK property can be used to relate a domain specific service to the task. For example, it might be a URI pointing to a web page where the status of the task can be directly manipulated.

```
LINK;LINKREL="vacation-system";VALUE=URI:
 http://example.com/vacation-approval?id=1234
```

Additionally, it might be used to link data specific to the task, for example an electronic copy of a signature taken to confirm delivery of a package.

```
LINK;LINKREL="electronic-signature";VALUE=URI:
 http://example.com/delivery/sig1234.jpg
```

7. Task Deadlines, Milestones and Time Planning

7.1. Deadlines

Deadlines for starting and finishing a task are defined by the DTSTART, DUE and DURATION properties. DTSTART represents the earliest start time for beginning work on a task. DUE, or DTSTART + DURATION represent the latest finish time for a task. Thus, these

properties define a "window" within which a task has to be performed. However, there is currently no way to indicate how long the task is expected to take. This document defines a new property, in Section 12.1 ESTIMATED-DURATION, to allow the estimated time that a task should take to be specified separately from the deadlines for starting and finishing a task. This supports time planning by enabling calendar user agents to display when tasks should occur and therefore allow calendar users to visualize when tasks should be performed and allocate time to them.

7.2. Milestones

A task that has intermediary deadlines (i.e., milestones) SHOULD be expressed by child VTODO components (i.e., sub-tasks associated with each of the milestones) in conjunction with the RELATED-TO property to relate the parent and child tasks.

8. Task Scheduling and Assignment

Tasks are assigned to actors using one or more [RFC5545] ATTENDEE properties and/or one or more [RFC9073] PARTICIPANT components.

Communication of task assignment or delegation to one or more actors who are allocated to a task by the organizer is directly supported by iTIP, i.e., all included ATTENDEES in an iTIP REQUEST are expected to perform the task.

The offering or advertising of a task to one or more (potential) actors where only one or a subset of the candidates may accept the task will be addressed by a later specification.

9. Status Reporting

9.1. Improved granularity in status reporting information

This document defines a new "VSTATUS" component (see section Section 14.1) that can be used to group related information about the status of the task. This might include information on why (REASON) and when (DTSTAMP) a status has changed. In addition, new status values are specified to provide for task suspension, failure and preparation.

Note that while VSTATUS is intended to allow multiple date-stamped status changes to be stored it is not intended to be used as a history of changes to a tasks properties.

9.2. Relating reason and comments to ATTENDEE status changes.

The [RFC9073] PARTICIPANT component can be used to provide additional information about why an ATTENDEE participation status

has changed. The COMMENT property can also be used to include additional human-readable information about why the associated STATUS or ATTENDEE property changed. For example, if a driver failed to deliver a package because of a puncture it might be expressed as

```
BEGIN:VSTATUS
STATUS:FAILED
REASON:https://example.com/reason/delivery-failed
SUBSTATE:ERROR
DTSTAMP:20130212T120000Z
COMMENT:Breakdown
END:VSTATUS

ATTENDEE;PARTSTAT=FAILED:mailto:xxx@example.com
...
BEGIN:PARTICIPANT
CALENDAR-ADDRESS:mailto:xxx@example.com
DTSTAMP:20130226T1104510Z
REASON:https://example.com/reason/van-break-down
COMMENT:Puncture
END:PARTICIPANT
```

9.3. Comments associated to reasons and status changes

Multiple comments and reasons may have the same status. As situations change further VSTATUS components can be added to provide additional information..

```
CONCEPT:https://example.com/task/delivery
BEGIN:VSTATUS
STATUS:FAILED
SUBSTATE:ERROR
DTSTAMP:20220212T104900Z
COMMENT:Out of time
END:VSTATUS
BEGIN:VSTATUS
STATUS:FAILED
COMMENT:Traffic Accident on E44
REASON:https://example.com/reason/traffic
DTSTAMP:20220212T110451Z
END:VSTATUS
BEGIN:VSTATUS
STATUS:FAILED
COMMENT:Arrived after office hours
REASON:https://example.com/reason/closed
DTSTAMP:20220212T180451Z
END:VSTATUS
```

9.4. Task Alerts and Notifications

Different needs to alert or notify task actors of pending or actual task status changes are recognized:

Alarms Alarms (VALARM components) operate in the calendar user agent space to notify the task actor of a pending task state for a task they are assigned to or are interested in.

