CalConnect TC

Calendaring — Task Extensions to iCalendar

Standard Standard

Warning for drafts

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

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

The Calendaring and Scheduling Consortium, Inc. 2024

The Calendaring and Scheduling Consortium, Inc.

4390 Chaffin Lane
McKinleyville
California 95519
United States of America

copyright@calconnect.org
www.calconnect.org

Contents

Abstract

The Internet Calendaring and Scheduling Core Object Specification (iCalendar) (RFC5545) VTODO calendar component has been seen as the poor relation of VEVENT — useful only 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.

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

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

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

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

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

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

This document was prepared by Technical Committee *CALENDAR*.

Introduction

This document specifies extensions to the existing Internet Calendaring and Scheduling Core Object Specification (iCalendar) IETF RFC 5545, 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 IETF RFC 5545 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. Normative references

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

IETF RFC 2119, S. BRADNER. *Key words for use in RFCs to Indicate Requirement Levels*. In: BCP. 1997. RFC Publisher. https://www.rfc-editor.org/info/rfc2119. [viewed: December 8, 2024].

IETF RFC 4791, C. DABOO, B. DESRUISSEAUX and L. DUSSEAULT. *Calendaring Extensions to WebDAV (CalDAV)*. In: RFC. 2007. RFC Publisher. https://www.rfc-editor.org/info/rfc4791. [viewed: December 8, 2024].

IETF RFC 4918, L. DUSSEAULT (ed.). *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*. In: RFC. 2007. RFC Publisher. https://www.rfc-editor.org/info/rfc4918. [viewed: December 8, 2024].

IETF RFC 5545, B. DESRUISSEAUX (ed.). *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. In: RFC. 2009. RFC Publisher. https://www.rfc-editor.org/info/rfc5545. [viewed: December 8, 2024].

IETF RFC 5546, C. DABOO (ed.). *iCalendar Transport-Independent Interoperability Protocol (iTIP)*. In: RFC. 2009. RFC Publisher. https://www.rfc-editor.org/info/rfc5546. [viewed: December 8, 2024].

IETF RFC 8174, B. LEIBA. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. In: BCP. 2017. RFC Publisher. https://www.rfc-editor.org/info/rfc8174. [viewed: December 8, 2024].

IETF RFC 9073, M. DOUGLASS. *Event Publishing Extensions to iCalendar*. In: RFC. 2021. RFC Publisher. https://www.rfc-editor.org/info/rfc9073. [viewed: December 8, 2024].

IETF RFC 9074, C. DABOO. *“VALARM” Extensions for iCalendar*. In: RFC. 2021. RFC Publisher. https://www.rfc-editor.org/info/rfc9074. [viewed: December 8, 2024].

IETF RFC 9253, M. DOUGLASS. *Support for iCalendar Relationships*. In: RFC. 2022. RFC Publisher. https://www.rfc-editor.org/info/rfc9253. [viewed: December 8, 2024].

IETF draft-ietf-calext-subscription-upgrade, [NO INFORMATION AVAILABLE]

Terms defined and used in this specification include:

Assignee
A calendar user assigned to perform a given task. An assignee is equivalent to an “A”ttendee” 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

) Software with which the calendar user communicates with a calendar service or local calendar store to access calendar information.
) 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” IETF RFC 5545.

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. Examples of such resources are locations such as a booakble 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 IETF RFC 5545, these are “VTODO” calendar components, which are groupings of component properties and possibly “VALARM” calendar components that represent an action-item or assignment.

Calendaring — Task Extensions to iCalendar

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 (CalConnect Task Architecture V1.0.

```
Task +-------+ Trigger | +---------------------V---------------------+
+-----------+ | Task Generating System | | | | +-------------------------
+ | | | | | O | | | | | | /|\ | | | | | | / \ | | | | | | Task Organizer
| <----> | | +-^--------^--------------+ | | | | | | | | | | +--------
V-+ +----V-----+ +----------+ | | | | | Task | | Process | | Task | |
| | | |Assignment| | Logic <----> Domain | | | | | | Rules | | | | Data
| | | | | +-----------+ +-----------+ +-----------+ | | | | | | | +------
^----------+-----^-------------------+ | | | | | | | Availability Task Task
| | | | Status | | | | | | | +------v----------v-----+-------------------
+ | | | Calendar and Scheduling System | | Directory | | +---------
+ +---------+ | | Service | | | | | Task | <----> | | |Schedule | |
Lists | | | | | | | | | | | | | +---------+ +---------+ Server | |
| +-------------------------------------------+ | | | Client | | | |
+----------------------+ +-----------+ | | | | | Calendar | | Task | |
| | | | User Agent +----> Specific | <----> | | | | |Application| | | |
| +----------------------+ +-----------+ | | | | | | | +-----^---------
^--------+---------+--------+ | | | | | | | | +-----V---------V--------
V---------V--------+ | | | Task Actors | | | | O O O O | | | | /|\ /|\ /|
\ /|\ +----> | | / \ / \ / \ / \ | | | | Candidate(s) Observer(s) | | | |
Assignee(s) Resource(s) | | | +-------------------------------------------+
+-----------+
```

