Network Working Group
Internet Draft
<draft-desruisseaux-caldav-sched-07>
Intended status: Standards Track
Expires: December 2009

C. Daboo
Apple Inc.
B. Desruisseaux
Oracle
June 2009

CalDAV Scheduling Extensions to WebDAV
draft-desruisseaux-caldav-sched-07

Status of this Memo

CONFORMANCE UNDEFINED.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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”.

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire in December 2009.

Copyright Notice

Abstract

This document defines extensions to the CalDAV "calendar-access" feature to specify a standard way of performing scheduling transactions with iCalendar-based calendar components. This document defines the "calendar-auto-schedule" feature of CalDAV.

Table of Contents

1. Introduction

This document specifies extensions to the CalDAV "calendar-access" [RFC4791] feature to enable scheduling of iCalendar-based [I-D.ietf-calsify-rfc2445bis] calendar components between Calendar Users. This extension leverages the scheduling methods defined in the iCalendar Transport-independent Interoperability Protocol (iTIP) [I-D.ietf-calsify-2446bis] to permit Calendar Users to perform scheduling transactions such as schedule, reschedule, respond to scheduling request or cancel scheduled calendar components, as well as search for busy time information.

Discussion of this Internet-Draft is taking place on the mailing list <https://www.ietf.org/mailman/listinfo/caldav>.

1.1 Terminology

This specification uses much of the same terminology as iCalendar [I-D.ietf-calsify-rfc2445bis], iTIP [I-D.ietf-calsify-2446bis], WebDAV [RFC4918], and CalDAV [RFC4791]. The following definitions are provided to aid the reader in understanding this specification.

Calendar User (CU): An entity (often a human) that accesses calendar information [RFC3283].

Calendar User Agent (CUA): Software with which the calendar user communicates with a calendar service or local calendar store to access calendar information [RFC3283].

Calendar collection: A resource that acts as a container of references to child calendar object resources [RFC4791].

Calendar object resource: A resource representing a calendar object (event, to-do, journal entry, or other calendar components) [RFC4791].

Scheduling object resource: A calendar object resource contained in a calendar collection for which the server will take care of sending scheduling messages on behalf of the owner of the calendar collection.

Organizer scheduling object resource: A scheduling object resource owned by an Organizer.

Attendee scheduling object resource: A scheduling object resource owned by an Attendee.

Automatic scheduling transaction: Add, change or remove operations on a scheduling object resource for which the server will deliver scheduling messages to other Calendar Users.

Scheduling message: A calendar object resource that describes a scheduling transaction such as schedule, reschedule, reply, or cancel.

Scheduling Outbox collection: A resource at which busy time information requests are targeted.

Scheduling Inbox collection: A collection in which incoming scheduling messages are delivered.

1.2 Approach

iTIP [I-D.ietf-calsify-2446bis] outlines a model where Calendar Users exchange scheduling messages with one another. Often times, Calendar User Agents are made responsible for generating and sending scheduling messages as well as processing incoming scheduling messages. This approach yields a number of problems, including:

- For most updates to a scheduled calendar component, Calendar User Agents need to address a separate scheduling messages to the Organizer or the Attendees.
- The handling of incoming scheduling messages and the updates to calendars impacted by those messages only occurs when Calendar User Agents are active.
- Due to the update latency, it is possible for calendars of different Calendar Users to reflect different, inaccurate states.

This specification uses an alternative approach where the server is made responsible for sending scheduling messages and processing incoming scheduling messages. This approach frees the Calendar User Agents from the submission and processing of scheduling messages and ensures better consistency of calendar data across users' calendars. The operation of creating, modifying or deleting a scheduled calendar component in a calendar is enough to trigger the server to deliver the necessary scheduling messages to the appropriate Calendar Users.

1.3 Limitations

While the scheduling features described in this specification are based on iTIP [I-D.ietf-calsify-2446bis], some of its more complex features have deliberately been left out in order to keep this specification simple. In particular, the following iTIP [I-D.ietf-calsify-2446bis] features are not covered: publishing, countering, delegating, refreshing and forwarding calendar components, as well as replacing the Organizer of a calendar component.

The goal of this specification is to provide the essential scheduling features needed. It is expected that future extensions will be developed to address the more complex features.

1.4 Notational Conventions

The Augmented BNF (ABNF) syntax used by this document to specify the format definition of new iCalendar elements is defined in [RFC5234].

The Augmented BNF (ABNF) syntax used by this document to specify the format definition of new message header fields to be used with the HTTP/1.1 protocol is described in Section 2.1 of [RFC2616]. Since this Augmented BNF uses the basic production rules provided in Section 2.2 of [RFC2616], these rules apply to this document as well.

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

When XML element types in the namespaces "DAV:" and "urn:ietf:params:xml:ns:caldav" are referenced in this document outside of the context of an XML fragment, the string "DAV:" and "CALDAV:" will be prefixed to the element types respectively.

1.5 XML Namespaces and Processing

Definitions of XML elements in this document use XML element type declarations (as found in XML Document Type Declarations), described in Section 3.2 of [W3C.REC-xml-20081126].

The XML elements specified in this document are defined in the "urn:ietf:params:xml:ns:caldav" XML namespace registered by CalDAV [RFC4791].

The XML declarations used in this document do not include namespace information. Thus, implementers must not use these declarations as the only way to create valid CalDAV properties or to validate CalDAV XML element type. Some of the declarations refer to XML elements defined by WebDAV [RFC4918] which use the "DAV:" namespace. Wherever such XML elements appear, they are explicitly prefixed with "DAV:" to avoid confusion. Additionally, some of the elements used here are defined in CalDAV "calendar-access" [RFC4791].

Also note that some CalDAV XML element names are identical to WebDAV XML element names, though their namespace differs. Care must be taken not to confuse the two sets of names.

Processing of XML by CalDAV clients and servers MUST follow the rules described in Section 17 of [RFC4918].

2. Scheduling Process

The process of scheduling an event between different parties often involves a series of steps with different actors playing particular roles during the whole process. Typically there is an event "Organizer" whose role is to setup an event between one or more "Attendees", and this is done by sending out invitations and handling responses from each Attendee.

This process can typically be broken down into two phases.

In the first phase, the Organizer will query the busy time information of each Attendee to determine the most appropriate time for the event. This request is sometimes called a freebusy lookup.

In the second phase, the Organizer sends out invitations to each Attendee using the time previously determined from the freebusy lookup. There then follows exchanges between Organizer and Attendees regarding the invitation. Some Attendees may choose to attend at the time proposed by the Organizer, others may decline to attend. The Organizer needs to process each of the replies from the Attendees and take appropriate action to confirm the event, reschedule it or perhaps cancel it.

The user expectation as to how a calendaring and scheduling system should respond in each of these two phases is somewhat different. In the case of a freebusy lookup, users expect to get back results immediately so that they can then move on to the invitation phase as quickly as possible. In the case of invitations, it is expected that each Attendee will reply with their participation status in their own time, so delays in receiving replies are anticipated. Thus calendaring and scheduling systems should treat these two operational phases in different ways to accommodate the user expectations, and this specification does that.

While the scenario described above only covers the case of scheduling events between Calendar Users, and requesting busy time information, this specification also provides support for the scheduling of to-dos between Calendar Users. For the majority of the following discussion, scheduling of events and freebusy lookups will be discussed, as these are the more common operations.

3. Scheduling Support

A server that supports the features described in this document MUST include "calendar-auto-schedule" as a field in the DAV response header from an OPTIONS request on any resource that supports any scheduling actions, properties, privileges or methods.

To advertise support for the CalDAV "calendar-auto-schedule" feature a server is REQUIRED to support and advertise support for the CalDAV "calendar-access" [RFC4791] feature.

>> Request <<

```
OPTIONS /home/cyrus/calendars/inbox/ HTTP/1.1
Host: cal.example.com
```

>> Response <<

```
HTTP/1.1 204 No Content
Date: Thu, 31 Mar 2005 09:00:00 GMT
Allow: OPTIONS, GET, HEAD, DELETE, TRACE, PROPFIND
Allow: PROPPATCH, LOCK, UNLOCK, REPORT, ACL
DAV: 1, 2, 3, access-control
DAV: calendar-access, calendar-auto-schedule
```

In this example, the OPTIONS response indicates that the server supports the "calendar-access" and "calendar-auto-schedule" features and that resource "/home/cyrus/calendars/inbox/" supports the scheduling actions, properties, privileges and methods defined in this specification.

4. Scheduling Collections

This specification introduces new collection resource types that are used to manage scheduling object resources, scheduling privileges as well as provide scheduling functionality.

4.1 Scheduling Outbox Collection

A scheduling Outbox collection is used as the target for busy time information requests.

A scheduling Outbox collection MUST report the DAV:collection and CALDAV:schedule-outbox XML elements in the value of the DAV:resourcetype property. The element type declaration for CALDAV:schedule-outbox is:

```
<!ELEMENT schedule-outbox EMPTY>
```

Example:

```
<D:resourcetype xmlns:D="DAV:">
  <D:collection/>
  <C:schedule-outbox xmlns:C="urn:ietf:params:xml:ns:caldav"/>
</D:resourcetype>
```

New WebDAV ACL [RFC3744] privileges can be set on the scheduling Outbox collection to control who is allowed to send scheduling messages on behalf of the Calendar User associated with the scheduling Outbox collection. See Section 13.1 for more details.

A scheduling Outbox collection MUST NOT be a child (at any depth) of a calendar collection resource.

The following WebDAV properties specified in CalDAV "calendar-access" [RFC4791] MAY also be defined on scheduling Outbox collections:

CALDAV:supported-calendar-component-set - when present this indicates the allowed calendar component types for scheduling messages submitted to the scheduling Outbox collection with the POST method.

CALDAV:supported-calendar-data - when present this indicates the allowed media types for scheduling messages submitted to the scheduling Outbox collection with the POST method.

CALDAV:max-resource-size - when present this indicates the maximum size of a resource in octets that the server is willing to accept for scheduling messages submitted to the scheduling Outbox collection with the POST method.

CALDAV:min-date-time - when present this indicates the earliest date and time (in UTC) that the server is willing to accept for any DATE or DATE-TIME value in scheduling messages submitted to the scheduling Outbox collection with the POST method.

CALDAV:max-date-time - when present this indicates the latest date and time (in UTC) that the server is willing to accept for any DATE or DATE-TIME value in scheduling messages submitted to the scheduling Outbox collection with the POST method.

CALDAV:max-instances - when present this indicates the maximum number of recurrence instances in scheduling messages submitted to the scheduling Outbox collection with the POST method.

CALDAV:max-attendees-per-instance - when present this indicates the maximum number of ATTENDEE properties in any instance of scheduling messages submitted to the scheduling Outbox collection with the POST method.

While there is currently no defined use for child resources in a scheduling Outbox collection, a scheduling Outbox collection MAY contain child resources.

4.2 Scheduling Inbox Collection

A scheduling Inbox collection contains copies of incoming scheduling messages. These may be requests sent by an Organizer, or replies sent by an Attendee in response to a request.

A scheduling Inbox collection MUST report the DAV:collection and CALDAV:schedule-inbox XML elements in the value of the DAV:resourcetype property. The element type declaration for CALDAV:schedule-inbox is:

```
<!ELEMENT schedule-inbox EMPTY>
```

Example:

```
<D:resourcetype xmlns:D="DAV:">
  <D:collection/>
  <C:schedule-inbox xmlns:C="urn:ietf:params:xml:ns:caldav"/>
</D:resourcetype>
```

Scheduling Inbox collections MUST only contain calendar object resources that obey the restrictions specified in iTIP [I-D.ietf-calsify-2446bis]. Consequently, scheduling Inbox collections MUST NOT contain any types of collection resources. Restrictions defined in Section 4.1 of CalDAV "calendar-access" [RFC4791] on calendar object resources contained in calendar collections (e.g., "UID" uniqueness) don't apply to calendar object resources contained in a scheduling Inbox collection. Multiple calendar object resources contained in a scheduling Inbox collection MAY have the same "UID" property value (i.e., multiple scheduling messages for the same calendar component).

New WebDAV ACL [RFC3744] privileges can be set on the scheduling Inbox collection to control who the Calendar User associated with the scheduling Inbox collection will accept scheduling messages from. See Section 13.1 for more details.

A scheduling Inbox collection MUST NOT be a child (at any depth) of a calendar collection resource.

The following WebDAV properties specified in CalDAV "calendar-access" [RFC4791] MAY also be defined on scheduling Inbox collections:

CALDAV:calendar-timezone - when present this contains a time zone that the server can use when calendar date-time operations are carried out, for example when a time-range CALDAV:calendar-query REPORT is targeted at a scheduling Inbox collection.

CALDAV:supported-calendar-component-set - when present this indicates the allowed calendar component types for scheduling messages delivered to the scheduling Inbox collection.

CALDAV:supported-calendar-data - when present this indicates the allowed media types for scheduling messages delivered to the scheduling Inbox collection.