Current standards (see [RFC9074]) indicate VALARMs SHOULD be removed from incoming data and many systems in fact do so. In a task assignment scenario it may be appropriate for the organizer to be able to set alarms for the participants. A system implementing these standards may choose to preserve VALARMs but sending a task via some external service may result in them being removed. This issue is not addressed by this specification.

Escalations An escalation or notification to the ATTENDEE, ORGANIZER, or other task actor may be required if a deadline associated with a task is exceeded or for some other reason. Process Logic identifying when and who to propagate escalations to is the responsibility of the Task Generating System, e.g., a BPMS.

Notifications Task actors (observers) not directly involved in performing a task but with a known interest in a given task's status can be identified by the PARTICIPANT component [RFC9073] against certain components e.g. ALARM, to identify which task events the stakeholder/party is interested in. Notifications on shared calendars will allow task actors to register an interest in changes to tasks within a calendar (see Appendix A).

9.5. Automated Status Changes

A new property, TASK-MODE, is introduced to instruct servers to apply automated operations for changing the status of a task.

10. Modifications to Calendar Components

The following changes to the syntax defined in iCalendar [RFC5545] are made here. New elements are defined in subsequent sections.

```
; Addition of VSTATUS as a valid component for VEVENT
eventc     = "BEGIN" ":" "VEVENT" CRLF
             eventprop *alarmc *participantc *locationc *resourcec *stat
             "END" ":" "VEVENT" CRLF

; Addition of VSTATUS as a valid component for VTODO
todoc      = "BEGIN" ":" "VTODO" CRLF
             todoprop *alarmc *participantc *locationc *resourcec *statu
             "END" ":" "VTODO" CRLF

; Addition of properties ESTIMATED-DURATION and TASK-MODE to VTODO
todoprop =/ est-duration /
            task-mode

; Addition of VSTATUS as a valid component for VJOURNAL
journalc   = "BEGIN" ":" "VJOURNAL" CRLF
              jourprop *statusc
              "END" ":" "VJOURNAL" CRLF


; Addition of VSTATUS as a valid component for VFREEBUSY
freebusyc  = "BEGIN" ":" "VFREEBUSY" CRLF
             fbprop *participantc *locationc *resourcec *statusc
             "END" ":" "VFREEBUSY" CRLF

; Addition of VSTATUS as a valid component for PARTICIPANT
participantc = "BEGIN" ":" "PARTICIPANT" CRLF
               partprop *locationc *resourcec
               *statusc
               "END" ":" "PARTICIPANT" CRLF


; Addition of propertY REASON to PARTICIPANT
partprop =/ reason
```

11. New Parameter Values

11.1. Redefined VTODO Participant Status

Participant status parameter type values are defined in Section 3.2.12 of [RFC5545]. This specification redefines that type to include the new value FAILED for VTODO iCalendar components.

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

```
partstat-todo    =/ *("FAILED")  ; To-do cannot be completed
```

Example

```
ATTENDEE;REASON="https://example.com/reason/not-enough-time";
 PARTSTAT=FAILED:mailto:jsmith@example.com
```

12. New Properties

12.1. Estimated Duration

Property Name ESTIMATED-DURATION

Purpose This property specifies the estimated positive duration of time the corresponding task will take to complete.

Value Type DURATION

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

Conformance This property can be specified in "VTODO" calendar components.

Format Definition This property is defined by the following notation:

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

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

Description In a "VTODO" calendar component the property MAY be used to specify the estimated duration for the to-do, with or without an explicit time window in which the event should be started and completed. When present, DTSTART and DUE/DURATION represent the window in which the task can be performed. ESTIMATED-DURATION SHOULD be passed from ORGANIZER to ATTENDEE in iTIP [RFC5546] messages.

Example The following is an example of this property that estimates the duration of a task to be one hour:

```
ESTIMATED-DURATION:PT1H
```

12.2. Reason

Property name REASON

Purpose To indicate the reason for a status change or change of attendee participation status.

Value Type URI

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

Conformance This property can be specified in "VSTATUS" and PARTICIPANT calendar components.

