Network Working Group C. Daboo
Internet-Draft
Expires: August 25, 2006

B. Desruisseaux
Oracle
L. Dusseault
OSAF

February 21, 2006

# Calendaring Extensions to WebDAV (CalDAV)
# draft-dusseault-caldav-10

## Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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

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

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

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

This Internet-Draft will expire on August 25, 2006.

## Copyright Notice

## Abstract

This document specifies a set of methods, headers, message bodies, properties, and reports that define calendar access extensions to the WebDAV protocol. The new protocol elements are intended to make WebDAV-based calendaring and scheduling an interoperable standard that supports calendar access, calendar management, calendar sharing,

and calendar publishing.

Table of Contents

## 1. Introduction

The concept of using HTTP [RFC2616] and WebDAV [RFC2518] as a basis for a calendar access protocol is by no means a new concept: it was discussed in the IETF CALSCH working group as early as 1997 or 1998. Several companies have implemented calendar access protocols using HTTP to upload and download iCalendar [RFC2445] objects, and using WebDAV to get listings of resources. However, those implementations do not interoperate because there are many small and big decisions to be made in how to model calendaring data as WebDAV resources, as well as how to implement required features that aren't already part of WebDAV. This document proposes a way to model calendar data in WebDAV, with additional features to make an interoperable calendar access protocol.

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

### 1.1. 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 term "protected" is used in the Conformance field of property definitions as defined in Section 1.4.2 of [RFC3253].

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

### 1.2. XML Namespaces

Definitions of XML elements in this document use XML element type declarations (as found in XML Document Type Declarations), described in Section 3.2 of [W3C.REC-xml-20040204].

The namespace "urn:ietf:params:xml:ns:caldav" is reserved for the XML elements defined in this specification, its revisions, and related CalDAV specifications. XML elements defined by individual implementations MUST NOT use the "urn:ietf:params:xml:ns:caldav" namespace, and instead should use a namespace that they control.

The XML declarations used in this document do not include namespace information. Thus, implementers MUST NOT use these declarations as the only way to create valid CalDAV properties or to validate CalDAV XML element type. Some of the declarations refer to XML elements

defined by WebDAV [RFC2518] which use the "DAV:" namespace. Wherever such XML elements appear, they are explicitly prefixed with "DAV:" to avoid confusion.

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

### 1.3. Method Preconditions and Postconditions

A "precondition" of a method describes the state of the server that must be true for that method to be performed. A "postcondition" of a method describes the state of the server that must be true after that method has been completed. If a method precondition or postcondition for a request is not satisfied, the response status of the request MUST be either 403 (Forbidden) if the request should not be repeated because it will always fail, or 409 (Conflict) if it is expected that the user might be able to resolve the conflict and resubmit the request.

In order to allow better client handling of 403 and 409 responses, a distinct XML element type is associated with each method precondition and postcondition of a request. When a particular precondition is not satisfied or a particular postcondition cannot be achieved, the appropriate XML element MUST be returned as the child of a top-level DAV:error element in the response body, unless otherwise negotiated by the request.

## 2. Requirements Overview

This section lists what functionality is required of a CalDAV server. To advertise support for CalDAV, a server:

- o MUST support iCalendar [RFC2445] as a media type for calendar object resource format;

- o MUST support WebDAV Class 1 [RFC2518];

- o MUST support WebDAV ACL [RFC3744] with the additional privilege defined in Section 6.1 of this document;

- o MUST support transport over TLS [RFC2246] as defined in [RFC2818];

- o MUST support ETags [RFC2616] with additional requirements specified in Section 5.3.4 of this document;

- MUST support all calendaring REPORTs defined in Section 7 of this document; and

- MUST advertise support on all calendar collections and calendar object resources for the calendaring REPORTs in the DAV:supported-report-set property as defined in Versioning Extensions to WebDAV [RFC3253].

In addition, a server:

- SHOULD support the MKCALENDAR method defined in Section 5.3.1 of this document.

## 3. Calendaring Data Model

One of the features which has made WebDAV a successful protocol is its firm data model. This makes it a useful framework for other applications such as calendaring. This specification follows the same pattern by developing all features based on a well-described data model.

