calsify
Internet-Draft
Updates: RFC5545 (if approved)
Intended status: Standards Track
Expires: 8 January 2026

A. Apthorp
DHL Express
M. Douglass
Bedework Commercial Services

7 July 2025

Task Extensions to iCalendar
draft-ietf-calext-ical-tasks-14

Abstract

The Internet Calendaring and Scheduling Core Object Specification (iCalendar) (RFC5545) VTODO calendar component has seen limited adoption and use, mainly for personal reminders and to-do lists.

This document updates and defines extensions to VTODO to provide improved status tracking, scheduling and specification of tasks to allow its use in other contexts, such as process control and project management.

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 8 January 2026.

Copyright Notice

Table of Contents

1. Introduction 3
1.1. Terms and Definitions 4
2. Task Architecture 6
2.1. Task Architecture Elements 8
2.2. Architecture Foundations 9
3. Task Extensions 9
4. Task Specification 10
4.1. Task type 10
4.2. Task Context and Relationships 11
5. Task Deadlines, Milestones and Time Planning 11
5.1. Deadlines 12
5.2. Milestones 12
6. Task Scheduling and Assignment 12
7. Status Reporting 12
7.1. Improved granularity in status reporting information . . 13
7.2. Comments associated to reasons and status changes 13
7.3. Updating the overall status 14
7.4. Relating reason and comments to "ATTENDEE" property status changes. 14
7.5. Task Alerts and Notifications 14
7.6. Automated Status Changes 15
8. Modifications to Calendar Components 15
9. New Parameter Values 16
9.1. Redefined VTODO Participant Status 16
10. New Properties 17
10.1. Estimated Duration 17
10.2. Reason 18
10.3. Substate 19
10.4. Task Mode 20
11. Property Extensions and Clarifications 22
11.1. Updated DURATION Property definition for VTODO 22
11.2. Redefined STATUS Property 23
12. New Components 23
12.1. Status Component 23
13. CalDAV Support for Task Mode 25
13.1. CALDAV:supported-task-mode-set Property 25
14. Security Considerations 26

15. IANA Considerations 26
15.1. Component Registrations 26
15.2. Property Registrations 26
15.3. Initialization of the Status Value registry 27
15.4. Substate Value registry 28
15.5. Task Mode Value registry 29
15.6. Participation Statuses registry 29
16. Acknowledgements 30
17. Normative References 30
18. Informative References 31
Appendix A. Examples of Task State Lifecycle 31
A.1. Simple Case Status Change 31
A.2. Example for multiple Attendees 32
A.3. Example of Failure 34
Authors' Addresses 34

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. taxi dispatch systems, in smart power grids (see [WsCalendar]), 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 in this document enhance and standardize the following:

The specification of tasks: how a task type is defined and the relationships between tasks See Section 4.

Their planning and scheduling: Improved specification of deadlines and estimated time. See Section 5

Task assignment, and status reporting: The addition of VSTATUS, see Section 8 allows for more detailed reports. See Section 7.

These extensions are defined within the context of an overall architecture for task calendaring and scheduling, which elaborates that supported by [RFC5545] and related standards.

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] and updating specifications are referred to with capitalized, quoted-strings of text, followed by the words "calendar component". For example, "VEVENT" calendar component refers to the event calendar component, "VTODO" calendar component refers to the to-do calendar component, and "VJOURNAL" calendar component 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] and updating specifications 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 [RFC5545] and updating specifications are referred to with lowercase, quoted-strings of text, followed by the word "parameter". For example, "value" parameter refers to the iCalendar property parameter used to override the default 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 a task.

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 the "role" parameter set to 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. Examples of such resources are locations such as a bookable conference room or equipment such as projectors.

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. Central to this architecture is the Calendar and Scheduling System. 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].