CALDAV:max-resource-size - when present this indicates the maximum size of a resource in octets that the server is willing to accept for scheduling messages delivered to the scheduling Inbox collection.

CALDAV:min-date-time - when present this indicates the earliest date and time (in UTC) that the server is willing to accept for any DATE or DATE-TIME value in scheduling messages delivered to the scheduling Inbox collection.

CALDAV:max-date-time - when present this indicates the latest date and time (in UTC) that the server is willing to accept for any DATE or DATE-TIME value in scheduling messages delivered to the scheduling Inbox collection.

CALDAV:max-instances - when present this indicates the maximum number of recurrence instances in scheduling messages delivered to the scheduling Inbox collection.

CALDAV:max-attendees-per-instance - when present this indicates the maximum number of ATTENDEE properties in any instance of scheduling messages delivered to the scheduling Inbox collection.

4.3 Calendaring Reports Extensions

This specification extends the CALDAV:calendar-query and CALDAV:calendar-multiget REPORTs to return results for calendar object resources in scheduling Inbox collections when the report directly targets such a collection. That is, the Request-URI for a report MUST be the URI of the scheduling Inbox collection or of a child resource within a scheduling Inbox collection. A report run on a regular collection that includes a scheduling Inbox collection as a child resource at any depth MUST NOT examine or return any calendar object resources from within any scheduling Inbox collections.

When a CALDAV:calendar-query REPORT includes a time-range query and targets a scheduling Inbox collection, if any calendar object resources contain "VEVENT" calendar components that do not include a "DTSTART" iCalendar property (as allowed by iTIP [I-D.ietf-calsify-2446bis]) then such components MUST always match the time-range query test.

Note that the CALDAV:free-busy-query REPORT is not supported on scheduling Inbox collections.

5. Scheduling Transactions

When a calendar object resource is created, modified or removed from a calendar collection (either via a PUT, DELETE, COPY or MOVE HTTP request), the server examines the calendar data and checks to see whether the data represents a scheduling object resource. If it does, the server will automatically attempt to deliver a scheduling message to the appropriate Calendar Users. Several types of scheduling operation can occur in this case, equivalent to iTIP "REQUEST", "REPLY", "CANCEL", and "ADD" operations.

5.1 Identifying Scheduling Object Resources

Calendar object resources on which the server performs automatic scheduling transactons are refered to as scheduling object resources. There are two types of scheduling object resources: organizer scheduling object resources, and attendee scheduling object resources.

A calendar object resource is considered to be a valid organizer scheduling object resource if the "ORGANIZER" iCalendar property is present and set in all the calendar components to a value that matches one of the calendar user addresses of the owner of the calendar collection.

A calendar object resource is considered to be a valid attendee scheduling object resource if the "ORGANIZER" iCalendar property is present and set in all the calendar components to the same value and doesn't match one of the calendar user addresses of the owner of the calendar collection, and that at least one of the "ATTENDEE" iCalendar property values match one of the calendar user addresses of the owner of the calendar collection.

The creation of attendee scheduling object resources is typically done by the server, with the resource being stored in an appropriate calendar collection.

5.2 Handling Scheduling Object Resources

The server's behavior when processing a scheduling object resource depends on whether it is owned by the Organizer or an Attendee specified in the calendar data.

5.2.1 Organizer Scheduling Object Resources

An Organizer can create, modify or remove a scheduling object resource by issuing HTTP requests with an appropriate method. The create, modify and remove behaviors for the server are each described next, and the way these are invoked via HTTP requests is described in Section 5.2.3.

5.2.1.1 Create

When a scheduling object resource is created by the Organizer, the server will inspect each "ATTENDEE" property to determine if a scheduling message should be delivered to this Attendee according to the value of the "SCHEDULE-AGENT" property parameter (see Section 10.1) as described in the table below:

SCHEDULE-AGENT	iTIP METHOD
SERVER (default)	REQUEST
CLIENT	--
NONE	--

The attempt to deliver the scheduling message will either succeed or fail. In all cases, the server MUST add a "SCHEDULE-STATUS" iCalendar property parameter (see Section 10.3) to the "ATTENDEE" iCalendar property in the scheduling object resource being created, and set its value as described in Section 9.2. This will result in the created calendar object resource differing from the calendar data sent in the HTTP request. As a result clients MAY reload the calendar data from the server as soon as it is created on the server in order to update to the new server generated state information. Servers MUST NOT set the "SCHEDULE-STATUS" property parameter on the "ATTENDEE" property of Attendees for which it did not attempt to deliver a scheduling message.

Restrictions:

1. The server MAY reject any attempt to set the "PARTSTAT" iCalendar property parameter value of the "ATTENDEE" iCalendar property of other users in the calendar object resource to a value other than "NEEDS-ACTION" if the "SCHEDULE-AGENT" property parameter value is not present or set to the value "SERVER". To maintain consistency across Organizers and Attendees, a server will typically choose to enforce the requirement that only an Attendee can change their own "PARTSTAT" to a value other than "NEEDS-ACTION".
2. The server MAY reject attempts to create a scheduling object resource that specifies a "UID" property value already specified in a scheduling object resource contained in another calendar collection of the Organizer.
3. The server MUST take into account scheduling privileges as described in Section 13.1 when handling the creation of a scheduling object resource.
4. Restrictions on calendar object resources defined in Section 4.1 of [RFC4791] MUST also be enforced.

5.2.1.2 Modify

When a scheduling object resource is modified by the Organizer, the server will inspect each "ATTENDEE" property in the new calendar data to determine which ones have the "SCHEDULE-AGENT" iCalendar property parameter. It will then need to compare this with the "ATTENDEE" properties in the existing calendar object resource that is being modified.

For each Attendee in the old and new calendar data on a per-instance basis, and taking into account the addition or removal of Attendees, the server will determine whether to deliver a scheduling message to the Attendee. The following table determines whether the server needs to deliver a scheduling message, and if so which iTIP scheduling method to use. The values "SERVER", "CLIENT", and "NONE" in the top and left titles of the table refer to the "SCHEDULE-AGENT" parameter value of the "ATTENDEE" property, and the values "<Absent>" and "<Removed>" are used to cover the cases where the "ATTENDEE" property is not present (Old) or is being removed (New).

ATTENDEE		New			
		<Removed>	SERVER (default)	CLIENT	NONE
Old	<Absent>	--	REQUEST / ADD	--	--
	SERVER (default)	CANCEL	REQUEST	CANCEL	CANCEL
	CLIENT	--	REQUEST / ADD	--	--
	NONE	--	REQUEST / ADD	--	--

The attempt to deliver the scheduling message will either succeed or fail. In all cases, the server MUST add a "SCHEDULE-STATUS" iCalendar property parameter to the "ATTENDEE" iCalendar property in the scheduling object resource being modified, and set its value as described in Section 9.2. This will result in the created calendar object resource differing from the calendar data sent in the HTTP request. As a result clients MAY reload the calendar data from the server as soon as it is modified on the server in order to update to the new server generated state information.

Restrictions:

1. The server MAY reject any attempt to set the "PARTSTAT" iCalendar property parameter value of the "ATTENDEE" iCalendar property of other users in the calendar object resource to a value other than "NEEDS-ACTION" if the "SCHEDULE-AGENT" property parameter value is not present or set to the value "SERVER". To maintain consistency for Organizers and Attendees, a server will typically choose to enforce the requirement that only an Attendee can change their own "PARTSTAT" to a value other than "NEEDS-ACTION".
2. The server MUST take into account scheduling privileges as described in Section 13.1 when handling the modification of a scheduling object resource.
3. Restrictions on calendar object resources defined in Section 4.1 of [RFC4791] MUST also be enforced.

5.2.1.3 Remove

When a scheduling object resource is removed by the Organizer, the server will inspect each "ATTENDEE" property in the scheduling object resource being removed to determine which ones have the "SCHEDULE-AGENT" iCalendar property parameter.

For each Attendee the server will determine whether to attempt to deliver a scheduling message into the Attendee's scheduling Inbox collection, based on the table below:

SCHEDULE-AGENT	iTIP METHOD
SERVER (default)	CANCEL
CLIENT	--
NONE	--

Restrictions:

1. The server MUST take into account scheduling privileges as described in Section 13.1 when handling the deletion of a scheduling object resource.

5.2.2 Attendee Scheduling Object Resources

An Attendee can create, modify or remove a scheduling object resource by issuing HTTP requests with an appropriate method. The create, modify and remove behaviors for the server are each described next, and the way these are invoked via HTTP requests is described in Section 5.2.3.

5.2.2.1 Allowed Attendee Changes

Attendees are allowed to make some changes to a scheduling object resource, though key properties such as start time, end time, location, and summary are typically under the control of the Organizer.

The server MUST allow Attendees to:

1. change their own "PARTSTAT" iCalendar property parameter value.
2. add, modify or remove any "TRANSP" iCalendar properties.
3. add, modify or remove any "PERCENT-COMPLETE" iCalendar properties.
4. add, modify or remove any "VALARM" iCalendar components.

5. add, modify or remove the "CALSCALE" iCalendar property within the top-level "VCALENDAR" component.
6. modify the "PRODID" iCalendar property within the top-level "VCALENDAR" component.
7. add "EXDATE" iCalendar properties and possibly remove components for overridden recurrence instances.
8. add, modify or remove any "CREATED", "DTSTAMP" and "LAST-MODIFIED" iCalendar properties.
9. add new components to represent overridden recurrence instances, provided the only changes to the recurrence instance follow the rules above.

5.2.2.2 Create

Typically an Attendee does not create scheduling object resources, as scheduling messages delivered to them on the server are automatically processed by the server and placed on one of their calendars (see Section 6). However, in some cases a scheduling message may get delivered directly to the client, and the Attendee may wish to store that on the server. In that case the client creates a scheduling object resource in a suitable calendar belonging to the Attendee. Once stored, it is then subject to the usual rules for attendee scheduling object resources.

In some cases a server may not be able to process an Attendee scheduling object resource that originated from another system (i.e., where the server is unable to deliver scheduling messages to the Organizer). In such cases the server MUST add a "SCHEDULE-AGENT" iCalendar property parameter to all "ORGANIZER" iCalendar properties in the resource and set the value of each to "NONE". The server MAY reject any attempt by the client to remove the "SCHEDULE-AGENT" property parameter or change its value.

5.2.2.3 Modify

When a scheduling object resource is modified by an Attendee, the server will inspect the changes by comparing it with the existing scheduling object resource being replaced.

If the Attendee changes one or more "PARTSTAT" iCalendar property values on any component, or adds an overridden component with a changed "PARTSTAT" property, then the server MUST deliver an iTIP "REPLY" scheduling message to the Organizer to indicate the new participation status of the Attendee.

The attempt to deliver the scheduling message will either succeed or fail. In all cases, the server MUST add a "SCHEDULE-STATUS" iCalendar property parameter to the "ORGANIZER" iCalendar property in the scheduling object resource being created, and set its value as described in Section 9.2. This will result in the created calendar object resource differing from the calendar data sent in the HTTP request. As a result clients MAY reload the calendar data from the server as soon as it is stored in order to update to the new server generated state information.

5.2.2.4 Remove

When a scheduling object resource is removed by the Attendee, one of two possibilities exist:

1. If the HTTP request contains a "Schedule-Reply" request header set to the value "T" or there is no "Schedule-Reply" request header, then the server MUST attempt to deliver a scheduling message to the Organizer indicating that the Attendee has a "PARTSTAT" iCalendar property parameter value set to "DECLINED". That is, the Attendee has chosen not to attend any instances. If the server is unable to deliver the scheduling message, the remove action MUST fail, and an appropriate "SCHEDULE-STATUS" iCalendar property parameter set on the "ORGANIZER" property in the scheduling object resource stored by the server.
2. If the HTTP request contains a request header "Schedule-Reply" set to the value "F", the server MUST NOT attempt to deliver a scheduling message. The resource is simply removed. This provides the client a way to silently remove unwanted scheduling attempts.

5.2.3 HTTP Methods

This section describes how use of various HTTP methods on a scheduling object resource will cause a create, modify or remove action on that resource as described above. The use of these methods is subject to the restrictions in [RFC4791], in addition to what is described below.

5.2.3.1 PUT

When a PUT method request is received, the server will execute the following actions, provided all appropriate preconditions are met:

Existing Destination Resource	Resulting Destination Resource	Server Action
None	Calendar object resource	None
None	Scheduling object resource	Create
Calendar object resource	Calendar object resource	None
Calendar object resource	Scheduling object resource	Create
Scheduling object resource	Calendar object resource	Remove
Scheduling object resource	Scheduling object resource	Modify

5.2.3.2 COPY

When a COPY method request is received, the server will execute the following actions based on the source and destination collections in the request:

Source Collection	Destination Collection	Server Action
Non-calendar collection	Non-calendar collection	None
Non-calendar collection	Calendar collection	(1)
Calendar collection	Non-calendar collection	None
Calendar collection	Calendar collection	None

Note 1. The same rules as used for PUT above are applied for the destination of the COPY request.

