Internet Engineering Task Force (IETF)
Request for Comments: 6638
Updates: 4791, 5546
Category: Standards Track
ISSN: 2070-1721

C. Daboo
Apple Inc.
B. Desruisseaux
Oracle

June 2012

Scheduling Extensions to CalDAV

Abstract

This document defines extensions to the Calendaring Extensions to WebDAV (CalDAV) "calendar-access" feature to specify a standard way of performing scheduling operations with iCalendar-based calendar components. This document defines the "calendar-auto-schedule" feature of CalDAV.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741[1].

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6638[2].

Copyright Notice

[1] https://www.rfc-editor.org/rfc/rfc5741.html#section-2
[2] http://www.rfc-editor.org/info/rfc6638
[3] http://trustee.ietf.org/license-info

Table of Contents

1. Introduction

This document specifies extensions to the CalDAV "calendar-access" [RFC4791] feature to enable scheduling of iCalendar-based [RFC5545] calendar components between calendar users.

This extension leverages the scheduling methods defined in the iCalendar Transport-independent Interoperability Protocol (iTIP) [RFC5546] to permit calendar users to perform scheduling operations such as schedule, reschedule, respond to scheduling request, or cancel calendar components, as well as search for busy time information. However, the following iTIP [RFC5546] features are not covered: publishing, countering, delegating, refreshing, and forwarding calendar components, as well as replacing the "Organizer" of a calendar component. It is expected that future extensions will be developed to address these.

This specification defines a client/server scheduling protocol, where the server is made responsible for sending scheduling messages and processing incoming scheduling messages. The client operations of creating, modifying, or deleting a calendar component in a calendar are enough to trigger the server to deliver the necessary scheduling messages to the appropriate calendar users. This approach is sometimes referred to as "implicit scheduling".

This specification only addresses how scheduling occurs with users on a single system (i.e., scheduling between CalDAV servers, or some other calendaring and scheduling system, is not defined). However, this specification is compatible with servers being able to send or receive scheduling messages with "external" users (e.g., using the iCalendar Message-Based Interoperability Protocol (iMIP) [RFC6047]).

Section 3 defines the automated "Scheduling Operations" that allow a client to store iCalendar data on a CalDAV server, with the server taking specific actions in response. One of three scheduling operations can take place -- "create", "modify", or "remove", based on the HTTP method used for the request -- in addition to a comparison between any existing and any new iCalendar data.

Section 4 defines how the server processes scheduling messages sent as the result of a scheduling operation.

Section 5 defines how freebusy requests with an immediate response are accomplished.

Section 6 defines access control privileges for the scheduling operations defined in this specification.

For the majority of the following discussion, scheduling of events will be discussed. However, scheduling of to-dos is also fully supported by this specification.

This specification has been under development for a number of years, and most current implementations of CalDAV support it. With the publication of this document, it is expected that all new CalDAV implementations will support it by default. Interoperability tests have been performed regularly. Significant issues with incompatible CalDAV implementations are not anticipated.

1.1. Terminology

This specification reuses much of the same terminology as iCalendar [RFC5545], iTIP [RFC5546], WebDAV [RFC4918], and CalDAV [RFC4791]. Additional terms used by this specification are as follows:

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 the "Organizer".
Attendee scheduling object resource:	A scheduling object resource owned by an "Attendee".
Scheduling operation:	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 that describes a scheduling operation 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. Notational Conventions

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].

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

The 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 term "protected" is used in the Conformance field of WebDAV property definitions as defined in Section 15 of [RFC4918].

Calendaring and scheduling roles are referred to in quoted-strings of text with the first character of each word in uppercase. For example, "Organizer" refers to a role of a calendar user within the scheduling protocol defined by [RFC5546].

1.3. XML Namespaces and Processing

This document uses XML DTD fragments ([W3C.REC-xml-20081126], as a purely notational convention. WebDAV request and response bodies cannot be validated by a DTD due to the specific extensibility rules defined in Section 17 of [RFC4918] and due to the fact that all XML elements defined by that specification use the XML namespace name "DAV:". In particular,

1. element names use the "DAV:" namespace,
2. element ordering is irrelevant unless explicitly stated,
3. extension elements (elements not already defined as valid child elements) can be added anywhere, except when explicitly stated otherwise, and
4. extension attributes (attributes not already defined as valid for this element) can be added anywhere, except when explicitly stated otherwise.

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

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 strings "DAV:" and "CALDAV:" will be prefixed to the element types, respectively.

This document inherits, and sometimes extends, DTD productions from Section 14 of [RFC4918].

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

2. Scheduling Support

A server that supports the features described in this document is REQUIRED to support the CalDAV "calendar-access" [RFC4791] feature. Servers include "calendar-auto-schedule" as a field in the DAV response header from an OPTIONS request on any resource that supports any scheduling operations, properties, privileges, or methods.

This specification introduces new collection resource types that are used to manage scheduling object resources, and scheduling privileges (as per Section 6), as well as provide scheduling functionality. It is the server's responsibility to create these collection resources, and clients have no way to create or delete them.