Format Definition This property is defined by the following notation:

```
reason      = "REASON" reasonparam ":" uri CRLF

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

Description This property allows the change in status of a task or participant status to be qualified by the reason for the change with a codified reason. Typically, reasons are defined within the context of the task type and therefore SHOULD include the name-space of the authority defining the task.

Example REASON:https://example.com/reason/delivered-on-time

12.3. Sub-State

Property name SUBSTATE

Purpose To provide additional granularity of task status for e.g. IN-PROCESS.

Value Type TEXT

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

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

Format Definition This property is defined by the following notation:

```
substate          = "SUBSTATE" substateparam ":" substatevalue CRLF

substateparam     = *(";" other-param)

substatevalue       = ("OK"        ; everything is fine(the default)
                     / "ERROR"     ; something is wrong (the REASON
                     ;               code explains why)
                     / "SUSPENDED" ; waiting on some other task to
                     ;               complete or availability of a
                     ;               resource (REASON code explains
                     ;               why)
                     / iana-token) ; Other IANA-registered type
```

Description The sub-state property allows additional qualification and granularity of states to be recorded, in particular for the IN-PROCESS state. It allows individual sub-states to be recorded without the need to define and publish a sub-task associated with a parent task purely to track that a particular state has been reached. This property also allows parallel states to be expressed e.g. that a task has been suspended at whatever state it has reached.

Example

```
BEGIN:VSTATUS
STATUS:FAILED
REASON:https://example.com/reason/no-one-home
SUBSTATE:ERROR
END:VSTATUS

BEGIN:VSTATUS
STATUS:IN-PROCESS
REASON:https://example.com/reason/paint-drying
SUBSTATE:SUSPENDED
END:VSTATUS
```

12.4. Task Mode

Property Name TASK-MODE

Purpose This property specifies automatic operations that servers acting on behalf of the organizer apply to tasks based on changes in attendee status (PARTSTAT).

Value Type TEXT

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

Conformance This property can be specified zero or more times in a "VTODO" calendar component.

Format Definition

This property is defined by the following notation:

```
task-mode   = "TASK-MODE taskmodeparam ":" taskvalue
              *("," taskvalue) CRLF

taskvalue   = "AUTOMATIC-COMPLETION" ; set STATUS completed
                ;if all attendees have completed
                / "AUTOMATIC-FAILURE"
                / "SERVER"
                / "CLIENT"
                / iana-token
                / x-name

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

Description In a "VTODO" calendar component this property MAY be used to indicate to servers how they can automatically change the state of the task based on iTIP replies from Attendees. For example, the server can automatically set the overall task status to COMPLETED when every attendee has marked their own status (PARTSTAT) as COMPLETED, or the server could mark the task as FAILED if its DUE date passes without it being completed. TASK-MODE processing is performed on the organizer's copy of the task.

To set the status, add a VSTATUS component as specified in Section 14.1.

The property value is a list of one or more IANA registered tokens that defines modes to be used for the task. This specification defines three modes which are described in the following subsections.

Examples

```
TASK-MODE:AUTOMATIC-COMPLETION,AUTOMATIC-FAILURE
TASK-MODE:SERVER
TASK-MODE:AUTOMATIC-FAILURE
```

AUTOMATIC-COMPLETION Task Mode The task mode value "AUTOMATIC-COMPLETION" indicates to the server that it can change the "VTODO" component's status to "COMPLETED" as soon as all ATTENDEEs in the task have replied with a "PARTSTAT" parameter set to "COMPLETED".

AUTOMATIC-FAILURE Task Mode The task mode value "AUTOMATIC-FAILURE" indicates to the server that it SHOULD change the "VTODO" component's status to "FAILED" if either:

1. the PARTSTAT of one ATTENDEE is set to FAILED; or

2. the current time is past the effective due date of the component and the task has not yet been completed.

NOTE: The effective due date is either the "DUE" property value or the combination of the "DTSTART" and "DURATION" property values.

CLIENT Task Mode The task mode value "CLIENT" is an instruction to the server to honour the status set by the client.

SERVER Task Mode The task mode value "SERVER" indicates to the server that it can change the "VTODO" component's status to an appropriate value, based on implementation defined "business rules", as ATTENDEE responses are processed or as deadlines related to the task pass.
The server can add this property to a "VTODO" component to indicate to the client that it will be managing the status.

13. Property Extensions and Clarifications

13.1. Updated DURATION Property definition

[RFC5545] section 3.6.2 introduced a constraint on the duration property requiring that a DURATION MUST be accompanied by a DTSTART. This constraint is dropped reverting to the situation as specified previously.