5.2.3.3 MOVE

When a MOVE method request is received, the server will execute the following actions based on the source and destination collections in the request:

Source Collection	Destination Collection	Server Action
Non-calendar collection	Non-calendar collection	None
Non-calendar collection	Calendar collection	(1)
Calendar collection	Non-calendar collection	(2)
Calendar collection	Calendar collection	None

Note 1. The same rules as used for PUT above are applied for the destination of the MOVE request.

Note 2. The same rules as used for DELETE below are applied for the source of the MOVE request.

5.2.3.4 DELETE

When a DELETE method is targeted at a scheduling object resource the server will execute the Remove action.

When a DELETE method is targeted at a calendar collection the server will execute the Remove action on all scheduling object resources contained in the calendar collection.

5.2.4 Additional Method Preconditions

This specification defines additional method preconditions (see Section 16 of WebDAV [RFC4918]) to provide machine-parsable information in error responses.

5.2.4.1 CALDAV:unique-scheduling-object-resource Precondition

Name: unique-scheduling-object-resource

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

Apply to: PUT, COPY, and MOVE

Use with: 409 Conflict

Purpose: (precondition) -- Servers MAY reject requests to create a scheduling object resource with an iCalendar "UID" property value already in use by another scheduling object resource owned by the same user in other calendar collections. Servers SHOULD report the URL of the scheduling object resource that is already making use of the same "UID" property value in the DAV:href element.

Definition:

```
   <!ELEMENT unique-scheduling-object-resource (DAV:href?)>
```

Example:

```
   <C:unique-scheduling-object-resource xmlns:D="DAV:"
       xmlns:C="urn:ietf:params:xml:ns:caldav">
<D:href>/home/bernard/calendars/personal/abc123.ics</D:href>
   </C:unique-scheduling-object-resource>
```

5.2.4.2 CALDAV:same-organizer-in-all-components Precondition

Name: same-organizer-in-all-components

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

Apply to: PUT, COPY, and MOVE

Use with: 409 Conflict

Purpose: (precondition) -- All the calendar components in a scheduling object resource MUST contain the same "ORGANIZER" property value when present.

Definition:

```
   <!ELEMENT same-organizer-in-all-components EMPTY>
```

Example:

```
   <C:same-organizer-in-all-components
       xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

5.2.4.3 CALDAV:allowed-organizer-scheduling-object-change Precondition

Name: allowed-organizer-scheduling-object-change

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

Apply to: PUT, COPY, and MOVE

Use with: 409 Conflict

Purpose: (precondition) -- Servers MAY impose restrictions on modifications allowed by an

Organizer. For instance, servers MAY prevent the Organizer setting the "PARTSTAT" property parameter to a value other than "NEEDS-ACTION" if the corresponding "ATTENDEE" property has the "SCHEDULE-AGENT" property parameter set to "SERVER", or has no "SCHEDULE-AGENT" property parameter. See Section 5.2.1.

Definition:

```
<!ELEMENT allowed-organizer-scheduling-object-change EMPTY>
```

Example:

```
<C:allowed-organizer-scheduling-object-change
    xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

5.2.4.4 CALDAV:allowed-attendee-scheduling-object-change Precondition

Name: allowed-attendee-scheduling-object-change

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

Apply to: PUT, COPY, and MOVE

Use with: 409 Conflict

Purpose: (precondition) -- Servers MAY impose restrictions on modifications allowed by an Attendee. Attendee modifications that servers MUST allow are specified in Section 5.2.2.1.

Definition:

```
<!ELEMENT allowed-attendee-scheduling-object-change EMPTY>
```

Example:

```
<C:allowed-attendee-scheduling-object-change
    xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

5.2.5 DTSTAMP and SEQUENCE Properties

Whenever the server generates a scheduling message for delivery to a Calendar User, it MUST ensure that a "DTSTAMP" iCalendar property is present and MUST set the value to the UTC time that the scheduling message was generated (as required by iCalendar).

iTIP [I-D.ietf-calsify-2446bis] places certain requirements on how the "SEQUENCE" iCalendar property value in scheduling messages changes. The server MUST ensure that for each type of scheduling operation, the "SEQUENCE" iCalendar property value is appropriately updated. If the client does not update the "SEQUENCE" iCalendar property itself when that is required, the server MUST update the property.

5.2.6 Limit Recurrence Instances Sent to Attendees

When delivering scheduling messages for recurring calendar components to Attendees, servers MUST ensure that Attendees only get information about recurrence instances that explicitly include them as an Attendee.

For example, if an Attendee is invited to a single recurrence instance of a recurring event, and no others, the scheduling object resource contained in the Organizer's calendar collection will contain an overridden instance in the form of a separate calendar component. That separate calendar component will include the "ATTENDEE" property referencing the "one-off" Attendee. That Attendee will not be listed in any other calendar components in the scheduling object resource. The scheduling message that will be delivered to the Attendee will only contain information about this overridden instance.

As another example, an Attendee could be excluded from one instance of a recurring event. In that case the scheduling object resource contained in the calendar collection of the Organizer will include an overridden instance with an "ATTENDEE" list that does not include the Attendee being excluded. The scheduling message that will be delivered to the Attendee will not specify the overridden instance but rather include an "EXDATE" property in the master recurring component defining the recurrence set.

5.2.7 Forcing the Server to Send a Scheduling Message

The iCalendar property parameter "SCHEDULE-FORCE-SEND" defined in Section 10.2 can be used by a Calendar User to force the server to send a scheduling message to an Attendee or the Organizer in a situation where the server would not normally send a scheduling message. For instance, an Organizer could use this property parameter to request an Attendee, that previously declined an invitation, to reconsider their participation status without being forced to modify the event.

6. Processing Incoming Scheduling Messages

Scheduling operations can cause the delivery of a scheduling message into an Organizer's or Attendee's scheduling Inbox collection. In the former case the scheduling messages are replies from Attendees, in the latter case the scheduling messages are requests, cancellations or additions from the Organizer.

The server will automatically process incoming scheduling messages and make them available in the scheduling Inbox collection as an indicator to the client that a scheduling operation has taken place.

The server MUST take into account privileges on the scheduling Inbox collection, when processing incoming scheduling messages, to determine whether delivery of the scheduling message is allowed. Privileges on calendars containing any matching scheduling object resource are not considered in this case. Additionally, servers MUST take into account any scheduling Inbox collection preconditions (see Section 4.2) when delivering the scheduling message, and it MUST take into account the similar preconditions on any calendar collection which contains, or would contain, the corresponding scheduling object resource.

6.1 Processing Attendee Replies

For a scheduling message reply sent by an Attendee, the server first locates the corresponding scheduling object resource belonging to the Organizer.

The server MUST then update the "PARTSTAT" iCalendar property parameter value of each "ATTENDEE" iCalendar property in the scheduling object resource to match the changes indicated in the reply (taking into account the fact that an Attendee could have created a new overridden iCalendar component to indicate different participation status on one or more recurrence instances of a recurring event).

The server MUST also update or add the "SCHEDULE-STATUS" property parameter on each matching "ATTENDEE" iCalendar property and sets its value to that of the "REQUEST-STATUS" property in the reply, or to "2.0" if "REQUEST-STATUS" is not present (also taking into account recurrence instances). If there are multiple "REQUEST-STATUS" properties in the reply, the "SCHEDULE-STATUS" property parameter value is set to a comma-separated list of status codes, one from each "REQUEST-STATUS" property.

The server SHOULD send scheduling messages to all the other Attendees indicating the change in participation status of the Attendee replying, subject to the recurrence requirements of Section 5.2.6.

In this case, the scheduling message MUST only appear in the Organizer's scheduling Inbox collection once all automatic processing has been done.

6.2 Processing Organizer Requests, Additions, and Cancellations

For a scheduling message sent by an Organizer, the server first tries to locate a corresponding scheduling object resource belonging to the Attendee. If no matching scheduling object resource exists, the server treats the scheduling message as a new message, otherwise it is treated as an update.

In the case of a new message, the server MUST process the scheduling message and create a new scheduling object resource in an appropriate calendar collection for the Attendee.

In the case of an update, the server MUST process the scheduling message and update the matching scheduling object resource belonging to the Attendee to reflect the changes sent by the Organizer.

In any case, the scheduling message MUST only appear in the Attendee's scheduling Inbox collection once all automatic processing has been done.

6.3 Default Calendar Collection

The server is REQUIRED to process scheduling messages that specify a request for a new calendar component received for an Attendee by creating a new scheduling object resource in a calendar collection belonging to the Attendee. A Calendar User who can participate as an Attendee in a scheduling operation MUST have at least one valid calendar collection available. If there is no valid calendar collection, then the server MUST reject the attempt to deliver the scheduling message to the Attendee.

Servers MAY provide support for a default calendar collection, that is, the calendar collection in which new scheduling object resources will be created on reception of scheduling messages that specify a request for a new calendar component. The CALDAV:schedule-default-calendar-URL WebDAV property, which MAY be defined on the scheduling Inbox collection of a Calendar User, specifies if this Calendar User has a default calendar collection. See Section 12.2.

Servers MUST create new scheduling object resources in the default calendar collection, if the CALDAV:schedule-default-calendar-URL WebDAV property is set.

Servers MAY allow clients to change the default calendar collection by changing the value of the CALDAV:schedule-default-calendar-URL WebDAV property on the scheduling Inbox collection. However, they MUST ensure that any new value stored for that property refers to a valid calendar collection belonging to the owner of the scheduling inbox collection.

Servers MUST reject any attempt to delete the default calendar collection.

6.3.1 Additional Method Preconditions

This specification defines additional method preconditions (see Section 16 of WebDAV [RFC4918]) to provide machine-parsable information in error responses.

6.3.1.1 CALDAV:default-calendar-delete-allowed Precondition

Name: default-calendar-delete-allowed

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

Apply to: DELETE

Use with: 403 Forbidden

Purpose: (precondition) -- The client attempted to delete the calendar collection currently referenced by the CALDAV:schedule-default-calendar-URL property, or attempted to remove the CALDAV:schedule-default-calendar-URL property on the scheduling Inbox collection on a server that doesn't allow such operations.

Definition:

```
<!ELEMENT default-calendar-delete-allowed EMPTY>
```

Example:

```
<C:default-calendar-delete-allowed
    xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

6.3.1.2 CALDAV:valid-schedule-default-calendar-URL Precondition

Name: valid-schedule-default-calendar-URL

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

Apply to: PROPPATCH

Use with: 403 Forbidden

Purpose: (precondition) -- The client attempted to set the CALDAV:schedule-default-calendar-URL property to a DAV:href element that doesn't reference a valid calendar collection. Note: Servers that don't allow clients to change the CALDAV:schedule-default-calendar-URL property would simply return the DAV:cannot-modify-protected-property precondition defined in Section 16 of WebDAV [RFC4918].

Definition:

```
<!ELEMENT valid-schedule-default-calendar-URL EMPTY>
```

Example:

```
<C:valid-schedule-default-calendar-URL
    xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

6.4 Scheduling Messages as Notifications

Once the processing of an incoming scheduling message is completed by the server, the message is made available as a child resource in the scheduling Inbox collection of the Calendar User that received the message, to serve as a notification that a change has been made to the corresponding scheduling object resource. Scheduling messages are typically removed from the scheduling Inbox collection by the client once it has acknowledged the change.

7. Request for Busy Time Information

The POST method is used to request busy time information of one or more Calendar Users by targeting the request at a scheduling Outbox collection. The request body of a POST method MUST contain a "VFREEBUSY" calendar compoment with the "METHOD" iCalendar property set to the value "REQUEST" as specified in Section 3.3.2 of iTIP [I-D.ietf-calsify-2446bis]. The resource identified by the Request-URI MUST be a resource collection of type CALDAV:schedule-outbox (Section 4.1).

7.1 Status Codes

The following are examples of response codes one would expect to be used for this method. Note, however, that unless explicitly prohibited any 2/3/4/5xx series response code may be used in a response.

200 (OK) - The command succeeded.

204 (No Content) - The command succeeded.

400 (Bad Request) - The client has provided an invalid scheduling message.

403 (Forbidden) - The client cannot submit a scheduling message to the specified Request-URI.

404 (Not Found) - The URL in the Request-URI was not present.

423 (Locked) - The specified resource is locked and the client either is not a lock owner or the lock type requires a lock token to be submitted and the client did not submit it.

7.2 Additional Method Preconditions

This specification defines additional method preconditions for the POST method. Preconditions defined in WebDAV ACL [RFC3744] and CalDAV [RFC4791] that applies to the POST method are also listed here for completeness.

7.2.1 DAV:need-privileges Precondition

Name: need-privileges

Namespace: DAV:

Apply to: POST

Use with: 403 Forbidden

Purpose: (precondition) -- The currently authenticated user MUST be granted the CALDAV:schedule-send or CALDAV:schedule-send-freebusy privilege on the scheduling Outbox collection being targeted by the request.

Definition:

```
   <!ELEMENT DAV:need-privileges (DAV:resource)* >
   <!ELEMENT DAV:resource (DAV:href, DAV:privilege) >
```

Example:

```
   <D:need-privileges xmlns:D="DAV:"
xmlns:C="urn:ietf:params:xml:ns:caldav"/>
     <D:resource>
       <D:href>/home/bernard/calendars/outbox/</D:href>
       <D:privilege><C:schedule-send-freebusy/></D:privilege>
     </D:resource>
   </D:need-privileges>
```

7.2.2 CALDAV:supported-collection Precondition

Name: supported-collection

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

Apply to: POST

Use with: 400 Bad Request

Purpose: (precondition) -- The Request-URI MUST identify the location of a scheduling Outbox collection.

Definition:

```
   <!ELEMENT supported-collection EMPTY >
```

Example:

```
   <C:supported-collection
xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

7.2.3 CALDAV:supported-calendar-data Precondition

Name: supported-calendar-data

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

Apply to: POST

Use with: 400 Bad Request

Purpose: (precondition) -- The resource body submitted in the POST request MUST be a supported media type (e.g., text/calendar).