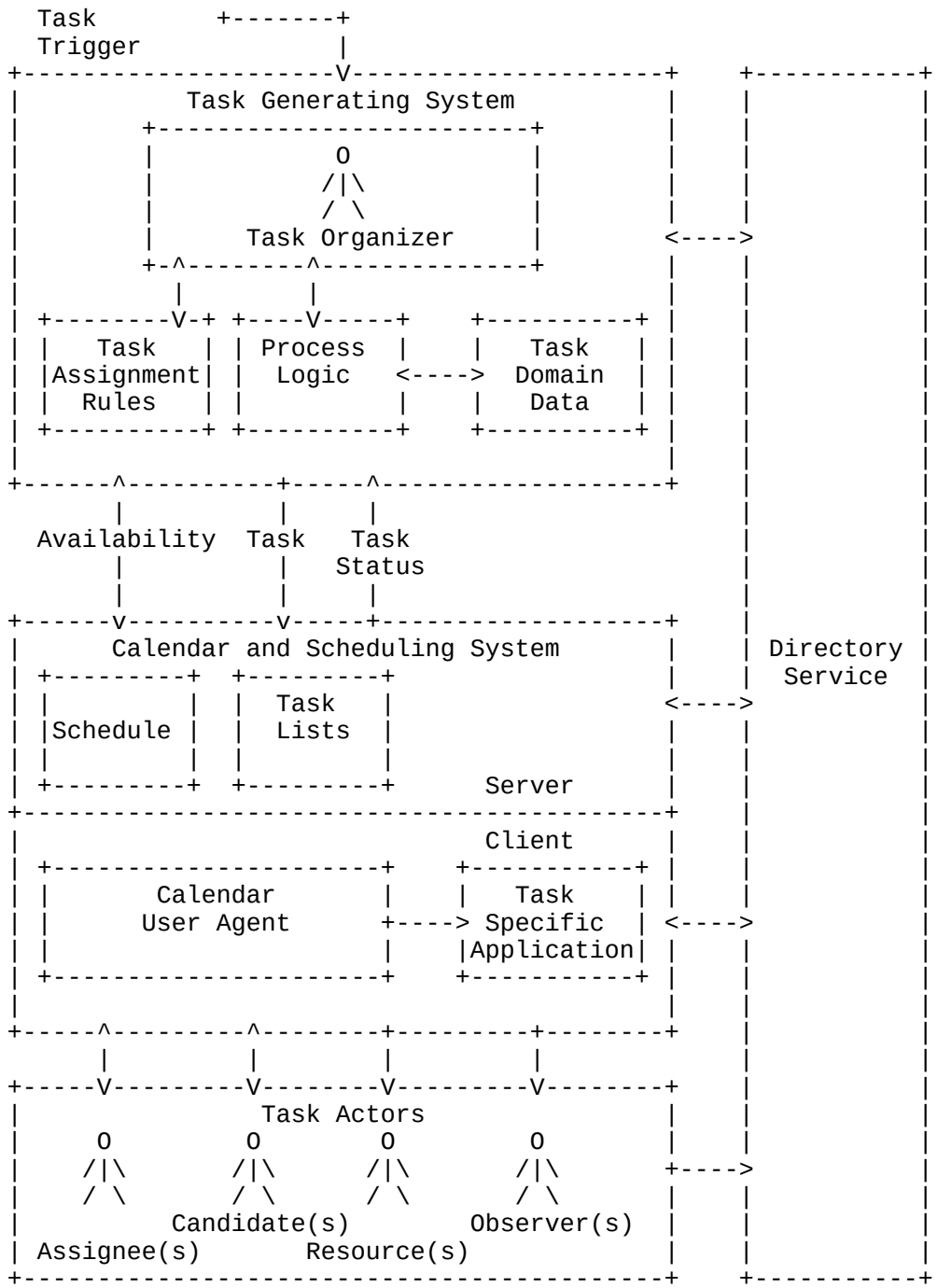
Task
Trigger
Task Generating System
Task Organizer
Task Assignment Rules
Process Logic
Task Domain Data
Availability
Task Status
Calendar and Scheduling System
Schedule
Task Lists
Server
Client
Calendar User Agent
Task Specific Application
Directory Service
Task Actors
Candidate(s)
Observer(s)
Assignee(s)
Resource(s)

Figure 1: Task architecture diagram

2.1. 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 creator of a task, who also determines task assignment and scheduling, and monitors task progress.