Thus the text:

```
; Either 'due' or 'duration' MAY appear in
; a 'todoprop', but 'due' and 'duration'
; MUST NOT occur in the same 'todoprop'.
; If 'duration' appear in a 'todoprop',
; then 'dtstart' MUST also appear in
; the same 'todoprop'.
```

is replaced by

```
; Either 'due' or 'duration' MAY appear in
; a 'todoprop', but 'due' and 'duration'
; MUST NOT occur in the same 'todoprop'.
```

This allows a VTODO to only have a DURATION property.

Furthermore, the following text:

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

is replaced by

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

13.2. Redefined STATUS Property

The Status property is defined in Section 3.8.1.11 of [RFC5545]. This specification extends that property to include new values associated with VTODO iCalendar components (See Appendix A for examples of the task state lifecycle).

Format Definition The "STATUS" property parameter list is augmented as follows:

```
statvalue-todo = / "PENDING"    ;Indicates a to-do has been
                                ;created and accepted, but has
                                ; not yet started.
                 / "FAILED"     ;Indicates to-do has failed.
;Extended status values for "VTODO".
```

Description:

PENDING - A to-do has been created and accepted but has not yet started and is ready to start subject to other dependencies (e.g. preceding task or DTSTART). This is the default state.

FAILED - to-do has failed and may need some follow-up from the organizer to re-schedule or cancel

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

```
STATUS:FAILED
```

14. New Components

14.1. Status Component

Component Name VSTATUS

Purpose This component allows information to be associated with a status, for example comments and date stamps.

Conformance This component can be specified multiple times in any calendar component.

Description This component provides a way for multiple date-stamped statuses to be associated with a component such as a participant, task or event.

This component may be added to the [RFC9073] PARTICIPANT component to allow participants in a task to specify their own status.

For backwards compatibility, when a VSTATUS component is added the [RFC5545] STATUS property MUST be set on the parent component.

Format Definition This component is defined by the following notation:

```
statusc = "BEGIN" ":" "VSTATUS" CRLF
          statusprop
          "END" ":" "VSTATUS" CRLF

statusprop     = *(
                 ;
                 ; The following is REQUIRED,
                 ; but MUST NOT occur more than once.
                 ;
                 status /
                 ;
                 ; The following are OPTIONAL,
                 ; but MUST NOT occur more than once.
                 ;
                 description / dtstamp / reason / substate / summary
                 ;
                 ; The following are OPTIONAL,
                 ; and MAY occur more than once.
                 ;
                 comment / styleddescription / iana-prop
                 ;
                 )
```

Examples

```
BEGIN:VSTATUS
STATUS:COMPLETED
REASON: https://example.com/reason/delivered-on-time
DTSTAMP:20220212T120000Z
END:VSTATUS
```

15. CalDAV Support for Task Mode

The CalDAV [RFC4791] calendar access protocol allows clients and servers to exchange iCalendar data. With the introduction of the "TASK-MODE" property in this specification, different automated task management behaviours may be delegated to the server by the Task Organizer depending upon the value of "TASK-MODE".

In order for a CalDAV client to know what task modes are available, a CalDAV server advertises a CALDAV:supported-task-mode-set WebDAV property on calendar home or calendar collections if it supports the

use of the "TASK-MODE" property as described in this specification. The server can advertise a specific set of supported task modes by including one or more CALDAV:supported-task-mode XML elements within the CALDAV:supported-task-mode-set XML element. If no CALDAV:supported-task-mode XML elements are included in the WebDAV property, then clients can try any task mode, but need to be prepared for a failure when attempting to store the calendar data.

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

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

It is possible for a "TASK-MODE" value to be present in calendar data on the server being accessed by a client that does not support the "TASK-MODE" property. It is expected that existing clients, unaware of "TASK-MODE", will fail gracefully by ignoring the calendar property.

15.1. CALDAV:supported-task-mode-set Property

Name supported-task-mode-set

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

Purpose Enumerates the set of supported iCalendar "TASK-MODE" element values supported by the server.

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

Description See above.

Definition

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

Example

```
<C:supported-task-mode-set xmlns:C="urn:ietf:params:xml:ns:caldav">
  <C:supported-task-mode>AUTOMATIC-COMPLETION</C:supported-task-mode>
  <C:supported-task-mode>AUTOMATIC-FAILURE</C:supported-task-mode>
  <C:supported-task-mode>SERVER</C:supported-task-mode>
  <C:supported-task-mode>CLIENT</C:supported-task-mode>
</C:supported-task-mode-set>
```

16. Security Considerations

This specification introduces no new security considerations beyond those identified in [RFC5545], [RFC5546] and [RFC4791].