Definition:

```
   <!ELEMENT supported-calendar-data EMPTY >
```

Example:

```
   <C:supported-calendar-data
      xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

7.2.4 CALDAV:valid-calendar-data Precondition

Name: valid-calendar-data

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

Apply to: POST

Use with: 400 Bad Request

Purpose: (precondition) -- The resource submitted in the POST request MUST be valid data for the media type being specified (e.g., a valid iCalendar object).

Definition:

```
   <!ELEMENT valid-calendar-data EMPTY>
```

Example:

```
   <C:valid-calendar-data
xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

7.2.5 CALDAV:valid-scheduling-message Precondition

Name: valid-scheduling-message

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

Apply to: POST

Use with: 400 Bad Request

Purpose: (precondition) -- The resource submitted in the POST request MUST obey all restrictions specified for the POST request (e.g., the scheduling message follow the restrictions of iTIP).

Definition:

```
   <!ELEMENT valid-scheduling-message EMPTY >
```

Example:

```
   <C:valid-scheduling-message
      xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

7.2.6 CALDAV:organizer-allowed Precondition

Name: organizer-allowed

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

Apply to: POST

Use with: 409 Conflict

Purpose: (precondition) -- The Calendar User identified by the "ORGANIZER" property in the POST request's scheduling message MUST be the Calendar User (or one of the Calendar Users) associated with the scheduling Outbox collection being targeted by the request;

Definition:

```
   <!ELEMENT organizer-allowed EMPTY >
```

Example:

```
   <C:organizer-allowed
xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

7.2.7 CALDAV:max-resource-size Precondition

Name: max-resource-size

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

Apply to: POST

Use with: 403 Forbidden

Purpose: (precondition) -- The resource submitted in the POST request MUST have an octet size less than or equal to the value of the CALDAV:max-resource-size property (defined in Section 5.2.5 of [RFC4791]) specified on the scheduling Outbox collection targeted by the request.

Definition:

```
   <!ELEMENT max-resource-size EMPTY >
```

Example:

```
   <C:max-resource-size
xmlns:C="urn:ietf:params:xml:ns:caldav"/>
```

7.3 Response to a POST request

A POST request may deliver a scheduling message to one or more Calendar Users. Since the behavior of each recipient may vary, it is useful to get response status information for each recipient in the overall POST response. This specification defines a new XML response to convey multiple recipient status.

A response to a POST method that indicates status for one or more recipients MUST be a CALDAV:schedule-response XML element. This MUST contain one or more CALDAV:response elements for each recipient, with each of those containing elements that indicate which recipient they correspond to, the scheduling status for that recipient, any error codes and an optional description. See Section 14.1.

In the case of a freebusy request, the CALDAV:response elements can also contain CALDAV:calendar-data elements which contain freebusy information (e.g., an iCalendar VFREEBUSY component) indicating the busy state of the corresponding recipient, assuming that the freebusy request for that recipient succeeded. See Appendix B.5 for an example freebusy request and response.

8. Conditional Requests on Scheduling Object Resources

Because replies from Attendees and updates from Organizers are automatically processed by the server, clients might be in a situation where their copy of a calendar resource is different from the one currently on the server. When an Attendee or Organizer makes a change to the client's copy of the calendar resource, if the client writes the data to the server it could overwrite the changes already made there. Typically, HTTP clients use the ETag value and If-Match request header to avoid the "lost update problem".

Calendar user agents can also use ETag and If-Match to avoid this problem. However, when doing so the client will likely have to resolve the differences between the new resource and the original one, and the changes made by the Attendee or Organizer in the client. This can be a complicated comparison particularly when recurring components are present.

Additionally, the data on the server may change frequently as Attendees change their participation status, triggering updates to the Organizer and consequently other Attendees' copies of the scheduling object resource. If the ETag/If-Match behavior were used, clients would be forced to reconcile their cached copy of a scheduling object resource with the updated one on the server in order to attempt to write the user's changes back. This could lead to a race condition that can effectively result in a temporary denial of service when, for example, there is an event with a large Attendee list. A "storm" of updates will occur if Attendees all start responding at the same time, and this would prevent Attendees and the Organizer from being able to update their own copies of the scheduling object resource as the server copy is changing frequently.

What would be preferable is having the server determine the best way to merge changes made on the server with changes being made by the client. For example, if an Attendee changes their participation status and triggers an update to the Organizer's copy of the event, but the Organizer also updates their cached copy of the event and attempts to write it back, rather than failing on a conditional If-Match when the Organizer writes their data, the server would instead take the changes made by the Organizer and apply the Attendee changes and store the result. Thus a form of "weak" ETag matching behavior is needed such that scheduling changes made automatically on the server do not invalidate the tag, so that when clients store data conditionally based on the tag value, the server knows it can apply the merge behavior.

In order to do that, this specification introduces a new WebDAV resource property CALDAV:schedule-tag with a corresponding response header "Schedule-Tag", and a new "If-Schedule-Tag-Match" request header to allow client changes to be appropriately merged with server changes in the case where the changes on the server were the result of an "inconsequential" scheduling message update. An "inconsequential" scheduling message is one which simply updates the status information of Attendees due to a reply from an Attendee.

Servers MUST support conditional requests targeted at scheduling object resources using the "If-Schedule-Tag-Match" request header. Consequently, the server MUST support the "Schedule-Tag" response header and CALDAV:schedule-tag property for scheduling object resources. Servers MUST automatically resolve conflicts with "inconsequential" changes done to scheduling object resources when the "If-Schedule-Tag-Match" request header is specified.

The If-Schedule-Tag-Match request header applies only to the Request-URI, and not to the Destination of a COPY or MOVE in the same way as the If-Match request header.

Clients SHOULD use conditional requests using the If-Schedule-Tag-Match request header.

A response to any successful GET or PUT request targeting a scheduling object resource MUST include a Schedule-Tag response header with the value set to the same value as the CALDAV:schedule-tag WebDAV property of the resource.

A response to any successful COPY or MOVE request that specifies a Destination request header targeting a scheduling object resource MUST include a Schedule-Tag response header with the value set to the same value as the CALDAV:schedule-tag WebDAV property of the resource identified in the Request-URI.

The Schedule-Tag feature is designed to be used to address the problem of "inconsequential" changes on the server only. Normal ETag operations are used in all other cases, e.g., for synchronization.

The value of the CALDAV:schedule-tag property changes according to these rules:

- For an Organizer's copy of a scheduling object resource:
    1. The server MUST NOT change the CALDAV:schedule-tag property value when the scheduling object resource is updated as the result of automatically processing a scheduling message reply from an Attendee. For instance, when an Attendee replies to the Organizer, the CALDAV:schedule-tag property is unchanged after the Organizer's scheduling object resource has been automatically updated by the server with the Attendee's new participation status.
    2. The server MUST change CALDAV:schedule-tag property value when the schedulng object resource is changed directly via an HTTP request (e.g., PUT, COPY or MOVE).
- For an Attendee's copy of a scheduling object resource:
    1. The server MUST change the CALDAV:schedule-tag property value when the scheduling object resource is changed as the result of processing a scheduling message update from an Organizer that contains changes other than just the participation status of Attendees.
    2. The server MUST NOT change the CALDAV:schedule-tag property value when the scheduling object resource is changed as the result of processing a scheduling message update from an Organizer that only specify changes in the participation status of Attendees. For instance, when Attendee "A" replies to Organizer "O", and Attendee "B" receives a scheduling message update from Organizer "O" with the new participation status of Attendee "A", the CALDAV:schedule-tag property of Attendee "B"s scheduling object resource MUST NOT be changed.
    3. The server MUST change the CALDAV:schedule-tag property value when the scheduling object resource is changed directly via an HTTP request (e.g., PUT, COPY or MOVE).

8.1 PUT

Clients can use the If-Schedule-Tag-Match request header to do a conditional PUT request that ensures that "inconsequential" changes on the server do not result in a precondition error. The value of the request header is set to the last Schedule-Tag value received for the resource being modified. If the value of the If-Schedule-Tag-Match header matches the current value of the CALDAV:schedule-tag property the server MUST take any "ATTENDEE" property changes for all Attendees other than the owner of the scheduling object resource and apply those to the new resource being stored. Otherwise, the server MUST fail the request with a 412 Precondition Failed status code.

8.2 DELETE

Clients can use the If-Schedule-Tag-Match request header to do a conditional DELETE request that ensures that "inconsequential" changes on the server do not result in a precondition error. The value of the request header is set to the last Schedule-Tag value received for the resource being deleted. If the value of the If-Schedule-Tag-Match header matches the current value of the CALDAV:schedule-tag property the server performs the normal DELETE request processing for the resource. Otherwise, the server MUST fail the request with a 412 Precondition Failed status code.

8.3 COPY or MOVE

Clients can use the If-Schedule-Tag-Match request header to do conditional COPY or MOVE requests that ensures that "inconsequential" changes on the server do not result in a precondition error. The value of the request header is set to the last Schedule-Tag value received for the resource being copied or moved. If the value of the If-Schedule-Tag-Match header matches the current value of the CALDAV:schedule-tag property the server performs the normal COPY or MOVE request processing for the resource. Otherwise, the server MUST fail the request with a 412 Precondition Failed status code.

9. Other Scheduling Considerations

9.1 Attendee Participation Status

This section specifies additional requirements on the handling of the "PARTSTAT" property parameter when the "SCHEDULE-AGENT" property parameter on the corresponding "ATTENDEE" property is set to the value "SERVER" or is not present.

Clients SHOULD, and servers MUST reset the "PARTSTAT" property parameter value of all "ATTENDEE" properties, except the one that corresponds to the Organizer, to "NEEDS-ACTION" when the Organizer reschedules an event.

A reschedule of an event occurs when any "DTSTART", "DTEND", "DURATION", "DUE", "RRULE", "RDATE", or "EXDATE" property changes in a calendar component such that existing recurrence instances are impacted by the changes, as shown in the table below.

Property	Description
DTSTART DTEND DURATION DUE	Any change to these properties MUST result in "PARTSTAT" being set to "NEEDS-ACTION"
RRULE	A change to or addition of this property that results in the addition of new recurring instances or a change in time for existing recurring instances MUST result in "PARTSTAT" being reset to "NEEDS-ACTION" on each affected component.
RDATE	A change to or addition of this property that results in the addition of new recurring instances or a change in time for existing recurring instances MUST result in "PARTSTAT" being reset to "NEEDS-ACTION" on each affected component.
EXDATE	A change to or removal of this property that results in the re-instatement of recurring instances MUST result in "PARTSTAT" being reset to "NEEDS-ACTION" on each affected component.

The server MAY allow the Organizer's client to change an Attendee's "PARTSTAT" property parameter value to "NEEDS-ACTION" at any other time (e.g., when the "LOCATION" property value changes, an Organizer might wish to re-invite Attendees who may be impacted by the change).

9.2 Schedule Status Values

When scheduling with an Attendee there are two types of status information that can be returned during the transaction. The first status information is a "delivery" status that indicates whether the scheduling message from the Organizer to the Attendee was delivered or not, or what the current status of delivery is. The second status information is a "reply" status corresponding to the Attendee's own "REQUEST-STATUS" information from the scheduling message reply that is sent back to the Organizer.

Similarly, when an Attendee sends a reply back to the Organizer, there will be "delivery" status information for the scheduling message sent to the Organizer. However, there is no "REQUEST-STATUS" sent back by the Organizer, so there is no equivalent of the "reply" status as per scheduling messages to Attendees.