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.

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.

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]).

2.2. 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 "VTODO" calendar 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 calendar components to be specified.

3. Task Extensions

In order to support the task architecture described in Section 2, this document defines a number of extensions to iCalendar [RFC5545] 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 patterns of scheduling and assigning tasks

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

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

4. 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 [RFC5545] "VTODO" calendar 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.

4.1. Task type

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

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

4.2. 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=describedby;VALUE=URI:
 mid:752142.141482.307E5@mx123.example.com
```

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

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.

(Note: link relations below are for illustration only)

```
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
```

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 ("VTODO" calendar components) may also be related to other calendar components; for example to a "VEVENT" calendar component to block time to perform a task.

5. Task Deadlines, Milestones and Time Planning

5.1. Deadlines

Deadlines for starting and finishing a task are defined by the "DTSTART", "DUE" and "DURATION" properties. The "DTSTART" property represents the earliest start time for beginning work on a task. The "DUE", or "DTSTART" + "DURATION" properties represent the latest finish time for a task. Thus, these properties define a "window" within which a task has to be performed. However, [RFC5545] provides no way to indicate how long the task is expected to take. This document defines a new "ESTIMATED-DURATION" property, in Section 10.1, 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.

5.2. Milestones

A task that has intermediary deadlines (i.e., milestones) MUST be expressed by child "VTODO" calendar 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.

6. Task Scheduling and Assignment

Tasks are assigned to actors using one or more [RFC5545] "ATTENDEE" properties and/or one or more [RFC9073] "PARTICIPANT" calendar components as described in Section 7.4.

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 "ATTENDEE" properties 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.

7. Status Reporting

7.1. Improved granularity in status reporting information

This document defines a new "VSTATUS" calendar component (see section Section 12.1) that can be used to group related information about the status of the task. This might include information on why, the "REASON" property and when, the "DTSTAMP" property, a status has changed. In addition, new status values are specified to provide for task suspension, failure and preparation.

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

7.2. Comments associated to reasons and status changes

Multiple comments and reasons may have the same status. As situations change, further "VSTATUS" calendar 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
```

Note that the "VSTATUS" calendar component is not intended to be used as a history of changes to a tasks properties. The purpose of the "VSTATUS" calendar component is only to document changes related to fulfilling the tasks.

7.3. Updating the overall status

Only the Task Organizer, or the server acting as the Organizers proxy, may change or add "VSTATUS" calendar components and the "STATUS" property.

The overall VSTATUS will be changed in response to incoming Attendee replies. Each change of the overall VSTATUS MUST be accompanied by a change to the STATUS.

Note there is no defined ordering of properties and components so the DTSTAMP property SHOULD be set for each VSTATUS component to preserve ordering.

7.4. Relating reason and comments to "ATTENDEE" property status changes.

The [RFC9073] "PARTICIPANT" calendar component can be used to provide additional information about why an "ATTENDEE" property participation status has changed following the guidelines set out in Section 7.1 of [RFC9073]. 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

```
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
```

7.5. Task Alerts and Notifications

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

Alarms: "VALARM" calendar 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 "VALARM" calendar components 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 "VALARM" calendar components 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 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" calendar component [RFC9073] against certain components e.g. the "VALARM" calendar component, 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).