17. IANA Considerations

17.1. New and updated iCalendar Elements Registration

This specification updates [RFC5545] by adding and updating a number of elements. The procedures and templates specified in Section 8.2 of [RFC5545]

17.1.1. Initialization of the Status registry

This specification updates [RFC5545] by adding a Status value registry to the iCalendar Elements registry and initializing it as per [RFC5545].

Name	Status	Reference
CANCELLED	Current	Section 3.8.1.11 of [RFC5545]
COMPLETED	Current	Section 3.8.1.11 of [RFC5545]
CONFIRMED	Current	Section 3.8.1.11 of [RFC5545]
DRAFT	Current	Section 3.8.1.11 of [RFC5545]
FINAL	Current	Section 3.8.1.11 of [RFC5545]
IN-PROCESS	Current	Section 3.8.1.11 of [RFC5545]
NEEDS-ACTION	Current	Section 3.8.1.11 of [RFC5545]
TENTATIVE	Current	Section 3.8.1.11 of [RFC5545]

Table 1: Initial Status Value Registry

17.1.2. Update of the Status registry

This specification further updates the Status registry with additional values defined in this document.

Value	Status	Reference
PENDING	Current	This Spec, Section 13.2
FAILED	Current	This Spec, Section 13.2

Table 2: Updated Status Value Registry

17.1.3. Sub-State value registry

The following table has been used to initialize the Sub-State registry.

Substate	Status	Reference
OK	Current	This Spec, Section 12.3
ERROR	Current	This Spec, Section 12.3
SUSPENDED	Current	This Spec, Section 12.3

Table 3: Sub-State registry

17.1.4. Task Mode value registry

The following table has been used to initialize the Task Mode registry.

Task Mode	Status	Reference
AUTOMATIC-COMPLETION	Current	This Spec, Section 12.4
AUTOMATIC-FAILURE	Current	This Spec, Section 12.4
CLIENT	Current	This Spec, Section 12.4
SERVER	Current	This Spec, Section 12.4

Table 4: Task Mode Value Registry

17.1.5. Participation Statuses registry

The following table has been used to update the Participation Statuses registry.

Value	Status	Reference
FAILED	Current	This Spec, Section 11.1

Table 5: Participation Statuses Registry

17.1.6. Components Registry

The following table has been used to update the Components registry defined in Section 8.3.1 of [RFC5545].

Component	Status	Reference
VSTATUS	Current	This Spec, Section 14.1

Table 6: Updated Components Registry

17.1.7. Properties registry

The following table has been used to update the Properties registry defined in Section 8.3.2 of [RFC5545].

Property	Status	Reference
ESTIMATED_DURATION	Current	This Spec, Section 12.1
REASON	Current	This Spec, Section 12.2
SUBSTATE	Current	This Spec, Section 12.3

Property	Status	Reference
STATUS	Current	This Spec, Section 13.2
TASK-MODE	Current	This Spec, Section 12.4

Table 7: Updated Properties Registry

18. Acknowledgements

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

John Chaffee, Marten Gajda, Ken Murchison

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

19. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", IETF, DOI 10.17487/RFC2119, BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault, "Calendaring Extensions to WebDAV (CalDAV)", IETF, DOI 10.17487/RFC4791, RFC 4791, March 2007, <https://www.rfc-editor.org/info/rfc4791>.

[RFC4918] Dusseault, L., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", IETF, DOI 10.17487/RFC4918, RFC 4918, June 2007, <https://www.rfc-editor.org/info/rfc4918>.

[RFC5545] Desruisseaux, B., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", IETF, DOI 10.17487/RFC5545, RFC 5545, September 2009, <https://www.rfc-editor.org/info/rfc5545>.

[RFC5546] Daboo, C., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", IETF, DOI 10.17487/RFC5546, RFC 5546, December 2009, <https://www.rfc-editor.org/info/rfc5546>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", IETF, DOI 10.17487/RFC8174, BCP 14, RFC

8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9073] Douglass, M., "Event Publishing Extensions to iCalendar", IETF, DOI 10.17487/RFC9073, RFC 9073, August 2021, <https://www.rfc-editor.org/info/rfc9073>.

[RFC9074] Daboo, C. and K. Murchison, ""VALARM" Extensions for iCalendar", IETF, DOI 10.17487/RFC9074, RFC 9074, August 2021, <https://www.rfc-editor.org/info/rfc9074>.