The "delivery" status information on an "ORGANIZER" or "ATTENDEE" iCalendar property is conveyed in the "SCHEDULE-STATUS" property parameter value (Section 10.3). The status code value for "delivery" status can be one of the following:

Delivery Status Code	Description
1.0	The scheduling message is pending. i.e. the server is still in the process of sending the message. The status code value can be expected to change once the server has completed its sending and delivery attempts.
1.1	The scheduling message has been successfully sent. However, the server does not have explicit information about whether the scheduling message was successfully delivered to the recipient. This state can occur with "store and forward" style scheduling protocols such as iMIP [I-D.ietf-calsify-rfc2447bis] (iTIP using email).
1.2	The scheduling message has been successfully delivered.
3.7	The scheduling message was not delivered because the server did not recognize the calendar user address as a valid calendar user.
3.8	The scheduling message was not delivered due to insufficient privileges.
5.1	The scheduling message was not delivered because the server could not complete delivery of the message. This is likely due to a temporary failure, and the originator can try to send the message again at a later time.
5.2	The scheduling message was not delivered because the server was not able to find a suitable way to deliver the message. This is likely a permanent failure, and the originator should not try to send the message again, at least without verifying/correcting the calendar user address of the recipient.
5.3	The scheduling message was not delivered and was rejected because scheduling with that recipient is not allowed. This is likely a permanent failure, and the originator should not try to send the message again.

The status code for "reply" status can be any of the valid iTIP [I-D.ietf-calsify-2446bis] "REQUEST-STATUS" values.

9.3 Organizer is an Attendee

The Organizer of a scheduled event may also be an Attendee of that event. In such cases the server MUST NOT send a scheduling message to the Attendee that matches the Organizer.

10. Additional iCalendar Property Parameters

This specification defines additional iCalendar property parameters to support the CalDAV scheduling extensions.

10.1 Schedule Agent Parameter

Parameter Name: SCHEDULE-AGENT

Purpose: To specify the agent expected to deliver scheduling messages to the corresponding Organizer or Attendee.

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

```
   scheduleagentparam = "SCHEDULE-AGENT" "="
                     ("SERVER"      ; The server
handles scheduling
                    / "CLIENT"      ; The client
handles scheduling
                    / "NONE"        ; No
automatic scheduling
                    / x-name        ;
Experimental type
                    / iana-token)   ; Other IANA
registered type
                                    ;
                                    ; Default is
SERVER
```

Description: This property parameter MAY be specified on "ORGANIZER" or "ATTENDEE" iCalendar properties. In the absence of this parameter, the value "SERVER" MUST be used for the default behavior. The value determines whether or not an automatic scheduling transaction on a server will cause a scheduling message to be sent to the corresponding Calendar User identified by the "ORGANIZER" or "ATTENDEE" property value. When the value "SERVER" is specified, or the parameter is absent, then it is the server's responsibility to send a scheduling message as part of an automatic scheduling transaction. When the value "CLIENT" is specified, that indicates that the client is handling scheduling messages with the Calendar User itself. When "NONE" is specified, no scheduling messages are being sent to the Calendar User.

Servers MUST NOT include this parameter in any scheduling messages sent as the result of an automatic scheduling transaction.

Clients SHOULD NOT include this parameter in any scheduling messages that they themselves send.

Servers and clients MUST treat x-name and iana-token values they don't recognize the same way as they would the "NONE" value.

Example:

```
ATTENDEE;SCHEDULE-AGENT=SERVER:mailto:bernard@example.com

ATTENDEE;SCHEDULE-AGENT=NONE:mailto:cyrus@example.com
```

10.2 Schedule Force Send Parameter

Parameter Name: SCHEDULE-FORCE-SEND

Purpose: To force a scheduling message to be sent to the Calendar User specified by the property.

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

```
   scheduleforcesendparam = "SCHEDULE-FORCE-SEND"
"="
                           ("REQUEST"    ; Force
a "REQUEST"
                          / "REPLY"      ; Force
a "REPLY"
                          / iana-token)  ; IANA
registered method
```

Description: This property parameter MAY be specified on "ATTENDEE" and "ORGANIZER" properties on which the "SCHEDULE-AGENT" property parameter is set to the value "SERVER" or is not specified. This property parameter is used to force a server to send a scheduling message to a specific Calendar User in situations where the server would not send a scheduling message otherwise (e.g., when no change that warrants the delivery of a new scheduling message was performed on the scheduling object resource). An Organizer MAY specify this parameter on an "ATTENDEE" property with the value "REQUEST" to force a "REQUEST" scheduling message to be sent to this Attendee. An Attendee MAY specify this parameter on the "ORGANIZER" with the value "REPLY" to force a "REPLY" scheduling message to be sent to the Organizer.

Servers MUST NOT preserve this property parameter in scheduling object resources, nor include it in any scheduling messages sent as the result of an automatic scheduling transaction.

Clients SHOULD NOT include this parameter in any scheduling messages that they themselves send.

Servers MUST set the "SCHEDULE-STATUS" parameter of the "ATTENDEE" or "ORGANIZER" to 2.3 (i.e., "Success, invalid property parameter ignored", see Section 3.6 of [I-D.ietf-calsify-2446bis]) when the "SCHEDULE-FORCE-SEND" parameter is set to a x-name or iana-token value they don't recognize.

Example:

```
ATTENDEE;SCHEDULE-FORCE-SEND=REQUEST:mailto:bernard@example.com

ORGANIZER;SCHEDULE-FORCE-SEND=REPLY:mailto:bernard@example.com
```

10.3 Schedule Status Parameter

Parameter Name: SCHEDULE-STATUS

Purpose: To specify the status codes returned from processing of the most recent scheduling message sent to the corresponding Attendee, or received from the corresponding Organizer.

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

```
   schedulestatusparam = "SCHEDULE-STATUS" "="
```

```
                        ( statcode
                       / DQUOTE statcode *(","
statcode) DQUOTE)
   ; "statcode" is defined in Section 3.8.8.3 of
   ; [I-D.ietf-calsify-rfc2445bis]. Value is a
single
   ; "statcode" or a comma-separated list of
"statcode" values.
```

Description:

This property parameter MAY be specified on the "ATTENDEE" and "ORGANIZER" properties.

Servers MUST add this property parameter to any "ATTENDEE" properties corresponding to Calendar Users who were sent a scheduling message via an automatic scheduling transaction. Clients SHOULD NOT change or remove this parameter if it was provided by the server. In the case where the client is handling the scheduling, the client MAY add, change or remove this parameter to indicate the last scheduling message status it received.

Servers MUST add this parameter to any "ORGANIZER" properties corresponding to Calendar Users who were sent a scheduling message reply by an Attendee via an automatic scheduling transaction. Clients SHOULD NOT change or remove this parameter if it was provided by the server. In the case where the client is handling the scheduling the client MAY add, change or remove this parameter to indicate the last scheduling message status it received.

Servers MUST NOT include this parameter in any scheduling messages sent as the result of an automatic scheduling transaction.

Clients SHOULD NOT include this parameter in any scheduling messages that they themselves send.

Suitable values for this property parameter are described in Section 9.2.

Example:

```
ATTENDEE;SCHEDULE-STATUS="2.0":mailto:bernard@example.com

ATTENDEE;SCHEDULE-STATUS="2.0,2.4":mailto:cyrus@example.com
```

11. Additional Message Header Fields

This specification defines additional HTTP request and response headers for use with CalDAV.

11.1 Schedule-Reply Request Header

```
Schedule-Reply = "Schedule-Reply" ":" ("T" | "F")
```

Example:

```
Schedule-Reply: F
```

When an Attendee executes an HTTP DELETE request on a scheduling object resource, and the Schedule-Reply header is not present, or present and set to the value "T", the server MUST send an appropriate reply scheduling message with the Attendee's "PARTSTAT" iCalendar property parameter value set to "DECLINED" as part of its normal automatic scheduling transaction processing.

When the Schedule-Reply header is set to the value "F", the server MUST NOT send a scheduling message as part of its normal automatic scheduling transaction processing.

The Schedule-Reply request header is used by a client to indicate to a server whether or not an automatic scheduling transaction should occur when an Attendee deletes a scheduling object resource. In particular it controls whether a reply scheduling message is sent to the Organizer as a result of the deletion. There are situations in which unsolicited scheduling messages need to be silently deleted (or ignored) for security or privacy reasons. This request header allows the scheduling object resource to be deleted if such a need arises.

All scheduling object resources MUST support the Schedule-Reply request header.

11.2 Schedule-Tag Response Header

The Schedule-Tag response header provides the current value of the CALDAV:schedule-tag property value. The behavior of this response header is described in Section 8.

All scheduling object resources MUST support the Schedule-Tag header.

```
Schedule-Tag = "Schedule-Tag" ":" opaque-tag
; "opaque-tag" is defined in Section 3.11 of [RFC2616]
```

Example:

```
Schedule-Tag: "12ab34-cd56ef"
```

11.3 If-Schedule-Tag-Match Request Header

The If-Schedule-Tag-Match request header field is used with a method to make it conditional. Clients can set this header to the value returned in the Schedule-Tag response header, or the CALDAV:schedule-tag property, of a scheduling object resource previously retrieved from the server to avoid overwriting "consequential" changes to the scheduling object resource.

All scheduling object resources MUST support the If-Schedule-Tag-Match header.

```
If-Schedule-Tag-Match = "If-Schedule-Tag-Match" ":" opaque-tag
; "opaque-tag" is defined in Section 3.11 of [RFC2616]
```

Example:

```
   If-Schedule-Tag-Match: "12ab34-cd56ef"
```

12. Additional WebDAV Properties

The CalDAV scheduling extension defines the following new WebDAV properties for use with CalDAV.

12.1 CALDAV:schedule-calendar-transp Property

Name: schedule-calendar-transp

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

Purpose: Determines whether the calendar object resources in a calendar collection will affect the owner's freebusy.

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

COPY/MOVE behavior: This property value SHOULD be kept during a MOVE operation, but is normally re-initialized when a resource is created with a COPY. It should not be set in a COPY.

Description: This property SHOULD be defined on all calendar collections. If present, it contains one of two XML elements that indicate whether the calendar object resources in the calendar collection should contribute to the owner's freebusy or not. When the CALDAV:opaque element is used, all calendar object resources in the corresponding calendar collection MUST contribute to freebusy, assuming access privileges and other iCalendar properties allow it to. When the CALDAV:transparent XML element is used, the calendar object resources in the corresponding calendar collection MUST NOT contribute to freebusy.

If this property is not present on a calendar collection, then the default value CALDAV:opaque MUST be assumed.

Definition:

```
   <!ELEMENT schedule-calendar-transp (opaque |
transparent) >

   <!ELEMENT opaque EMPTY>
   <!-- Affect busy time searches -->

   <!ELEMENT transparent EMPTY>
   <!-- Invisible to busy time searches -->
```

Example:

```
<C:schedule-calendar-transp
     xmlns:C="urn:ietf:params:xml:ns:caldav">
  <C:opaque/>
</C:schedule-calendar-transp>
```

12.2 CALDAV:schedule-default-calendar-URL Property

Name: schedule-default-calendar-URL

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

Purpose: Specifies a default calendar for an Attendee where new scheduling object resources are created.

Protected: This property MAY be protected in the case where a server does not support changing the default calendar, or does not support a default calendar.

COPY/MOVE behavior: This property is only defined on a scheduling Inbox collection which cannot be moved or copied.

Description: This property MAY be defined on a scheduling Inbox collection. If present, it contains zero or one DAV:href XML elements. When a DAV:href element is present, its value indicates a URL to a calendar collection that is used as the default calendar. When no DAV:href element is present, it indicates that there is no default calendar. In the absence of this property there is no default calendar. When there is no default calendar the server is free to choose the calendar in which a new scheduling object resource is created. See Section 6.3.

Definition:

```
<!ELEMENT schedule-default-calendar-URL
(DAV:href?) >
```

Example:

```
   <C:schedule-default-calendar-URL
xmlns:D="DAV:"
xmlns:C="urn:ietf:params:xml:ns:caldav">
<D:href>/home/cyrus/calendars/work/</D:href>
   </C:schedule-default-calendar-URL>
```

12.3 CALDAV:schedule-tag Property

Name: schedule-tag

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

Purpose: Indicates whether a scheduling object resource has had a "consequential" change made to it.

Value: opaque-tag (defined in Section 3.11 of [RFC2616])

Protected: This property MUST be protected as only the server can update the value.

COPY/MOVE behavior: This property is only defined on scheduling object resources. It MUST be preserved when a scheduling object resource is copied or moved and the resulting resource is also a scheduling object resource. If the source resource is not a scheduling object resource but the destination resource is, this property MUST be added to the destination resource.

Description: The CALDAV:schedule-tag property MUST be defined on all scheduling object resources. This property is described in Section 8.

Definition:

```
   <!ELEMENT schedule-tag (#PCDATA) >
```

Example:

```
   <C:schedule-tag
xmlns:C="urn:ietf:params:xml:ns:caldav"
   >"12345-67890"</C:schedule-tag>
```

13. Scheduling Access Control

13.1 Scheduling Privileges

CalDAV servers MUST support and adhere to the requirements of WebDAV ACL [RFC3744]. Furthermore, CalDAV servers that advertise support for the "calendar-auto-schedule" feature MUST also support the scheduling privileges defined in this section.

All the scheduling privileges MUST be non-abstract and MUST appear in the DAV:supported-privilege-set property of scheduling Outbox and Inbox collections on which they are defined.