2.1. Scheduling Outbox Collection

A scheduling Outbox collection is used as the target for busy time information requests, and to manage privileges that apply to outgoing scheduling 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>
```

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 and apply to scheduling messages submitted to the scheduling Outbox collection with the POST method:

- CALDAV:supported-calendar-component-set
- CALDAV:supported-calendar-data
- CALDAV:max-resource-size
- CALDAV:min-date-time
- CALDAV:max-date-time
- CALDAV:max-attendees-per-instance

The use of child resources in a scheduling Outbox collection is reserved for future revisions or extensions of this specification.

The following WebDAV property is defined on principal resources and used to locate the corresponding Outbox collection for the associated principal.

2.1.1. 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 DAV: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. If not present, then the associated calendar user is not enabled for the sending of scheduling messages on the server.
Definition:	`<!ELEMENT schedule-outbox-URL (DAV:href)>`

2.2. Scheduling Inbox Collection

A scheduling Inbox collection contains copies of incoming scheduling messages. These can be requests sent by an "Organizer", or replies sent by an "Attendee" in response to a request. The scheduling Inbox collection is also used to manage scheduling privileges.

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 [RFC5546]. 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., Unique Identifier ("UID") uniqueness) do not apply to calendar object resources contained in a scheduling Inbox collection. Thus, multiple calendar object resources contained in a scheduling Inbox collection can have the same "UID" property value (i.e., multiple scheduling messages for the same calendar component).

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 and apply to scheduling messages delivered to the collection:

- CALDAV:supported-calendar-component-set
- CALDAV:supported-calendar-data
- CALDAV:max-resource-size
- CALDAV:min-date-time
- CALDAV:max-date-time
- CALDAV:max-instances
- CALDAV:max-attendees-per-instance
- CALDAV:calendar-timezone

The following WebDAV property is defined on principal resources and used to locate the corresponding Inbox collection for the associated principal.

2.2.1. 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 DAV: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 allows a client to determine where the scheduling Inbox collection of the current user is located so that processing of scheduling messages can occur. If not present, then the associated calendar user is not enabled for reception of scheduling messages on the server.
Definition:	`<!ELEMENT schedule-inbox-URL (DAV:href)>`

2.3. 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 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 [RFC5546]) 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.

2.4. Additional Principal Properties

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

2.4.1. 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 DAV: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 his calendar user address, the URI of his 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. If not present, then the associated calendar user is not enabled for scheduling on the server.

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>
```

2.4.2. 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 [RFC5545].

Protected: This property MAY be protected.