As a brief overview, a CalDAV calendar is modeled as a WebDAV collection with a defined structure; each calendar collection contains a number of resources representing calendar objects as its direct child resource. Each resource representing a calendar object (event or to-do, or journal entry, or other calendar components) is called a "calendar object resource". Each calendar object resource and each calendar collection can be individually locked and have individual WebDAV properties. Requirements derived from this model are provided in Section 4.1 and Section 4.2.

### 3.1. Calendar Server

A CalDAV server is a calendaring-aware engine combined with a WebDAV repository. A WebDAV repository is a set of WebDAV collections, containing other WebDAV resources, within a unified URL namespace. For example, the repository "http://www.example.com/webdav/" may contain WebDAV collections and resources, all of which have URLs beginning with "http://www.example.com/webdav/". Note that the root URL "http://www.example.com/" may not itself be a WebDAV repository (for example, if the WebDAV support is implemented through a servlet or other Web server extension).

A WebDAV repository MAY include calendar data in some parts of its URL namespace, and non-calendaring data in other parts.

A WebDAV repository can advertise itself as a CalDAV server if it

supports the functionality defined in this specification at any point within the root of the repository. That might mean that calendaring data is spread throughout the repository and mixed with non-calendar data in nearby collections (e.g., calendar data may be found in /home/lisa/calendars/ as well as in /home/bernard/calendars/, and non-calendar data in /home/lisa/contacts/). Or, it might mean that calendar data can be found only in certain sections of the repository (e.g., /calendar/). Calendaring features are only required in the repository sections that are or contain calendar object resources. So a repository confining calendar data to the /calendar/ collection would only need to support the CalDAV required features within that collection.

The CalDAV server or repository is the canonical location for calendar data and state information. Both CalDAV servers and clients MUST ensure that the data is consistent and compliant. Clients may submit requests to change data or download data. Clients may store calendar objects offline and attempt to synchronize at a later time. However, clients MUST be prepared for calendar data on the server to change between the time of last synchronization and when attempting an update, as calendar collections may be shared and accessible via multiple clients. Entity tags and other features make this possible.

### 3.2. Recurrence and the Data Model

Recurrence is an important part of the data model because it governs how many resources are expected to exist. This specification models a recurring calendar component and its recurrence exceptions as a single resource. In this model, recurrence rules, recurrence dates, exception rules, and exception dates are all part of the data in a single calendar object resource. This model avoids problems of limiting how many recurrence instances to store in the repository, how to keep recurrence instances in sync with the recurring calendar component, and how to link recurrence exceptions with the recurring calendar component. It also results in less data to synchronize between client and server, and makes it easier to make changes to all recurrence instances or to a recurrence rule. It makes it easier to create a recurring calendar component, and easier to delete all recurrence instances.

Clients are not forced to retrieve information about all recurrence instances of a recurring component. The CALDAV:calendar-query and CALDAV:calendar-multiget REPORTs defined in this document allow clients to retrieve only recurrence instances that overlap a given time range.

# 4. Calendar Resources

## 4.1. Calendar Object Resources

Calendar object resources contained in calendar collections MUST NOT contain more than one type of calendar component (e.g., VEVENT, VTODO, VJOURNAL, VFREEBUSY, etc.) with the exception of VTIMEZONE components which MUST be specified for each unique TZID parameter value specified in the iCalendar object. For instance, a calendar object resource can contain two VEVENT components and one VTIMEZONE component, but it cannot contain one VEVENT component and one VTODO component.

Calendar object resources contained in calendar collections MUST NOT specify the iCalendar METHOD property.

The UID property value of the calendar components contained in a calendar object resource MUST be unique in the scope of the calendar collection in which they are stored.

Calendar components in a calendar collection that have different UID property values MUST be stored in separate calendar object resources.

Calendar components with the same UID property value, in a given calendar collection, MUST be contained in the same calendar object resource. This ensures that all components in a recurrence "set" are contained in the same calendar object resource. It is possible for a calendar object resource to just contain components that represent "overridden" instances (ones which modify the behavior of a regular instance, and thus include a RECURRENCE-ID property), without also including the "master" recurring component (the one that defines the recurrence "set" and does not contain any "RECURRENCE-ID" property).