The tables specified in Appendix A clarify which scheduling methods (e.g., "REQUEST", "REPLY", etc.) are controlled by each scheduling privilege defined in this section.

13.1.1 Privileges on Scheduling Inbox Collections

This section defines new WebDAV ACL privileges that are for use on scheduling Inbox collections. These privileges determine whether delivery of scheduling messages from a calendar user is allowed by the calendar user who "owns" the scheduling Inbox collection. This allows calendar users to choose which other calendar users can schedule with them.

Note that when a scheduling message is delivered to a calendar user, in addition to a scheduling object resource being created in the calendar user's scheduling Inbox collection, a new scheduling object resource might be created or an existing one updated in a calendar belonging to the calendar user. In that case, the ability to create or update the scheduling object resource in the calendar is controlled by the privileges assigned to the scheduling Inbox collection.

The privileges defined in this section are ignored if applied to a resource other than a scheduling Inbox collection.

13.1.1.1 CALDAV:schedule-deliver Privilege

CALDAV:schedule-deliver is an aggregate privilege that contains all the scheduling privileges that control the processing and delivery of incoming scheduling messages, that is, CALDAV:schedule-deliver-invite and CALDAV:schedule-deliver-reply, as well as freebusy requests targeted at the owner of the scheduling Inbox collection, that is, CALDAV:schedule-query-freebusy.

```
<!ELEMENT schedule-deliver EMPTY >
```

13.1.1.2 CALDAV:schedule-deliver-invite Privilege

The CALDAV:schedule-deliver-invite privilege controls the processing and delivery of scheduling messages coming from an Organizer.

```
<!ELEMENT schedule-deliver-invite EMPTY >
```

13.1.1.3 CALDAV:schedule-deliver-reply Privilege

The CALDAV:schedule-deliver-reply privilege controls the processing and delivery of scheduling messages coming from an Attendee.

```
<!ELEMENT schedule-deliver-reply EMPTY >
```

13.1.1.4 CALDAV:schedule-query-freebusy Privilege

The CALDAV:schedule-query-freebusy privilege controls freebusy requests targeted at the owner of the scheduling Inbox collection.

<!ELEMENT schedule-query-freebusy EMPTY >

13.1.2 Privileges on Scheduling Outbox Collections

This section defines new WebDAV ACL privileges that are defined for use on scheduling Outbox collections. These privileges determine which calendar users are allowed to send scheduling messages on behalf of the calendar user who "owns" the scheduling Outbox collection. This allows calendar users to choose other calendar users who can act on their behalf to send schedule messages to other calendar users (e.g. assistants working on behalf of their boss).

The privileges defined in this section are ignored if applied to a resource other than a scheduling Outbox collection.

13.1.2.1 CALDAV:schedule-send Privilege

CALDAV:schedule-send is an aggregate privilege that contains all the scheduling privileges that control the use of methods that will cause scheduling messages to be delivered to other users, that is, CALDAV-schedule-send-invite and CALDAV-schedule-send-reply, as well as freebusy requests to be targeted at other users, that is, CALDAV-schedule-send-freebusy.

<!ELEMENT schedule-send EMPTY >

13.1.2.2 CALDAV:schedule-send-invite Privilege

The CALDAV:schedule-send-invite privilege controls the sending of scheduling messages by Organizers.

Users granted the DAV:bind privilege on a calendar collection, or DAV:write privilege on scheduling object resources, will also need the CALDAV:schedule-send-invite privilege granted on the scheduling Outbox collection of the owner of the calendar collection or scheduling object resource in order to be allowed to create, modify or delete scheduling object resources in a way that will trigger the CalDAV server to deliver organizer scheduling messages to other calendar users.

<!ELEMENT schedule-send-invite EMPTY >

13.1.2.3 CALDAV:schedule-send-reply Privilege

The CALDAV:schedule-send-invite privilege controls the sending of scheduling messages by Attendees.

Users granted the DAV:bind privilege on a calendar collection, or DAV:write privilege on scheduling object resources, will also need the CALDAV:schedule-send-reply privilege granted on the scheduling Outbox collection of the owner of the calendar collection or scheduling object resource in order to be allowed to create, modify or delete scheduling object resources in a way that will trigger the CalDAV server to deliver attendee scheduling messages to other calendar users.

<!ELEMENT schedule-send-reply EMPTY >

13.1.2.4 CALDAV:schedule-send-freebusy Privilege

The CALDAV:schedule-send-freebusy privilege controls the use of the POST method to submit scheduling messages that specify the scheduling method "REQUEST" with a "VFREEBUSY" calendar component.

<!ELEMENT schedule-send-freebusy EMPTY >

13.1.3 Aggregation of Scheduling Privileges

Server implementations MUST aggregate the scheduling privileges as follows:

DAV:all MUST contain CALDAV:schedule-send and CALDAV:schedule-deliver;

CALDAV:schedule-send MUST contain CALDAV:schedule-send-invite, CALDAV:schedule-send-reply, and CALDAV:schedule-send-freebusy;

CALDAV:schedule-deliver MUST contain CALDAV:schedule-deliver-invite, CALDAV:schedule-deliver-reply, and CALDAV:schedule-query-freebusy.

The following diagram illustrates how scheduling privileges are aggregated according to the above requirements.

```
      [DAV:all] (aggregate)
          |
          +-- [CALDAV:schedule-deliver] (aggregate)
          |       |
          |       +-- [CALDAV:schedule-deliver-invite]
          |       +-- [CALDAV:schedule-deliver-reply]
          |       +-- [CALDAV:schedule-query-freebusy]
          |
          +-- [CALDAV:schedule-send] (aggregate)
                  |
                  +-- [CALDAV:schedule-send-invite]
                  +-- [CALDAV:schedule-send-reply]
                  +-- [CALDAV:schedule-send-freebusy]
```

13.2 Additional Principal Properties

This section defines new properties for WebDAV principal resources as defined in RFC3744 [RFC3744]. These properties are likely to be protected but the server MAY allow them to be written by appropriate users.

13.2.1 CALDAV:schedule-inbox-URL Property

Name: schedule-inbox-URL

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

Purpose: Identify the URL of the scheduling Inbox collection owned by the associated principal resource.

Protected: This property MAY be protected.

PROPFIND behavior: This property SHOULD NOT be returned by a PROPFIND allprop request (as defined in Section 14.2 of [RFC4918]).

COPY/MOVE behavior: This property value SHOULD be preserved in COPY and MOVE operations.

Description: This property is needed for a client to determine where the scheduling Inbox collection of the current user is located so that processing of scheduling messages can occur.

Definition:

```
   <!ELEMENT schedule-inbox-URL (DAV:href)>
```

13.2.2 CALDAV:schedule-outbox-URL Property

Name: schedule-outbox-URL

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

Purpose: Identify the URL of the scheduling Outbox collection owned by the associated principal resource.

Protected: This property MAY be protected.

PROPFIND behavior: This property SHOULD NOT be returned by a PROPFIND allprop request (as defined in Section 14.2 of [RFC4918]).

COPY/MOVE behavior: This property value SHOULD be preserved in COPY and MOVE operations.

Description: This property is needed for a client to determine where the scheduling Outbox collection of the current user is located so that sending of scheduling messages can occur.

Definition:

```
   <!ELEMENT schedule-outbox-URL DAV:href>
```

13.2.3 CALDAV:calendar-user-address-set Property

Name: calendar-user-address-set

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

Purpose: Identify the calendar addresses of the associated principal resource.

Protected: This property MAY be protected.

PROPFIND behavior: This property SHOULD NOT be returned by a PROPFIND allprop request (as defined in Section 14.2 of [RFC4918]).

COPY/MOVE behavior: This property value SHOULD be preserved in COPY and MOVE operations.

Description: Support for this property is REQUIRED. This property is needed to map calendar user addresses in iCalendar data to principal resources and their associated scheduling Inbox and Outbox collections. In the event that a user has no well defined identifier for their calendar user address, the URI of their principal resource can be used. This property SHOULD be searchable using the DAV:principal-property-search REPORT. The DAV:principal-search-property-set REPORT SHOULD identify this property as such.

Definition:

```
   <!ELEMENT calendar-user-address-set
(DAV:href*)>
```

Example:

```
   <C:calendar-user-address-set xmlns:D="DAV:"
xmlns:C="urn:ietf:params:xml:ns:caldav">
<D:href>mailto:bernard@example.com</D:href>
<D:href>mailto:bernard.desruisseaux@example.com</D:href>
   </C:calendar-user-address-set>
```

13.2.4 CALDAV:calendar-user-type Property

Name: calendar-user-type

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

Purpose: Identifies the calendar user type of the associated principal resource.

Value: Same values allowed for the iCalendar "CUTYPE" property parameter defined in Section 3.2.3 of [I-D.ietf-calsify-rfc2445bis].

Protected: This property MAY be protected.

PROPFIND behavior: This property SHOULD NOT be returned by a PROPFIND allprop request (as defined in Section 14.2 of [RFC4918]).

COPY/MOVE behavior: This property value SHOULD be preserved in COPY and MOVE operations.

Description: This property MAY be defined on principal resources to indicate the type of calendar user associated with this principal resource. Its value is the same as the iCalendar "CUTYPE" property parameter that can be used on "ATTENDEE" properties. This property SHOULD be searchable using the DAV:principal-property-search REPORT. The DAV:principal-search-property-set REPORT SHOULD identify this property as such.

Definition:

```
   <!ELEMENT calendar-user-type (#PCDATA) >
```

Example:

```
   <C:calendar-user-type
xmlns:C="urn:ietf:params:xml:ns:caldav">INDIVIDUAL<
    /C:calendar-user-type>
```

14. XML Element Definitions

14.1 CALDAV:schedule-response XML Element

Name: schedule-response

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

Purpose: Contains the set of responses for a POST method request.

Description: See Section 7.3.

Definition:

```
    <!ELEMENT schedule-response (response*)>
```

14.2 CALDAV:response XML Element

Name: response

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

Purpose: Contains a single response for a POST method request.

Description: See Section 7.3.

Definition:

```
<!ELEMENT response (recipient,
                    request-status,
                    calendar-data?,
                    DAV:error?,
                    DAV:responsedescription?)>
```

14.3 CALDAV:recipient XML Element

Name: recipient

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

Purpose: The calendar user address that the enclosing response for a POST method request is for.

Description: See Section 7.3.

Definition:

```
    <!ELEMENT recipient (DAV:href)>
```

14.4 CALDAV:request-status XML Element

Name: request-status

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

Purpose: The iTIP "REQUEST-STATUS" property value for this response.

Description: See Section 7.3.

Definition:

```
    <!ELEMENT request-status (#PCDATA) >
```

15. Security Considerations

The process of scheduling involves the sending and receiving of scheduling messages. As a result, the security problems related to messaging in general are relevant here. In particular the authenticity of the scheduling messages needs to be verified. Servers and clients MUST use an HTTP connection protected with TLS as defined in [RFC2818] for all scheduling transactions.

15.1 Verifying Scheduling Transactions

When handling a scheduling transaction:

> Servers MUST verify that the principal associated with the DAV:owner of the calendar collection in which a scheduling object resource is being manipulated contains a CALDAV:schedule-outbox-URL property value.
>
> Servers MUST verify that the currently authenticated user has the CALDAV:schedule-send privilege, or a suitable sub-privilege aggregated under this privilege, on the scheduling Outbox collection of the DAV:owner of the calendar collection in which a scheduling object resource is being manipulated.
>
> Servers MUST only deliver scheduling messages to recipients when the CALDAV:schedule-deliver privilege, or a suitable sub-privilege aggregated under this privilege, is granted on the recipient's scheduling Inbox collection for the principal associated with the DAV:owner of the calendar collection in which a scheduling object resource is being manipulated.
>
> To prevent impersonation of calendar users, the server MUST verify that the "ORGANIZER" property in an organizer scheduling object resource matches one of the calendar user addresses of the DAV:owner of the calendar collection in which the resource is stored.
>
> To prevent spoofing of an existing scheduling object resource, servers MUST verify that the "UID" iCalendar property value in a new scheduling object resource does not match that of an existing scheduling object resource with a different "ORGANIZER" property value.

15.2 Verifying Busy Time Information Requests

When handling a POST request on a scheduling Outbox collection:

> Servers MUST verify that the principal associated with the calendar user address specified in the "ORGANIZER" property of the scheduling message data in the request contains a CALDAV:schedule-outbox-URL property value that matches the scheduling Outbox collection targeted by the request.
>
> Servers MUST verify that the currently authenticated user has the CALDAV:schedule-send privilege, or a sub-privilege aggregated under this privilege, on the scheduling Outbox collection targeted by the request.
>
> Servers MUST only return valid freebusy information for recipients when the CALDAV:schedule-deliver privilege, or a sub-privilege aggregated under this privilege, is granted on the recipient's scheduling Inbox collection for the principal associated with the DAV:owner of the scheduling Outbox collection targeted by the request.