PROPFIND behavior: This property SHOULD NOT be returned by a PROPFIND DAV: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: Clients can query principal resources in order to look up "Attendees" available on the server. When doing this, it is useful to know, or restrict the query to, certain types of calendar users (e.g., only search for "people", or only search for "rooms"). This property MAY be defined on principal resources to indicate the type of calendar user associated with the 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>
```

3. Scheduling Operations

When a calendar object resource is created, modified, or removed from a calendar collection, 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 operations can occur in this case, equivalent to iTIP "REQUEST", "REPLY", "CANCEL", and "ADD" operations.

3.1. Identifying Scheduling Object Resources

Calendar object resources on which the server performs scheduling operations are referred 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 if at least one of the "ATTENDEE" iCalendar property values matches 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 created in an appropriate calendar collection (see Section 4.3).

3.2. 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.

3.2.1. Organizer Scheduling Object Resources

An "Organizer" can create, modify, or remove a scheduling object resource, subject to access privileges, preconditions, and the restrictions defined in Section 4.1 of [RFC4791]. These operations are each described next, and how they are invoked via HTTP requests is described in Section 3.2.3.

The "Organizer" of a calendar component can also be an "Attendee" of that calendar component. In such cases, the server MUST NOT send a scheduling message to the "Attendee" that matches the "Organizer".

The server SHOULD 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".

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".

3.2.1.1. Create

When an "Organizer" creates a scheduling object resource, the server MUST inspect each "ATTENDEE" property to determine whether to send a scheduling message. The table below indicates the appropriate iTIP method used by the server, taking into account any "SCHEDULE-AGENT" property parameter (see Section 7.1) specified on each "ATTENDEE" property.

SCHEDULE-AGENT	iTIP METHOD
SERVER (default)	REQUEST
CLIENT	--

SCHEDULE-AGENT	iTIP METHOD
NONE	--

"SCHEDULE-STATUS" iCalendar property parameters are added or changed on "ATTENDEE" iCalendar properties in the scheduling object resource being created as described in Section 7.3, with the value set as described in Section 3.2.9. 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 in order to update to the new server-generated state information.

The server MUST add a "SCHEDULE-STATUS" iCalendar property parameter (see Section 7.3) to the "ATTENDEE" iCalendar property in the scheduling object resource being created, and set its value as described in Section 3.2.9. 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 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.

The server MUST return an error with the CALDAV:allowed-organizer-scheduling-object-change precondition code (Section 3.2.4.3) when the "Organizer" attempts to change the iCalendar data in a manner that is forbidden.

3.2.1.2. Modify

When an "Organizer" modifies a scheduling object resource, the server MUST inspect each "ATTENDEE" property in both the original and modified iCalendar data on a per-instance basis to determine whether to send a scheduling message. The table below indicates the appropriate iTIP method used by the server, taking into account any "SCHEDULE-AGENT" property parameter (see Section 7.1) specified on each "ATTENDEE" property. 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 (Original) or is being removed (Modified).

		Modified			
		<Removed>	SERVER (default)	CLIENT	NONE
Original	<Absent>	--	REQUEST / ADD	--	--
	SERVER (default)	CANCEL	REQUEST	CANCEL	CANCEL
	CLIENT	--	REQUEST / ADD	--	--
	NONE	--	REQUEST / ADD	--	--

"SCHEDULE-STATUS" iCalendar property parameters are added or changed on "ATTENDEE" iCalendar properties in the scheduling object resource being modified as described in Section 7.3, with the value set as described in Section 3.2.9. 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 in order to update to the new server-generated state information.

The server MUST return an error with the CALDAV:allowed-organizer-scheduling-object-change precondition code (Section 3.2.4.3) when the "Organizer" attempts to change the iCalendar data in a manner that is forbidden.

3.2.1.3. Remove

When an "Organizer" removes a scheduling object resource, the server MUST inspect each "ATTENDEE" property to determine whether to send a scheduling message. The table below indicates the appropriate iTIP method used by the server, taking into account any "SCHEDULE-AGENT" property parameter (see Section 7.1) specified on each "ATTENDEE" property.

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

3.2.2. Attendee Scheduling Object Resources

An "Attendee" can create, modify, or remove a scheduling object resource. These operations are each described next, and how they are invoked via HTTP requests is described in Section 3.2.3.

3.2.2.1. 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".

Servers MUST allow "Attendees" to make the following iCalendar data changes, subject to other restrictions, such as access privileges and preconditions:

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 "COMPLETED" iCalendar properties.
5. add, modify, or remove any "VALARM" iCalendar components.
6. add, modify, or remove the "CALSCALE" iCalendar property within the top-level "VCALENDAR" component.
7. modify the "PRODID" iCalendar property within the top-level "VCALENDAR" component.
8. add "EXDATE" iCalendar properties and possibly remove components for overridden recurrence instances.
9. add, modify, or remove any "CREATED", "DTSTAMP", and "LAST-MODIFIED" iCalendar properties.
10. add, modify, or remove "SCHEDULE-STATUS" iCalendar property parameters on "ATTENDEE" properties that have a "SCHEDULE-AGENT" parameter set to "CLIENT".
11. add new components to represent overridden recurrence instances, provided the only changes to the recurrence instance follow the rules above.

The server MUST return an error with the CALDAV:allowed-attendee-scheduling-object-change precondition code (Section 3.2.4.4) when the "Attendee" attempts to change the iCalendar data in a manner forbidden by the server.

3.2.2.2. Create

Typically, an "Attendee" does not create scheduling object resources, as scheduling messages delivered to him on the server are automatically processed by the server and placed on one of his calendars (see Section 4). However, in some cases, a scheduling message can get delivered directly to the client (e.g., via email [RFC6047]), and the "Attendee" might wish to store that on the server. In that case, the client creates a scheduling object resource in a calendar belonging to the "Attendee". It can then set the "SCHEDULE-AGENT" iCalendar property parameter on all "ORGANIZER" iCalendar properties in the resource to

determine how the server treats the resource. The value of the "SCHEDULE-AGENT" iCalendar property parameter on all "ORGANIZER" iCalendar properties MUST be the same.

SCHEDULE-AGENT	Action
SERVER (default)	The server will attempt to process changes to the resource using the normal rules for attendee scheduling object resources.
CLIENT	The server does no special processing of the resource. client is assumed to be handling "Attendee" replies, etc.
NONE	The server does no special processing of the resource.

"SCHEDULE-STATUS" iCalendar property parameters are added or changed on "ORGANIZER" iCalendar properties in the scheduling object resource being created as described in Section 7.3, with the value set as described in Section 3.2.9.

3.2.2.3. Modify

When a scheduling object resource is modified by an "Attendee", the server's behavior depends on the value of the "SCHEDULE-AGENT" iCalendar property parameter on the "ORGANIZER" iCalendar properties:

SCHEDULE-AGENT	Action
SERVER (default)	The server will attempt to process the update using the behavior listed below.
CLIENT	The server does no special processing of the resource. client is assumed to be handling any "Attendee" replies, etc.
NONE	The server does no special processing of the resource.

The server will inspect the changes by comparing the new scheduling object resource with the existing scheduling object resource.

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".

If the "Attendee" adds an "EXDATE" property value to effectively remove a recurrence instance, the server MUST deliver an iTIP "REPLY" scheduling message to the "Organizer" to indicate that the "Attendee" has declined the instance.

"SCHEDULE-STATUS" iCalendar property parameters are added or changed on "ORGANIZER" iCalendar properties in the scheduling object resource being modified as described in Section 7.3, with the value set as described in Section 3.2.9. This will result in the updated 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 in order to update to the new server-generated state information.

3.2.2.4. Remove

When a scheduling object resource is removed by an "Attendee", the server's behavior depends on the value of the "SCHEDULE-AGENT" iCalendar property parameter on the "ORGANIZER" iCalendar properties:

SCHEDULE-AGENT	Action
SERVER (default)	The server will attempt to process the removal, taking into account any "Schedule-Reply" request header as per Section 8.1.

SCHEDULE-AGENT	Action
CLIENT	The server does no special processing of the resource. client is assumed to be handling any "Attendee" replies, etc.
NONE	The server does no special processing of the resource.

3.2.3. HTTP Methods

This section describes how the use of various HTTP [RFC2616] and WebDAV [RFC4918] methods on a scheduling object resource will cause a create, modify, or remove operation 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.

3.2.3.1. PUT

When the server receives a PUT method request, it MUST execute the following operations, provided all appropriate preconditions are met:

Existing Destination Resource	Resulting Destination Resource	Server Operation
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

3.2.3.2. DELETE

When the server receives a DELETE method request targeted at a scheduling object resource, it MUST execute the Remove operation.

When the server receives a DELETE method request targeted at a calendar collection, it MUST execute the Remove operation on all scheduling object resources contained in the calendar collection.

3.2.3.3. COPY

When the server receives a COPY method request, it MUST execute the following operations based on the source and destination collections in the request:

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

Note (1): The rules in Section 3.2.3.1 are applied for the destination of the COPY request.

Note (2): The server MAY reject this as per Section 3.2.4.1; otherwise, None.

The behavior of a COPY method request on a calendar collection is undefined.

3.2.3.4. MOVE

When the server receives a MOVE method request, it MUST execute the following operations based on the source and destination collections in the request:

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

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

Note (1): The rules in Section 3.2.3.1 are applied for the destination of the MOVE request.

Note (2): The rules in Section 3.2.3.2 are applied for the source of the MOVE request.

The behavior of a MOVE method request on a calendar collection is undefined.

3.2.4. Additional Method Preconditions

This specification defines method preconditions (see Section 16 of WebDAV [RFC4918]), in addition to those in [RFC4791], to provide machine-parseable information in error responses.

3.2.4.1. 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: 403 Forbidden

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>
```