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 OMG BPMN 2.0.2).

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) IETF RFC 5545 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) IETF RFC 5546 for task assignment and scheduling, and Calendaring Extensions to WebDAV (CalDAV) IETF RFC 4791 for client server communication.

Additionally, this specification uses definitions from Support for iCalendar Relationships IETF RFC 9253. 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.

5. Task Extensions

In order to support the task architecture described in Clause 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 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.

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 IETF RFC 5545 “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.

6.1. Task type

The IETF RFC 9253 “CONCEPT” property is used to identify the type of task, for example;

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

Figure 2

6.2. Task Context and Relationships

The IETF RFC 9253 “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
```

Figure 3

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

The IETF RFC 9253 “REFID” property is used to identify a key used to group tasks by that key.

```
REFID:Manhattan

REFID:1234567890
```

Figure 4

Extensions to the “RELATED-TO” property defined in IETF RFC 9253 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.

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
```

Figure 5

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
```

Figure 6

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. 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, IETF RFC 5545 provides no way to indicate how long the task is expected to take. This document defines a new “ESTIMATED-DURATION” property, in Clause 11.1, 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.

7.2. Milestones

A task that has intermediary deadlines (i.e., milestones) SHOULD 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.

8. Task Scheduling and Assignment

Tasks are assigned to actors using one or more IETF RFC 5545 “ATTENDEE” properties and/or one or more IETF RFC 9073 “PARTICIPANT” calendar 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 “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.

9. Status Reporting

9.1. Improved granularity in status reporting information

This document defines a new “VSTATUS” calendar component (see section Clause 13.1) 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
```

Figure 7

9.2. 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
```

Figure 8

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

9.3. Relating reason and comments to “ATTENDEE” property status changes.

The IETF RFC 9073 “PARTICIPANT” calendar component can be used to provide additional information about why an “ATTENDEE” property 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

```
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
```

Figure 9

9.4. 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 IETF RFC 9074) 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 IETF RFC 9073 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).

9.5. Automated Status Changes

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

10. New Parameter Values

10.1. Redefined VTODO Participant Status

Participant status parameter type values are defined in IETF RFC 5545, Section 3.2.12. This specification redefines that type to include the new value FAILED for “VTODO” calendar components.

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

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

Figure 10

Example

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

Figure 11

11. New Properties

11.1. 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)
```

Figure 12

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” 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 IETF RFC 5546 messages.

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

```
ESTIMATED-DURATION:PT1H
```

Figure 13

11.2. 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)
```

Figure 14

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

:2024

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

Figure 15

11.3. 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
```

Figure 16

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
```

Figure 17

11.4. 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
                 / x-name) CRLF

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

Figure 18

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 Clause 13.1.

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 can 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 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 can 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
```

Figure 19

12. Property Extensions and Clarifications

12.1. Updated DURATION Property definition for VTODO

IETF RFC 5545 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'.
```

Figure 20

is replaced by

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

Figure 21

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.
```

Figure 22

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.
```

Figure 23

12.2. Redefined STATUS Property

The Status property is defined in IETF RFC 5545, Section 3.8.1.11. 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 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" calendar component.
```

Figure 24

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
```

Figure 25

13. New Components

13.1. 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 IETF RFC 9073 “PARTICIPANT” component to allow participants in a task to specify their own status.

For backwards compatibility, when a VSTATUS component is added the IETF RFC 5545 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
               ;
               )
```

Figure 26

Examples

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

Figure 27

14. CalDAV Support for Task Mode

The CalDAV IETF RFC 4791 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.

14.1. 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 IETF RFC 4918).

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 -->
```

Figure 28

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>
```

Figure 29

15. Security Considerations

This specification introduces no new security considerations beyond those identified in IETF RFC 5545, IETF RFC 5546 and IETF RFC 4791.