15.3 Privacy Issues

As noted in Section 11.1, Attendees can use the Schedule-Reply request header with the value set to "F" to prevent notification to an Organizer that a scheduling object resource was deleted. This allows Attendees to remove unwanted scheduling messages without any response to the Organizer.

16. IANA Considerations

16.1 Message Header Field Registrations

The message header fields below should be added to the Permanent Message Header Field Registry (see [RFC3864]).

16.1.1 Schedule-Reply

Header field name: Schedule-Reply

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document(s): this specification (Section 11.1)

Related information: none

16.1.2 Schedule-Tag

Header field name: Schedule-Tag

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document(s): this specification (Section 11.2)

Related information: none

16.1.3 If-Schedule-Tag-Match

Header field name: If-Schedule-Tag-Match

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document(s): this specification (Section 11.3)

Related information: none

16.2 iCalendar Property Parameter Registrations

The following iCalendar property parameters should be added to the iCalendar Property Parameter Registry defined in Section 8.3.3 of [I-D.ietf-calsify-rfc2445bis].

Parameter	Status	Reference
SCHEDULE-AGENT	Current	RFCXXXX, Section 10.1

Parameter	Status	Reference
SCHEDULE-STATUS	Current	RFCXXXX, Section 10.3
SCHEDULE-FORCE-SEND	Current	RFCXXXX, Section 10.2

16.3 Additional iCalendar Elements Registries

The IANA should create and maintain the following additional registries for iCalendar elements with pointers to appropriate reference documents.

16.3.1 Schedule Agent Values Registry

The following table should be used to initialize the schedule agent values registry.

Schedule Agent	Status	Reference
SERVER	Current	RFC XXXX, Section 10.1
CLIENT	Current	RFC XXXX, Section 10.1
NONE	Current	RFC XXXX, Section 10.1

16.3.2 Schedule Force Send Values Registry

The following table should be used to initialize the schedule send values registry.

Schedule Force Send	Status	Reference
REQUEST	Current	RFC XXXX, Section 10.2
REPLY	Current	RFC XXXX, Section 10.2

17. Acknowledgements

The authors would like to thank the following individuals for contributing their ideas and support for writing this specification: Mike Douglass, Lisa Dusseault, Helge Hess, Arnaud Quillaud, Julian F. Reschke, Wilfredo Sanchez Vega, Simon Vaillancourt, and Jim Whitehead.

The authors would also like to thank the Calendaring and Scheduling Consortium for advice with this specification, and for organizing interoperability testing events to help refine it.

18. References

18.1 Normative References

[I-D.ietf-calsify-2446bis] Daboo, C, "iCalendar Transport-Independent Interoperability Protocol (iTIP)", Internet-Draft draft-ietf-calsify-2446bis-09 (work in progress), April 2009.

[I-D.ietf-calsify-rfc2445bis] Desruisseaux, B, "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", Internet-Draft draft-ietf-calsify-rfc2445bis-10 (work in progress), April 2009.

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

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3744] Clemm, G., Reschke, J. F., Sedlar, E., and J. Whitehead, "Web Distributed Authoring and Versioning (WebDAV) Access Control Protocol", RFC 3744, May 2004.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

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

[RFC4918] Dusseault, L., "HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)", RFC 4918, June 2007.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[W3C.REC-xml-20081126] Maler, E., Yergeau, F., Sperberg-McQueen, C., Paoli, J., and T. Bray, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

18.2 Informative References

[I-D.ietf-calsify-rfc2447bis] Melnikov, A, "iCalendar Message-Based Interoperability Protocol (iMIP)", Internet-Draft draft-ietf-calsify-rfc2447bis-05 (work in progress), June 2008.

[RFC3283] Mahoney, B., Babics, G., and A. Taler, "Guide to Internet Calendaring", RFC 3283, June 2002.

Authors' Addresses

Cyrus Daboo
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
USA
EMail: cyrus@daboo.name
URI: http://www.apple.com/

Bernard Desruisseaux
Oracle Corporation
600 Blvd. de Maisonneuve West
Suite 1900
Montreal, QC H3A 3J2
CANADA
EMail: bernard.desruisseaux@oracle.com
URI: http://www.oracle.com/

A. Scheduling Privileges Summary

A.1 Scheduling Inbox Privileges

The following tables specify which scheduling privileges grant the right to a calendar user to deliver a scheduling message to the scheduling Inbox collection of another calendar user. The appropriate behavior depends on the calendar component type as well as the scheduling "METHOD" specified in the scheduling message.

	METHOD for VEVENT and VTODO			
Scheduling Inbox Privilege	REQUEST	REPLY	ADD	CANCEL
schedule-deliver	*	*	*	*
schedule-deliver-invite	*		*	*
schedule-deliver-reply		*		
schedule-query-freebusy				

	METHOD for VFREEBUSY
Scheduling Inbox Privilege	REQUEST
schedule-deliver	*
schedule-deliver-invite	
schedule-deliver-reply	
schedule-query-freebusy	*

A.2 Scheduling Outbox Privileges

The following tables specify which scheduling privileges grant the right to a Calendar User to perform busy time information requests and to submit scheduling messages to other Calendar Users as the result of a scheduling transaction. The appropriate behavior depends on the calendar component type as well as the scheduling "METHOD" specified in the scheduling message.

	METHOD for VEVENT and VTODO			
Scheduling Outbox Privilege	REQUEST	REPLY	ADD	CANCEL
schedule-send	*	*	*	*
schedule-send-invite	*		*	*
schedule-send-reply		*		
schedule-send-freebusy				

	METHOD for VFREEBUSY
Scheduling Outbox Privilege	REQUEST
schedule-send	*
schedule-send-invite	
schedule-send-reply	
schedule-send-freebusy	*

B. Example Scheduling Transactions

This section describes some example scheduling transactions that give a general idea of how scheduling is carried out between CalDAV clients and servers from the perspective of meeting Organizers and Attendees.

In the following examples the requests and responses are incomplete and are only for illustrative purposes. In particular, HTTP authentication headers and behaviors are not shown, even though they are required in normal operation.

B.1 Example: Organizer Inviting Multiple Attendees

In the following example, Cyrus invites Wilfredo, Bernard and Mike to a single instance event by simply creating a new scheduling object resource in one of his calendar collection by using the PUT method.

>> Request <<

```
PUT /home/cyrus/calendars/work/9263504FD3AD.ics HTTP/1.1
Host: cal.example.com
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T185254Z
DTSTART:20090602T160000Z
DTEND:20090602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";CUTYPE=INDIVIDUAL;PARTSTAT
 =NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:wilfredo@
 example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@ex
 ample.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE:mailto:mike@example.org
END:VEVENT
END:VCALENDAR
```

>> Response <<

```
HTTP/1.1 201 Created
Content-Length: 0
Date: Tue, 02 Jun 2009 18:52:54 GMT
Last-Modified: Tue, 02 Jun 2009 18:52:54 GMT
ETag: "d85561cfe74a4e785eb4639451b434fb"
Schedule-Tag: "488177c8-2ea7-4176-a6cb-fab8cfccdea2"
```

Once the event creation has been completed, Cyrus's client will retrieve the event back from the server to get the schedule status of each Attendee. In this example, the server reports that a scheduling message was delivered to Wilfredo, a scheduling message is still pending for Bernard, and the server was unable to deliver a scheduling message to Mike.

>> Request <<

```
GET /home/cyrus/calendars/work/9263504FD3AD.ics HTTP/1.1
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 18:52:58 GMT
Last-Modified: Tue, 02 Jun 2009 18:52:58 GMT
ETag: "eb897deabc8939589da116714bc99265"
Schedule-Tag: "488177c8-2ea7-4176-a6cb-fab8cfccdea2"
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T185300Z
DTSTART:20090602T160000Z
DTEND:20090602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";CUTYPE=INDIVIDUAL;PARTSTAT
 =NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS=
 "1.2;Scheduling message has been delivered":mailto:wilfredo@e
 xample.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS="
 1.0;Scheduling message is pending":mailto:bernard@example.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE;SCHEDULE-STATUS="3.7;Invalid calendar user":m
 ailto:mike@example.org
END:VEVENT
END:VCALENDAR
```

B.2 Example: Attendee Receiving an Invitation

In the following example, Wilfredo's client retrieves and deletes the new scheduling message that appeared in his scheduling Inbox collection after the server automatically processed it and created a new scheduling object resource in his default calendar collection.

>> Request <<

```
GET /home/wilfredo/calendars/inbox/27d93fc0a58c.ics HTTP/1.1
Host: cal.example.com
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 18:59:58 GMT
Last-Modified: Tue, 02 Jun 2009 18:59:58 GMT
ETag: "da116714bc9926c89395895eb897deab"
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
METHOD:REQUEST
BEGIN:VEVENT
UID:9263504FD3AD
```

```
SEQUENCE:0
DTSTAMP:20090602T185254Z
DTSTART:20090602T160000Z
DTEND:20090602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";CUTYPE=INDIVIDUAL;PARTSTAT
 =NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:wilfredo@
 example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@ex
 ample.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE:mailto:mike@example.org
END:VEVENT
END:VCALENDAR
```

>> Request <<

```
DELETE /home/wilfredo/calendars/inbox/27d93fc0a58c.ics HTTP/1.1
Host: cal.example.com
```

>> Response <<

```
HTTP/1.1 204 No Content
Date: Tue, 02 Jun 2009 20:40:36 GMT
```

B.3 Example: Attendee Replying to an Invitation

In the following example, Wilfredo's accepts Cyrus's invitation and sets a reminder on the event.

>> Request <<

```
PUT /home/wilfredo/calendars/work/BB64861C2228.ics HTTP/1.1
Host: cal.example.com
If-Schedule-Tag-Match: "e78f23ed-0188-4bab-938d-2aeb3324c7e8"
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T185254Z
DTSTART:20090602T160000Z
DTEND:20090602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";CUTYPE=INDIVIDUAL;PARTSTAT
 =ACCEPTED;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:wilfredo@exam
 ple.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@ex
 ample.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE:mailto:mike@example.org
BEGIN:VALARM
```

```
TRIGGER:-PT15M
ACTION:DISPLAY
DESCRIPTION:Reminder
END:VALARM
END:VEVENT
END:VCALENDAR
```

>> Response <<

```
HTTP/1.1 200 OK
Content-Length: 0
Date: Tue, 02 Jun 2009 18:57:54 GMT
Last-Modified: Tue, 02 Jun 2009 18:57:54 GMT
ETag: "eb4639451b434fbd85561cfe74a4e785"
Schedule-Tag: "8893ee45-eb9d-428f-b53c-c777daf19e41"
```

Once the event modification has been completed, Wilfredo's client will retrieve the event back from the server to get the schedule status of the Organizer.

>> Request <<

```
PUT /home/wilfredo/calendars/work/BB64861C2228.ics HTTP/1.1
Host: cal.example.com
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 19:03:03 GMT
Last-Modified: Tue, 02 Jun 2009 19:02:21 GMT
ETag: "5eb897deabda116714bc9926c8939589"
Schedule-Tag: "8893ee45-eb9d-428f-b53c-c777daf19e41"
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T190221Z
DTSTART:20090602T160000Z
DTEND:20090602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Cyrus Daboo";SCHEDULE-STATUS="1.2;Scheduling mes
 sage has been delivered":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";CUTYPE=INDIVIDUAL;PARTSTAT
 =ACCEPTED;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:wilfredo@exam
 ple.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@ex
 ample.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE:mailto:mike@example.org
BEGIN:VALARM
TRIGGER:-PT15M
ACTION:DISPLAY
DESCRIPTION:Reminder
END:VALARM
END:VEVENT
END:VCALENDAR
```

B.4 Example: Organizer Receiving a Reply to an Invitation

On reception of Wilfredo's reply, Cyrus's server will automatically update Cyrus's scheduling object resource, make Wilfredo's scheduling message available in Cyrus's scheduling Inbox collection, and deliver an updated scheduling message to Bernard to share Wilfredo's updated participation status. In this example, Cyrus's client retrieves and deletes this scheduling message in his scheduling Inbox collection.

>> Request <<

```
GET /home/cyrus/calendars/inbox/c0a58c27d93f.ics HTTP/1.1
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 19:05:02 GMT
Last-Modified: Tue, 02 Jun 2009 19:04:20 GMT
ETag: "9265eb897deabc8939589da116714bc9"
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
METHOD:REPLY
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T185754Z
DTSTART:20090602T160000Z
DTEND:20090602T170000Z
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";PARTSTAT=ACCEPTED:mailto:w
 ilfredo@example.com
REQUEST-STATUS:2.0;Success
END:VEVENT
END:VCALENDAR
```

>> Request <<

```
DELETE /home/cyrus/calendars/inbox/c0a58c27d93f.ics HTTP/1.1
Host: cal.example.com
```

>> Response <<

```
HTTP/1.1 204 No Content
Date: Tue, 02 Jun 2009 19:05:05 GMT
```

Cyrus's client then retrieves the event back from the server with Wilfredo's updated participation status.