3.2.4.2. 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: 403 Forbidden

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>
```

3.2.4.3. 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: 403 Forbidden

Purpose: (precondition) -- Servers MAY impose restrictions on modifications allowed by an "Organizer". For instance, servers MAY prevent the "Organizer" from 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 does not have the "SCHEDULE-AGENT" property parameter. See Section 3.2.1.

Definition:

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

3.2.4.4. 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:	403 Forbidden
Purpose:	(precondition) -- Servers MAY impose restrictions on modifications allowed by an "Attendee", subject to the allowed changes specified in Section 3.2.2.1.

Definition:

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

3.2.5. DTSTAMP and SEQUENCE Properties

The server MUST ensure that a "DTSTAMP" iCalendar property is present and set the value to the UTC time that the scheduling message was generated (as required by iCalendar).

The server MUST ensure that for each type of scheduling operation, the "SEQUENCE" iCalendar property value is updated as per iTIP [RFC5546].

3.2.6. Restrict Recurrence Instances Sent to "Attendees"

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

For example, if an "Attendee" is invited to only a single instance of a recurring event, the organizer scheduling object resource 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. Any scheduling messages 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 organizer scheduling object resource will include an overridden instance with an "ATTENDEE" list that does not include the "Attendee" being excluded. Any scheduling messages delivered to the "Attendee" will not specify the overridden instance but rather will include an "EXDATE" property in the "master" component that defines the recurrence set.

3.2.7. Forcing the Server to Send a Scheduling Message

The iCalendar property parameter "SCHEDULE-FORCE-SEND", defined in Section 7.2, 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 his participation status without being forced to modify the event.

3.2.8. "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.

A reschedule 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. Servers MUST reset the "PARTSTAT" property parameter value of all "ATTENDEE" properties, except the one that corresponds to the "Organizer", to "NEEDS-ACTION" for each calendar component change that causes any instance to be rescheduled.

Property	Server Action
DTSTART, DTEND, DURATION, DUE	Any change to these properties results 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 results 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 results in "PARTSTAT" being reset to "NEEDS-ACTION" on each affected component.
EXDATE	A change to or removal of this property that results in the reinstatement of recurring instances results in "PARTSTAT" being set 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 might be impacted by the change).

3.2.9. Schedule Status Values

When scheduling with an "Attendee", there are two types of status information that can be returned during the operation. The first type of 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 type of 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 7.3). The status code value for "delivery" status can be one of the following:

Delivery Status Code	Description
1.0	The scheduling message is pending. is, 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. state can occur with "store and forward" style scheduling protocols such as iMIP [RFC6047] (iTIP using email).

Delivery Status Code	Description
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. that this code applies to both "Organizer" and "Attendee" calendar user addresses.
3.8	The scheduling message was not delivered due to insufficient privileges. that this code applies to privileges granted by both the "Organizer" and "Attendee" calendar users.
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 way to deliver the message. is likely a permanent failure, and the originator ought 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. is likely a permanent failure, and the originator ought not try to send the message again.

The status code for "reply" status can be any of the valid iTIP [RFC5546] "REQUEST-STATUS" values.

The 1.xx "REQUEST-STATUS" codes are new. This specification modifies item (2) of Section 3.6 of [RFC5546] by adding the following restriction:

> For a 1.xx code, all components MUST have exactly the same code.

Definition of the new 1.xx codes is as follows:

3.2.9.1. Status Code 1.0

Status Code:	1.0
Status Description:	Pending.
Status Exception Data:	None.
Description:	Delivery of the iTIP message is pending.

3.2.9.2. Status Code 1.1

Status Code:	1.1
Status Description:	Sent.
Status Exception Data:	None.
Description:	The iTIP message has been sent, though no information about successful delivery is known.

3.2.9.3. Status Code 1.2

Status Code:	1.2

Status Description:	Delivered.
Status Exception Data:	None.
Description:	The iTIP message has been sent and delivered.

3.2.10. Avoiding Conflicts when Updating Scheduling Object Resources

Scheduling object resources on the server might change frequently as "Attendees" change their participation status, triggering updates to the "Organizer", and refreshes of other "Attendees"' copies of the scheduling object resource. This can lead to an "inconsequential" change to a calendar user's data -- one that does not directly impact the user's own participation status. When this occurs, clients have to reload calendar data and reconcile with changes being made by calendar users. To avoid the need for this, the server can instead merge calendar data changes from a client with changes made as a result of a scheduling operation carried out by some other calendar user.

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 (one that simply updates the status information of "Attendees" due to a reply from another "Attendee").

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.

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 destination resource.

Clients SHOULD use the If-Schedule-Tag-Match header on requests that update scheduling object resources, instead of HTTP ETag-based precondition tests (e.g., If-Match). Normal ETag-based precondition tests 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 the CALDAV:schedule-tag property value when the scheduling 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 specifies 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 would remain the same.

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).

3.2.10.1. PUT