16. IANA Considerations

16.1. New and updated iCalendar Elements Registration

This specification updates IETF RFC 5545 by adding and updating a number of elements according to the procedures and templates specified in IETF RFC 5545, Section 8.2.

16.1.1. Update of the Status registry

This specification updates the Status registry defined in IETF draft-ietf-calext-subscription-upgrade with additional values defined in this document.

Table 1 — Updated Status Value Registry

Value	Status	Reference
PENDING	Current	This Spec, Clause 12.2
FAILED	Current	This Spec, Clause 12.2

16.1.2. Sub-State value registry

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

Table 2 — Sub-State registry

Substate	Status	Reference
OK	Current	This Spec, Clause 11.3
ERROR	Current	This Spec, Clause 11.3
SUSPENDED	Current	This Spec, Clause 11.3

16.1.3. Task Mode value registry

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

Table 3 — Task Mode Value Registry

Task Mode	Status	Reference
AUTOMATIC-COMPLETION	Current	This Spec, Clause 11.4
AUTOMATIC-FAILURE	Current	This Spec, Clause 11.4
AUTOMATIC	Current	This Spec, Clause 11.4
CLIENT	Current	This Spec, Clause 11.4
SERVER	Current	This Spec, Clause 11.4

16.1.4. Participation Statuses registry

The following table has been used to update the Participation Statuses registry defined in IETF RFC 5545, Section 8.3.7 and located here: <https://www.iana.org/assignments/icalendar>

Table 4 — Participation Statuses Registry

Value	Status	Reference
FAILED	Current	This Spec, Clause 10.1

16.1.5. Components Registry

The following table has been used to update the Components registry defined in IETF RFC 5545, Section 8.3.1 and located here: <https://www.iana.org/assignments/icalendar>.

Table 5 — Updated Components Registry

Component	Status	Reference
VSTATUS	Current	This Spec, Clause 13.1

16.1.6. Properties registry

The following table has been used to update the Properties registry defined in IETF RFC 5545, Section 8.3.2 and located here: <https://www.iana.org/assignments/icalendar>.

Table 6 — Updated Properties Registry

Property	Status	Reference
ESTIMATED-DURATION	Current	This Spec, Clause 11.1
REASON	Current	This Spec, Clause 11.2
SUBSTATE	Current	This Spec, Clause 11.3
STATUS	Current	This Spec, Clause 12.2
TASK-MODE	Current	This Spec, Clause 11.4

17. 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.

Appendix A
(informative)
Examples of Task State Lifecycle

A.1. Simple Case Status Change

Table A.1 — Example of status changes in assigning and performing a task with one attendee.

	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

A.2. Example for multiple Attendees

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

Table A.2 — Example for multiple Attendees

	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.

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.

Table A.3 — Example of Failure

	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

Appendix B (informative) Change log

V02. 2021-05-05 MD

- Redo in asciidoc
- Change STRUCTURED-CATEGORY to CONCEPT
- Add GROUP parameter definition

V01. 2015-08-23 AA

- Highlighted use of ESTIMATED-DURATION for time planning.
- Corrected PARTSTAT example section 5.1. Changed DECLINED to FAILED.
- Replaced Task Mode AUTOMATIC-STATUS with CLIENT and SERVER modes. Also, clarified that task mode processing is only done on the organizer’s copy.
- Clarified responsibility for setting MODIFIED.
- CalDAV support added.
- Updated normative references.

Appendix C
(informative)
Working Notes

C.1. Subscribing to task updates

Stakeholders should have the ability to subscribe to categories / types of tasks on an ongoing basis. Reference calendarserver.org notifications draft

Bibliography

[1] OMG BPMN 2.0.2, *Business Process Model and Notation*. 2014. https://www.omg.org/spec/BPMN/2.0.2/About-BPMN. [viewed: December 8, 2024].

[2] (CalConnect Task Architecture V1.0, APTHORP, A., C. DABOO and M. DOUGLASS. *CalConnect, Task Architecture V1.0*. n.p.: n.d. (CalConnect Task Architecture V1.0. https://www.calconnect.org/architectures/Task%20Architecture%201.0.pdf. [viewed: December 8, 2024].

[3] WfRP, RUSSELL, N., A.H.M. TER HOFSTEDE, T. EDMOND and W.M.P. VAN DER AALST. *Workflow Resource Patterns*. n.p.: Eindhoven University of Technology. 2004. WfRP. http://www.workflowpatterns.com/patterns/resource/. [viewed: December 8, 2024].