>> Request <<

```
GET /home/cyrus/calendars/work/9263504FD3AD.ics HTTP/1.1
Host: cal.example.com
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 19:05:02 GMT
Last-Modified: Tue, 02 Jun 2009 19:04:20 GMT
ETag: "eb897deabc8939589da116714bc99265"
Schedule-Tag: "132cab27-1fe3-67ab-de13-abd348d1dee3"
```

```
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T190420Z
DTSTART:20090602T160000Z
DTEND:20090602T170000Z
TRANSP:OPAQUE
SUMMARY:Lunch
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";CUTYPE=INDIVIDUAL;PARTSTAT
 =ACCEPTED;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS="2.0
 ;Reply has been received":mailto:wilfredo@example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS="
 1.0;Scheduling message is pending":mailto:bernard@example.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE;SCHEDULE-STATUS="3.7;Invalid calendar user":m
 ailto:mike@example.org
END:VEVENT
END:VCALENDAR
```

B.5 Example: Organizer Requesting Busy Time Information

In this example, Cyrus requests the busy time information of Wilfredo and Bernard.

>> Request <<

```
POST /home/cyrus/calendars/outbox/ HTTP/1.1
Host: cal.example.com
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
METHOD:REQUEST
BEGIN:VFREEBUSY
UID:4FD3AD926350
DTSTAMP:20090602T190420Z
DTSTART:20090602T000000Z
DTEND:20090604T000000Z
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega":mailto:wilfredo@example.com
ATTENDEE;CN="Bernard Desruisseaux":mailto:bernard@example.net
END:VFREEBUSY
END:VCALENDAR
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 20:07:34 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:schedule-response xmlns:D="DAV:"
       xmlns:C="urn:ietf:params:xml:ns:caldav">
<C:response>
<C:recipient>
<D:href>mailto:wilfredo@example.com<D:href>
```

```
</C:recipient>
<C:request-status>2.0;Success</C:request-status>
<C:calendar-data>BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
METHOD:REPLY
BEGIN:VFREEBUSY
UID:4FD3AD926350
DTSTAMP:20090602T200733Z
DTSTART:20090602T000000Z
DTEND:20090604T000000Z
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega":mailto:wilfredo@example.com
FREEBUSY;FBTYPE=BUSY:20090602T110000Z/20090602T120000Z
FREEBUSY;FBTYPE=BUSY:20090603T170000Z/20090603T180000Z
END:VFREEBUSY
END:VCALENDAR
</C:calendar-data>
</C:response>
<C:response>
<C:recipient>
<D:href>mailto:bernard@example.net<D:href>
</C:recipient>
<C:request-status>2.0;Success</C:request-status>
<C:calendar-data>BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Server//EN
METHOD:REPLY
BEGIN:VFREEBUSY
UID:4FD3AD926350
DTSTAMP:20090602T200733Z
DTSTART:20090602T000000Z
DTEND:20090604T000000Z
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Bernard Desruisseaux":mailto:bernard@example.net
FREEBUSY;FBTYPE=BUSY:20090602T150000Z/20090602T160000Z
FREEBUSY;FBTYPE=BUSY:20090603T090000Z/20090603T100000Z
FREEBUSY;FBTYPE=BUSY:20090603T180000Z/20090603T190000Z
END:VFREEBUSY
END:VCALENDAR
</C:calendar-data>
</C:response>
</C:schedule-response>
```

B.6 Example: User Attempting to Invite Attendee on behalf of Organizer

In the following example, Cyrus attempts to create, on behalf of Wilfredo, an event with Bernard specified as an Attendee. The request fails since Wilfredo didn't grant Cyrus the right to invite other Calendar Users on his behalf.

>> Request <<

```
PUT /home/wilfredo/calendars/work/def456.ics HTTP/1.1
Host: cal.example.com
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VEVENT
UID:3504F926D3AD
SEQUENCE:0
DTSTAMP:20090602T190221Z
DTSTART:20090602T230000Z
DTEND:20090603T000000Z
TRANSP:OPAQUE
SUMMARY:Dinner
```

```
ORGANIZER;CN="Wilfredo Sanchez Vega":mailto:wilfredo@example.com
ATTENDEE;CN="Wilfredo Sanchez Vega";CUTYPE=INDIVIDUAL;PARTSTAT=A
 CCEPTED:mailto:wilfredo@example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=NE
 EDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@exampl
 e.net
END:VEVENT
END:VCALENDAR
```

>> Response <<

```
HTTP/1.1 403 Forbidden
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<D:error xmlns:D="DAV:" xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:need-privileges>
    <D:resource>
      <D:href>/home/wilfredo/calendars/outbox/</D:href>
      <D:privilege><C:schedule-send-invite/></D:privilege>
    </D:resource>
  </D:need-privileges>
</D:error>
```

B.7 Example: Attendee Declining an Instance of a Recurring Event

In the following example, Bernard declines the second recurrence instance of a daily recurring event he's been invited to by Cyrus.

>> Request <<

```
PUT /home/bernard/calendars/work/4FD3AD926350.ics HTTP/1.1
Host: cal.example.com
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VTIMEZONE
TZID:America/Montreal
BEGIN:STANDARD
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
TZNAME:EST
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
TZNAME:EDT
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T185254Z
DTSTART;TZID=America/Montreal:20090601T150000
DTEND;TZID=America/Montreal:20090601T160000
RRULE:FREQ=DAILY;INTERVAL=1;COUNT=5
TRANSP:OPAQUE
SUMMARY:Review Internet-Draft
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
```

```
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 ACCEPTED;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@exampl
 e.net
END:VEVENT
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090603T183823Z
RECURRENCE-ID;TZID=America/Montreal:20090602T150000
DTSTART;TZID=America/Montreal:20090602T150000
DTEND;TZID=America/Montreal:20090602T160000
TRANSP:TRANSPARENT
SUMMARY:Review Internet-Draft
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 DECLINED;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@exampl
 e.net
END:VEVENT
END:VCALENDAR
```

>> Response <<

```
HTTP/1.1 200 OK
Content-Length: 0
Date: Tue, 02 Jun 2009 18:52:54 GMT
Last-Modified: Tue, 02 Jun 2009 18:52:54 GMT
ETag: "d85561cfe74a4e785eb4639451b434fb"
Schedule-Tag: "488177c8-2ea7-4176-a6cb-fab8cfccdea2"
```

Bernard's participation status update will cause his server to deliver a scheduling message to Cyrus. Cyrus's client will find the following reply message from Bernard in Cyrus's scheduling Inbox collection:

>> Request <<

```
GET /home/cyrus/calendars/inbox/9263504FD3AD.ics HTTP/1.1
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 18:52:58 GMT
Last-Modified: Tue, 02 Jun 2009 18:52:58 GMT
ETag: "eb897deabc8939589da116714bc99265"
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
METHOD:REPLY
BEGIN:VTIMEZONE
TZID:America/Montreal
BEGIN:STANDARD
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
TZNAME:EST
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
TZNAME:EDT
```

```
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090603T183823Z
RECURRENCE-ID;TZID=America/Montreal:20090602T150000
DTSTART;TZID=America/Montreal:20090602T150000
DTEND;TZID=America/Montreal:20090602T160000
SUMMARY:Review Internet-Draft
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Bernard Desruisseaux";PARTSTAT=DECLINED:
 mailto:bernard@example.net
REQUEST-STATUS:2.0;Success
END:VEVENT
END:VCALENDAR
```

B.8 Example: Attendee Removing an Instance of a Recurring Event

In the following example, Bernard removes from his calendar the third recurrence instance of a daily recurring event he's been invited to by Cyrus. This is accomplished by the addition of an "EXDATE" property to the scheduling object resource stored by Bernard.

>> Request <<

```
PUT /home/bernard/calendars/work/4FD3AD926350.ics HTTP/1.1
Host: cal.example.com
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VTIMEZONE
TZID:America/Montreal
BEGIN:STANDARD
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
TZNAME:EST
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
TZNAME:EDT
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090602T185254Z
DTSTART;TZID=America/Montreal:20090601T150000
DTEND;TZID=America/Montreal:20090601T160000
RRULE:FREQ=DAILY;INTERVAL=1;COUNT=5
EXDATE;TZID=America/Montreal:20090603T150000
TRANSP:OPAQUE
SUMMARY:Review Internet-Draft
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 ACCEPTED;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@exampl
 e.net
```

```
END:VEVENT
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090603T183823Z
RECURRENCE-ID;TZID=America/Montreal:20090602T150000
DTSTART;TZID=America/Montreal:20090602T150000
DTEND;TZID=America/Montreal:20090602T160000
TRANSP:TRANSPARENT
SUMMARY:Review Internet-Draft
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Cyrus Daboo";CUTYPE=INDIVIDUAL;PARTSTAT=ACCEPTED:
 mailto:cyrus@example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 DECLINED;ROLE=REQ-PARTICIPANT;RSVP=TRUE:mailto:bernard@exampl
 e.net
END:VEVENT
END:VCALENDAR
```

Bernard's deletion of a recurrence instance will cause his server to deliver a scheduling message to Cyrus. Cyrus's client will find the following reply message from Bernard in Cyrus's scheduling Inbox collection:

>> Request <<

```
GET /home/cyrus/calendars/inbox/6504923FD3AD.ics HTTP/1.1
```

>> Response <<

```
HTTP/1.1 200 OK
Date: Tue, 02 Jun 2009 18:52:58 GMT
Last-Modified: Tue, 02 Jun 2009 18:52:58 GMT
ETag: "eb897deabc8939589da116714bc99265"
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
METHOD:REPLY
BEGIN:VTIMEZONE
TZID:America/Montreal
BEGIN:STANDARD
DTSTART:20071104T020000
RRULE:FREQ=YEARLY;BYMONTH=11;BYDAY=1SU
TZNAME:EST
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:20070311T020000
RRULE:FREQ=YEARLY;BYMONTH=3;BYDAY=2SU
TZNAME:EDT
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
UID:9263504FD3AD
SEQUENCE:0
DTSTAMP:20090603T183823Z
RECURRENCE-ID;TZID=America/Montreal:20090603T150000
DTSTART;TZID=America/Montreal:20090603T150000
DTEND;TZID=America/Montreal:20090603T160000
SUMMARY:Review Internet-Draft
ORGANIZER;CN="Cyrus Daboo":mailto:cyrus@example.com
ATTENDEE;CN="Bernard Desruisseaux";PARTSTAT=DECLINED:
 mailto:bernard@example.net
REQUEST-STATUS:2.0;Success
```

```
END:VEVENT
END:VCALENDAR
```

C. Changes (to be removed by RFC Editor prior to publication)

C.1 Changes in -07

a. Restructured document.

b. Clarified that CALDAV:schedule-calendar-transp only applies to calendar collection.

c. Removed CALDAV:schedule-state property on scheduling messages in the scheduling Inbox collection.

d. Added conditional requests on scheduling object resources.

e. Added section on handling of PARTSTAT.

f. Added SCHEDULE-FORCE-SEND iCalendar property parameter.

g. Added clarification on child resources in scheduling Outbox collections.

h. Clarified Attendee changes that server MUST allow, and removed restrictions on changes that Attendee MUST NOT do.

i. Added Example Scheduling Transactions appendix.

j. Scheduling privileges are no longer required to be non-abstract.

k. Removed handling of REFRESH requests.

l. Removed handling of VJOURNAL components.

m. Completed IANA Considerations section.

n. Added references to RFC3283 and RFC5234.

o. Updated references to iCalendar, iTIP and iMIP.

C.2 Changes in -06

a. Removed distinction between scheduling calendar collections and basic calendar collections - now just have calendar collections.

b. Clients now "MAY" reload data rather than "SHOULD" reload data.

c. Fixed <C:recipient> in examples.

d. Removed CALDAV:attachments-allowed precondition on POST to Outbox as that is no longer relevant.

e. Added CALDAV:default-calendar-delete-allowed precondition for DELETE.

f. Relaxed MUST->MAY for Organizer setting PARTSTAT value.

g. Tweaked restrictions on Create/Modify to emphasize that 4791 restrictions also apply.

h. Added comment that 'opaque' is the default when the CALDAV:schedule-calendar-transp property is not present.

i. Description of Schedule-Reply header changed to reflect that it is only relevant for Attendees.

j. Minor typos fixed.

C.3 Changes in -05

This draft has changed substantially since the -04 version. The primary reason for this change was implementation experience from a number of vendors who implemented products based on the earlier drafts. Experience showed that the client/server interaction was not reliable in keeping scheduling messages synchronized between organizer and attendees. In addition the latency in updates due to clients being offline proved unacceptable to users. These issues led to the redesign of this specification to support a server-based processing model that eliminates all the problems seen previously. Whilst this adds significant complexity to the server in that it needs to be a full blown iTIP processing agent, it does remove a lot of the same complexity from clients, opening up the possibility of supporting complex scheduling behaviors even with "thin" clients.

In the judgement of the authors, we consider this new specification to be a substantial improvement over the old one and believe it represents a stronger protocol that will lead to better interoperability.

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

[1] mailto:ietf-ipr@ietf.org