Clients MAY use the If-Schedule-Tag-Match request header to do a 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.

3.2.10.2. DELETE, COPY, or MOVE

Clients MAY use the If-Schedule-Tag-Match request header to do a DELETE, COPY, or MOVE 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, COPY, or MOVE request processing for the resource. Otherwise, the server MUST fail the request with a 412 Precondition Failed status code.

4. Processing Incoming Scheduling Messages

Scheduling operations can cause the delivery of a scheduling message into an "Organizer's" or "Attendee's" scheduling Inbox collection. Servers MUST automatically process incoming scheduling messages using the rules defined by [RFC5546], by creating or updating the corresponding scheduling object resources on calendars owned by the owner of the scheduling Inbox collection. In addition, the scheduling message is stored in the scheduling Inbox collection as an indicator to the client that a scheduling operation has taken place. Scheduling messages are typically removed from the scheduling Inbox collection by the client once the calendar user has acknowledged the change.

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 (i.e., a schedule message from another user can cause modifications to resources in calendar collections that the other user would not normally have read or write access to). Additionally, servers MUST take into account any scheduling Inbox collection preconditions (see Section 2.2) when delivering the scheduling message, and MUST take into account the similar preconditions on any calendar collection that contains, or would contain, the corresponding scheduling object resource.

4.1. 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 processes the scheduling message and creates a new scheduling object resource as per Section 4.3.

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

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

4.2. 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". If the corresponding scheduling object resource cannot be found, the server SHOULD ignore the scheduling message.

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 instances of a recurring event).

The server MUST also update or add the "SCHEDULE-STATUS" property parameter on each matching "ATTENDEE" iCalendar property and set 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 3.2.6.

The scheduling message MUST only appear in the "Organizer's" scheduling Inbox collection once all automatic processing has been done.

4.3. Default Calendar Collection

The server processes scheduling messages received for an "Attendee" by creating a new scheduling object resource in a calendar collection belonging to the "Attendee", when one does not already exist. A calendar user that is 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. The CALDAV:schedule-default-calendar-URL WebDAV property, which can be present on the scheduling Inbox collection of a calendar user, specifies whether this calendar user has a default calendar collection. See Section 9.2.

Servers SHOULD 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, the server MUST ensure that any new value 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.

4.3.1. Additional Method Preconditions

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

4.3.1.1. CALDAV:default-calendar-needed Precondition

Name:	default-calendar-needed
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-needed EMPTY>`

4.3.1.2. 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 do not 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>`

5. Request for Busy Time Information

Busy time information of one or more calendar users can be determined by submitting a POST request targeted at the scheduling Outbox collection of the calendar user requesting the information (the "Organizer"). To accomplish this, the request body MUST contain a "VFREEBUSY" calendar component with the "METHOD" iCalendar property set to the value "REQUEST" as specified in Section 3.3.2 of iTIP [RFC5546]. The resource identified by the Request-URI MUST be a resource collection of type CALDAV:schedule-outbox (Section 2.1). The "ORGANIZER" property value in the "VFREEBUSY" component MUST match one of the calendar user addresses of the owner of the Outbox collection.

A response to a busy time request that indicates status for one or more calendar users MUST be an XML document with a CALDAV:schedule-response XML element as its root element. This element MUST contain one CALDAV:response element for each calendar user, with each such element in turn containing elements that indicate which calendar user they correspond to, the scheduling status for that calendar user, any error codes, and an optional description. For a successful busy time request, a CALDAV:calendar-data element is also present for each calendar user, containing the actual busy time information (i.e., an iCalendar "VFREEBUSY" component). See Section 10 for details on the child elements. See Appendix B.5 for an example busy time request and response.

5.1. Status Codes

The list below summarizes the most common status codes used for this method. However, clients need to be prepared to handle other 2/3/4/5xx series status codes as well.

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.

5.2. Additional Method Preconditions

The following are existing preconditions that are reused for the POST method on an Outbox collection.

- DAV:need-privileges [RFC3744]
- CALDAV:supported-calendar-data [RFC4791]
- CALDAV:valid-calendar-data [RFC4791]
- CALDAV:max-resource-size [RFC4791]

The following are new method preconditions for the POST method on an Outbox collection.

5.2.1. 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 the restrictions specified in Section 3.3.2 of iTIP [RFC5546].
Definition:	`<!ELEMENT valid-scheduling-message EMPTY>`

5.2.2. CALDAV:valid-organizer Precondition

Name:	valid-organizer
Namespace:	urn:ietf:params:xml:ns:caldav
Apply to:	POST
Use with:	403 Forbidden
Purpose:	(precondition) -- The "ORGANIZER" property value in the POST request's scheduling message MUST match one of the calendar user addresses of the owner of the scheduling Outbox collection being targeted by the request.
Definition:	`<!ELEMENT valid-organizer EMPTY>`

6. Scheduling Privileges

New scheduling privileges are 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.

6.1. Privileges on Scheduling Inbox Collections

This section defines new WebDAV Access Control List (ACL) [RFC3744] privileges that are defined 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.

6.1.1. CALDAV:schedule-deliver Privilege

CALDAV:schedule-deliver is an aggregate privilege as per Section 6.3.

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

6.1.2. 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>
```