[RFC9253] Douglass, M., "Support for iCalendar Relationships", IETF, DOI 10.17487/RFC9253, RFC 9253, August 2022, <https://www.rfc-editor.org/info/rfc9253>.

20. Informative References

[BPMN] "Business Process Model and Notation", OMG BPMN 2.0.2, January 2014, <https://www.omg.org/spec/BPMN/2.0.2/About-BPMN>.

[TARCH] Apthorp, A., Daboo, C., and M. Douglass, "CalConnect, Task Architecture V1.0", (CalConnect Task Architecture V1.0, <http://www.calconnect.org/architectures/Task%20Architecture%201.0.pdf>.

[EDISTS] UN Economic Commission for Europe, "STS STATUS", UN/EDIFACT, D14.A, 30 April 2014, <http://www.unece.org/fileadmin/DAM/trade/untdid/d14a/trsd/trsdsts.htm>.

[WfRP] Russell, N., ter Hofstede, A.H.M., Edmond, T., and W.M.P. van der Aalst,, "Workflow Resource Patterns", WfRP, 2004, <http://www.workflowpatterns.com/patterns/resource/>.

[WSCal] Considine, T., Considine, T., and M. Douglass, "WS-Calendar Version 1.0", OASIS WS-Calendar-CS01, 31 July 2011, <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.html>.

[WSHT] Ings,, D., "Web Services - Human Task (WS-HumanTask) Specification Version 1.1", OASIS ws-humantask-v1.1-CS01, 17 August 2010, <http://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html>.

Appendix A. Examples of Task State Lifecycle

A.1. Simple Case Status Change

	STATUS	PARTSTAT	Action
1	-	-	Organizer draft
2	NEEDS-ACTION	NEEDS-ACTION	Organizer sends iTIP request
3	NEEDS-ACTION	ACCEPTED	Attendee reply
4	PENDING	ACCEPTED	Task accepted but waiting on some "trigger" to start (e.g. another task has to finish first)
5	IN-PROCESS	IN-PROCESS	Attendee reply now working on the task
6	IN-PROCESS	COMPLETED	Attendee reply completed
7	COMPLETED	COMPLETED	Organizer changes overall state

Table 8: Example of status changes in assigning and performing a task with one attendee.

A.2. Example for multiple Attendees

Example of status changes in assigning and performing a task with two attendees (A1 and A2).

	STATUS	PARTSTAT (A1)	PARTSTAT (A2)	Action
1	-	-	-	Organizer draft.
2	NEEDS-ACTION	NEEDS-ACTION	NEEDS-ACTION	Organizer sends iTIP request.
4	NEEDS-ACTION	ACCEPTED	NEEDS-ACTION	Attendee 1 reply.
5	NEEDS-ACTION	ACCEPTED	ACCEPTED	Attendee 2 reply.
6	PENDING	ACCEPTED	ACCEPTED	Task accepted but waiting on some"trigger" to start (e.g. another task has to finish first)
7	IN-PROCESS	ACCEPTED	IN-PROCESS	Attendee 2 reply now working on the task.
8	IN-PROCESS	IN-PROCESS	IN-PROCESS	Attendee 1 reply now working on the task.
9	IN-PROCESS	COMPLETED	IN-PROCESS	Attendee 1 reply Completed (overall status still IN-PROCESS).
10	IN-PROCESS	COMPLETED	COMPLETED	Attendee 2 reply Completed
11	COMPLETED	COMPLETED	COMPLETED	Organizer changes overall state once both attendees are finished.

Table 9: Example for multiple Attendees

NOTE: The logic for determining the status change to the VTODO is determined by the task organizer based on the ATTENDEE status and other business logic.

A.3. Example of Failure

Example of status changes for a task that fails.

	STATUS	PARTSTAT	Action
1	-	-	Organizer draft
2	NEEDS-ACTION	NEEDS-ACTION	Organizer sends iTIP request
3	NEEDS-ACTION	ACCEPTED	Attendee reply
4	IN-PROCESS	IN-PROCESS	Attendee reply now working on the task
5	IN-PROCESS	FAILED	Attendee reply task failed
6	FAILED	FAILED	Organizer changes overall state

Table 10: Example of Failure

Authors' Addresses

Adrian Apthorp
DHL Express
Fritz-Erler-Str. 5
Bonn
Germany

Email: adrian.apthorp@dhl.com

Michael Douglass
Bedework Commercial Services
226 3rd Street
Troy, NY
United States of America

Email: mdouglass@bedework.com