7.6. Automated Status Changes

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

8. Modifications to Calendar Components

The following changes to the syntax defined in iCalendar [RFC5545] and Event Publishing Extensions to iCalendar [RFC9073] 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 *statusc
                "END" ":" "VEVENT" CRLF

   ; Addition of VSTATUS as a valid component for VTODO
   todoc      = "BEGIN" ":" "VTODO" CRLF
                todoprop *alarmc *participantc *locationc
                    *resourcec *statusc
                "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 properties PERCENT-COMPLETE and REASON to PARTICIPANT
   partprop =/ percent / ; OPTIONAL but MUST NOT occur more than once.
               reason    ; OPTIONAL but MUST NOT occur more than once.
```

9. New Parameter Values

9.1. 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" calendar components.

Format Definition: This property parameter is redefined by the following notation which adds the value "FAILED":

```
partstat-todo    = ("NEEDS-ACTION"    ; To-do needs action
                 / "ACCEPTED"         ; To-do accepted
                 / "DECLINED"         ; To-do declined
                 / "TENTATIVE"        ; To-do tentatively
                                      ; accepted
                 / "DELEGATED"        ; To-do delegated
                 / "COMPLETED"        ; To-do completed
                                      ; COMPLETED property has
                                      ; DATE-TIME completed
                 / "IN-PROCESS"       ; To-do in process of
                                      ; being completed
                 / "FAILED"           ; To-do cannot be
                                      ; completed
                 / x-name             ; Experimental status
                 / iana-token)        ; Other IANA-registered
                                      ; status
; These are the participation statuses for a "VTODO".
; Default is NEEDS-ACTION.
```

Example:

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

10. New Properties

10.1. 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" or "DTSTART" and "DURATION" properties represent the window in which the task can be performed. The "ESTIMATED-DURATION" property SHOULD be passed from the organizer to the "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
```

10.2. 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
```

10.3. Substate

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 substate property allows additional qualification and granularity of states to be recorded, in particular for the IN-PROCESS state. It allows individual substates to be recorded without the need to define and publish a subtask 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
```

10.4. 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 once in a "VTODO" calendar component.

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

```
task-mode   = "TASK-MODE taskmodeparam ":"
                ("AUTOMATIC-COMPLETION"
                / "AUTOMATIC-FAILURE"
                / "AUTOMATIC"
                / "SERVER"
                / "CLIENT"
                / iana-token) CRLF

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 12.1.

The property value is an IANA registered token that defines the mode to be used for the task. The modes are described in the following subsections.

If the "TASK-MODE" property is absent then the "CLIENT" value is assumed.

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

Failing the task MUST be handled by a client.

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

1. the PARTSTAT of one "ATTENDEE" property 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. The effective due date is either the "DUE" property value or the combination of the "DTSTART" and "DURATION" property values.

   Completing the task MUST be handled by a client.

AUTOMATIC Task Mode: This mode handles the automatic behavior of both "AUTOMATIC-COMPLETION" and "AUTOMATIC-FAILURE".

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 SHOULD change the "VTODO" calendar 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.

Examples:

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

11. Property Extensions and Clarifications

11.1. Updated DURATION Property definition for VTODO

[RFC5545] section 3.6.2 introduced a constraint on the use of the "DURATION" property in the "VTODO" calendar component, requiring that a "DURATION" property MUST be accompanied by a "DTSTART" property. 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" calendar component 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.

11.2. 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" calendar components (See Appendix A for examples of the task state lifecycle).

Format Definition: The "STATUS" property parameter list for tasks is redefined by the addition of the values "PENDING" and "FAILED":

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

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
```

12. New Components

12.1. 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
```

13. 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 MUST assume the server does not support this specification. The client MAY attempt to store iCalendar data containing "TASK-MODE" elements but needs to be prepared for a failure response from the server.

A server supporting this specification MUST 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.

13.1. 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>
```

14. Security Considerations

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

15. IANA Considerations

15.1. Component Registrations

This document defines the following new iCalendar component to be added to the Components registry defined in Section 8.3.1 of [RFC5545] and located here: <https://www.iana.org/assignments/icalendar#components>.