6.1.3. 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>
```

6.1.4. 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>
```

6.2. Privileges on Scheduling Outbox Collections

This section defines new WebDAV ACL [RFC3744] 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 (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.

6.2.1. CALDAV:schedule-send Privilege

CALDAV:schedule-send is an aggregate privilege as per Section 6.3.

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

6.2.2. 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 the 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 scheduling messages to "Attendees".

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

6.2.3. CALDAV:schedule-send-reply Privilege

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

Users granted the DAV:bind privilege on a calendar collection, or the 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 scheduling message replies to the "Organizer".

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

6.2.4. 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>
```

6.3. Aggregation of Scheduling Privileges

Server implementations MUST aggregate the scheduling privileges as follows:

DAV:all contains CALDAV:schedule-deliver and CALDAV:schedule-send;

CALDAV:schedule-deliver contains CALDAV:schedule-deliver-invite, CALDAV:schedule-deliver-reply, and CALDAV:schedule-query-freebusy;

CALDAV:schedule-send contains CALDAV:schedule-send-invite, CALDAV:schedule-send-reply, and CALDAV:schedule-send-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]
```

7. Additional iCalendar Property Parameters

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

7.1. 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 scheduling
                      / x-name        ; Experimental type
                      / iana-token)   ; Other IANA-
registered type
  ;
  ; If the parameter is not present, its value defaults
 to SERVER.
  ; "x-name" and "iana-token" are defined in Section 3.1
 of
  ; [RFC5545].
```

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 a scheduling operation 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 a scheduling operation. 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 a scheduling operation.

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

The parameter value MUST be the same on every "ORGANIZER" property in a scheduling object resource.

The parameter value MUST be the same on each "ATTENDEE" property whose values match in a scheduling object resource.

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

Example:

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

7.2. 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-token" is defined in Section 3.1 of [RFC5545].
  Its value
  ; MUST be an IANA-registered iCalendar "METHOD"
 property value.
```

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 a scheduling operation.

Clients MUST 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 [RFC5546]) when the "SCHEDULE-FORCE-SEND" parameter is set to an iana-token value they do not recognize.

Example:

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

7.3. 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
[RFC5545].  The
  ; 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 only add or change this property parameter on any "ATTENDEE" properties corresponding to calendar users who were sent a scheduling message via a scheduling operation. 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 a scheduling operation. 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 a scheduling operation.

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

Values for this property parameter are described in Section 3.2.9.

Example:

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

8. Additional Message Header Fields

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

8.1. Schedule-Reply Request Header

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

Example:

```
Schedule-Reply: F
```

When an "Attendee" removes a scheduling object resource as per Section 3.2.2.4, and the Schedule-Reply header is set to the value "T" (true) or is not present, 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 scheduling operation processing.

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

The Schedule-Reply request header is used by a client to indicate to a server whether or not a scheduling operation ought to 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 removal. There are situations in which unsolicited scheduling messages need to be silently removed (or ignored) for security or privacy reasons. This request header allows the scheduling object resource to be removed if such a need arises.

8.2. 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 3.2.10.

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"
```

8.3. 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"
```

9. Additional WebDAV Properties

This specification defines the following new WebDAV properties for use with CalDAV.

9.1. 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 busy time information.
Protected:	This property MAY be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 14.2 of [RFC4918]).
COPY/MOVE behavior:	This property value SHOULD be kept during a MOVE operation, and SHOULD be copied and preserved 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 ought to contribute to the owner's busy time. When the CALDAV:opaque element is used, all calendar object resources in the corresponding calendar collection MUST contribute to busy time, assuming that 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 busy time. 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>`
	`<!-- Affects 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>`

9.2. 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 that 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 4.3.
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>`

9.3. 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 value is determined by the server and MAY be different from the value on the source resource.
Description:	The CALDAV:schedule-tag property MUST be defined on all scheduling object resources. This property is described in Section 3.2.10.
Definition:	`<!ELEMENT schedule-tag (#PCDATA)>`

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

10. XML Element Definitions

10.1. 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 5.
Definition:	

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

10.2. 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 5.
Definition:	

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

<!-- CALDAV:calendar-data is defined in Section 9.6 of
RFC 4791 and when used here uses the definition with
content (#PCDATA) only. -->
```

10.3. 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 5.
Definition:	

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

10.4. 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 5.
Definition:	

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

11. 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 Transport Layer Security (TLS) as defined in [RFC2818] for all scheduling operations. Clients MUST use the procedures detailed in Section 6 of [RFC6125] to verify the authenticity of the server. Servers MUST make use of HTTP authentication [RFC2617] to verify the authenticity of the calendar user for whom the client is sending requests.

11.1. Preventing Denial-of-Service Attacks

Servers MUST ensure that clients cannot consume excessive server resources by carrying out "large" scheduling operations. In particular, servers SHOULD enforce CALDAV:max-resource-size, CALDAV:max-instances, and CALDAV:max-attendees-per-instance preconditions as applicable for scheduling Inbox and Outbox collections.

11.2. Verifying Scheduling Operations

When handling a scheduling operation:

1. 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.
2. 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 of the DAV:owner of the calendar collection in which a scheduling object resource is being manipulated.
3. Servers MUST only deliver scheduling messages to 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 calendar collection in which a scheduling object resource is being manipulated.
4. 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.
5. 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.

11.3. Verifying Busy Time Information Requests

When handling a POST request on a scheduling Outbox collection:

1. 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.
2. 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.
3. 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.

11.4. Privacy Issues

This specification only defines how calendar users on the same server are able to schedule with each other -- unauthenticated users have no way to carry out scheduling operations. Access control privileges (as per Section 6) can control which of those users can schedule with others. Calendar users not wishing to expose their calendar information to other users can do so by denying privileges to specific users, or all users, for all scheduling operations, or perhaps only freebusy.

"Attendees" can also use the Schedule-Reply request header (Section 8.1) 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".

Servers MUST NOT expose any private iCalendar data, or WebDAV resource state information (URLs, WebDAV properties, etc.) for one calendar user to another via scheduling messages or error responses to scheduling operations. In particular, as per Section 8.1 of [RFC4918], authorization errors MUST take preference over other errors.

11.5. Mitigation of iTIP Threats

Section 6.1 of iTIP [RFC5546] defines a set of potential threats in a scheduling system, and Section 6.2 of [RFC5546] defines recommendations on how those can be addressed in protocols using iTIP. This specification addresses the iTIP threats in the following manner:

Spoofing the "Organizer":	Addressed by item 4 in Section 11.2.
Spoofing the "Attendee":	Addressed by Section 3.2.2.1 and item 2 in Section 11.2.
Unauthorized Replacement of the "Organizer":	Addressed by item 5 in Section 11.2.
Eavesdropping and Data Integrity:	Addressed by requiring TLS.
Flooding a Calendar:	Addressed by requirements in Section 11.1.
Unauthorized REFRESH Requests:	This specification does not support the REFRESH method.

12. IANA Considerations

12.1. Message Header Field Registrations

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

12.1.1. Schedule-Reply

Header field name: Schedule-Reply

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document(s): this specification (Section 8.1)

Related information: none

12.1.2. Schedule-Tag

Header field name: Schedule-Tag

Applicable protocol: http

Status: standard

Author/Change controller: IETF

Specification document(s): this specification (Section 8.2)

Related information: none

12.1.3. 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 8.3)

Related information: none

12.2. iCalendar Property Parameter Registrations

The following iCalendar property parameter names have been added to the iCalendar Parameters Registry defined in Section 8.3.3 of [RFC5545].

Parameter	Status	Reference
SCHEDULE-AGENT	Current	RFC 6638, Section 7.1
SCHEDULE-STATUS	Current	RFC 6638, Section 7.3
SCHEDULE-FORCE-SEND	Current	RFC 6638, Section 7.2

12.3. iCalendar REQUEST-STATUS Value Registrations

The following iCalendar "REQUEST-STATUS" values have been added to the iCalendar REQUEST-STATUS Value Registry defined in Section 7.3 of [RFC5546].

Status Code	Status	Reference
1.0	Current	RFC 6638, Section 3.2.9.1
1.1	Current	RFC 6638, Section 3.2.9.2
1.2	Current	RFC 6638, Section 3.2.9.3

12.4. Additional iCalendar Elements Registries

Per this specification, two new IANA registries for iCalendar elements have been added. Additional codes MAY be used, provided the process described in Section 8.2.1 of [RFC5545] is used to register them.

12.4.1. Schedule Agent Values Registry

The following table has been used to initialize the Schedule Agent Values Registry.

Schedule Agent	Status	Reference
SERVER	Current	RFC 6638, Section 7.1
CLIENT	Current	RFC 6638, Section 7.1
NONE	Current	RFC 6638, Section 7.1

12.4.2. Schedule Force Send Values Registry

The following table has been used to initialize the Schedule Force Send Values Registry.

Schedule Force Send	Status	Reference
REQUEST	Current	RFC 6638, Section 7.2
REPLY	Current	RFC 6638, Section 7.2

13. Acknowledgements

The authors would like to thank the following individuals for contributing their ideas and support for writing this specification: Mike Douglass, Lisa Dusseault, Red Dutta, Jacob Farkas, Jeffrey Harris, Helge Hess, Eliot Lear, Andrew McMillan, Alexey Melnikov, Arnaud Quillaud, Julian F. Reschke, Wilfredo Sanchez Vega, and Simon Vaillancourt.

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.

14. References

14.1. Normative References

[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.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

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

[RFC3744] Clemm, G., Reschke, J., 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. Dusseault, "Calendaring Extensions to WebDAV (CalDAV)", RFC 4791, March 2007.

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

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

[RFC5545] Desruisseaux, B., Ed., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 5545, September 2009.

[RFC5546] Daboo, C., Ed., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", RFC 5546, December 2009.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[W3C.REC-xml-20081126] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "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>.

14.2. Informative References

[RFC6047] Melnikov, A., Ed., "iCalendar Message-Based Interoperability Protocol (iMIP)", RFC 6047, December 2010.

Appendix 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 operation. 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    |          *           |
+-----------------------------+----------------------+
```