Component	Status	Reference
VSTATUS	Current	This Spec, Section 12.1

Table 1: Addition to the Components Registry

15.2. Property Registrations

This document defines the following new iCalendar properties to be added to the Properties registry defined in Section 8.3.2 of [RFC5545] and located here: <https://www.iana.org/assignments/icalendar>.

Property	Status	Reference
ESTIMATED-DURATION	Current	This Spec, Section 10.1
REASON	Current	This Spec, Section 10.2
SUBSTATE	Current	This Spec, Section 10.3
STATUS	Current	This Spec, Section 11.2
TASK-MODE	Current	This Spec, Section 10.4

Table 2: Additions to the Properties Registry

15.3. Initialization of the Status Value registry

This document creates a new iCalendar registry for values of the "STATUS" property defined in Section 3.8.1.11 of [RFC5545] and located here: <https://www.iana.org/assignments/icalendar#status-values>

Additional values MAY be used, provided the process described in Section 8.2.1 of [RFC5545] is used to register them, using the template in Section 8.2.6 of [RFC5545].

The following table has been used to initialize the Status Value Registry.

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]
FAILED	Current	This Spec, Section 11.2
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]
PENDING	Current	This Spec, Section 11.2
TENTATIVE	Current	Section 3.8.1.11 of [RFC5545]

Table 3: Initial Status Value Registry

15.4. Substate Value registry

This document creates a new iCalendar registry for values of the "SUBSTATE" property defined in Section 10.3 and located here: <https://www.iana.org/assignments/icalendar#substate-values>

Additional values MAY be used, provided the process described in Section 8.2.1 of [RFC5545] is used to register them, using the template in Section 8.2.6 of [RFC5545].

The following table has been used to initialize the Substate Value Registry.

Substate	Status	Reference
OK	Current	This Spec, Section 10.3
ERROR	Current	This Spec, Section 10.3
SUSPENDED	Current	This Spec, Section 10.3

Table 4: Initial Substate Value registry

15.5. Task Mode Value registry

This document creates a new iCalendar registry for values of the "TASK-MODE" property defined in Section 10.4 and located here: <https://www.iana.org/assignments/icalendar#task-mode-values>

Additional values MAY be used, provided the process described in Section 8.2.1 of [RFC5545] is used to register them, using the template in Section 8.2.6 of [RFC5545].

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

Task Mode	Status	Reference
AUTOMATIC-COMPLETION	Current	This Spec, Section 10.4
AUTOMATIC-FAILURE	Current	This Spec, Section 10.4
AUTOMATIC	Current	This Spec, Section 10.4
CLIENT	Current	This Spec, Section 10.4
SERVER	Current	This Spec, Section 10.4

Table 5: Task Mode Value Registry

15.6. Participation Statuses registry

This document defines the following new iCalendar participation status to be added to the registry defined in Section 8.3.7 of [RFC5545] and located here: <https://www.iana.org/assignments/icalendar#participation-statuses>

Value	Status	Reference
FAILED	Current	This Spec, Section 9.1

Table 6: Participation Statuses Registry

16. 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.

17. 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>.

18. 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, <https://www.calconnect.org/architectures/Task%20Architecture%201.0.pdf>.

[WsCalendar] Considine, T. and M. Douglass, "WS-Calendar Version 1.0", WsCalendar, 2011, <https://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf>.

[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/>.

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	Overall state set - either by Organizer for TASK-MODE=CLIENT or by server for AUTOMATIC-COMPLETION

Table 7: 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	Overall state set once both attendees are finished. May be set by Organizer for TASK-MODE=CLIENT or by server for AUTOMATIC-COMPLETION

Table 8: Example for multiple Attendees

```
      |  NOTE: The logic for determining the status change to the
      |  "VTODO" calendar component is determined by the task organizer
      |  based on the "ATTENDEE" property 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	Overall state set by Organizer for TASK-MODE=CLIENT or by server for AUTOMATIC-FAILURE

Table 9: 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