Appendix B. Example Scheduling Operations

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

The server is assumed to be hosted in the "example.com" domain, and users whose email addresses are at the "example.com" domain are assumed to be hosted by the server. In addition, the email addresses in the "example.net" domain are also valid email addresses for calendar users hosted by the server. Calendar users with an email address at the "example.org" domain are assumed to not be hosted by the server.

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 collections 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
If-None-Match: *

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", as well as record the Schedule-Tag value for future use. 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
Host: cal.example.com
```

>> 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:mailto:wilfredo@e
 xample.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS=
 1.0:mailto:bernard@example.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE;SCHEDULE-STATUS=3.7:mailto: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 accepts Cyrus's invitation and sets an alarm reminder on the event. It uses the If-Schedule-Tag-Match precondition behavior to ensure it does not overwrite any significant changes from the "Organizer" that might have occurred after it retrieved the initial resource data.

>> 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 <<

```
GET /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:mailto:cyrus@ex
 ample.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
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: "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:
 mailto:wilfredo@example.com
ATTENDEE;CN="Bernard Desruisseaux";CUTYPE=INDIVIDUAL;PARTSTAT=
 NEEDS-ACTION;ROLE=REQ-PARTICIPANT;RSVP=TRUE;SCHEDULE-STATUS=1
 .0:mailto:bernard@example.net
ATTENDEE;CN="Mike Douglass";CUTYPE=INDIVIDUAL;PARTSTAT=NEEDS-A
 CTION;RSVP=TRUE;SCHEDULE-STATUS=3.7:mailto: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, Bernard, and Mike.

>> 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
ATTENDEE;CN="Mike Douglass":mailto:mike@example.org
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:response>
<C:recipient>
<D:href>mailto:mike@example.org</D:href>
</C:recipient>
<C:request-status>3.7;Invalid calendar user</C:request-status>
</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
If-None-Match: *

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

<?xml version="1.0" encoding="utf-8" ?>
<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
If-Schedule-Tag-Match: "7775FB30-7534-489E-A79A-0EA147B933EB"

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
Host: cal.example.com
```

>> 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
If-Schedule-Tag-Match: "488177c8-2ea7-4176-a6cb-fab8cfccdea2"

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
Host: cal.example.com
```

>> 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
```

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/