For example, given the following iCalendar object:

```
    BEGIN:VCALENDAR
    PRODID:-//Example Corp.//CalDAV Client//EN
    VERSION:2.0
    BEGIN:VEVENT
    UID:1@example.com
    SUMMARY:One-off Meeting
    DTSTAMP:20041210T183904Z
    DTSTART:20041207T120000Z
    DTEND:20041207T130000Z
    END:VEVENT
    BEGIN:VEVENT
    UID:2@example.com
    SUMMARY:Weekly Meeting
    DTSTAMP:20041210T183838Z
    DTSTART:20041206T120000Z
    DTEND:20041206T130000Z
    RRULE:FREQ=WEEKLY
    END:VEVENT
    BEGIN:VEVENT
    UID:2@example.com
    SUMMARY:Weekly Meeting
    RECURRENCE-ID:20041213T120000Z
    DTSTAMP:20041210T183838Z
    DTSTART:20041213T130000Z
    DTEND:20041213T140000Z
    END:VEVENT
    END:VCALENDAR
```

The VEVENT component with the UID value "1@example.com", would be stored in its own calendar object resource. The two VEVENT components with the UID value "2@example.com", which represent a recurring event where one recurrence instance has been overridden, would be stored in the same calendar object resource.

## 4.2. Calendar Collection

A calendar collection contains calendar object resources that represent calendar components within a calendar. A calendar collection is manifested to clients as a WebDAV resource collection identified by a URL. A calendar collection MUST report the DAV: collection and CALDAV:calendar XML elements in the value of the DAV: resourcetype property. The element type declaration for CALDAV: calendar is:

```
    <!ELEMENT calendar EMPTY>
```

A calendar collection can be created through provisioning (e.g., automatically created when a user's account is provisioned), or it

can be created with the MKCALENDAR method (see Section 5.3.1). This method can be useful for a user to create additional calendars (e.g., soccer schedule) or for users to share a calendar (e.g., team events or conference room). Note however that this document doesn't define what extra calendar collections are for. Users must rely on non-standard cues to find out what a calendar collection is for, or use the CALDAV:calendar-description property defined in Section 5.2.1 to provide such a cue.

Calendar collections MUST only contain calendar object resources and collections that are not calendar collections. Furthermore, collections contained in calendar collections MUST NOT contain calendar collections. This specification does not define how collections contained in calendar collections are used and may relate to the calendar object resources contained in the calendar collections.

Multiple calendar collections MAY be children of the same collection.

## 5. Calendar Access Feature

### 5.1. Calendar Access Support

A server supporting the features described in this document MUST include "calendar-access" as a field in the DAV response header from an OPTIONS request on any resource that supports any calendar properties, reports, method, or privilege. A value of "calendar-access" in the DAV response header MUST indicate that the server supports all MUST level requirements specified in this document.

#### 5.1.1. Example: Using OPTIONS for the Discovery of Calendar Access Support

>> Request <<

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

>> Response <<

```
HTTP/1.1 200 OK
Allow: OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, COPY, MOVE
Allow: PROPFIND, PROPPATCH, LOCK, UNLOCK, REPORT, ACL
DAV: 1, 2, access-control, calendar-access
Date: Fri, 11 Nov 2005 09:32:12 GMT
Content-Length: 0
```

In this example, the OPTIONS method returns the value "calendar-access" in the DAV response header to indicate that the collection "/home/bernard/calendars/" may support properties, reports, methods, or privilege defined in this specification.

### 5.2. Calendar Collection Properties

This section defines properties that MAY be defined on calendar collections.

#### 5.2.1. CALDAV:calendar-description Property

Name: calendar-description

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

Purpose: Provides a human-readable description of the calendar collection.

Conformance: This property MAY be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]).  An xml:lang attribute indicating the human language of the description SHOULD be set for this property by clients or through server provisioning.  Servers MUST return any xml:lang attribute if set for the property.

Description: The CALDAV:calendar-description property MAY be defined on any calendar collection.  If present, the property contains a description of the calendar collection that is suitable for presentation to a user.

Definition:

```
<!ELEMENT calendar-description (#PCDATA)>
PCDATA value: string
```

Example:

```
<C:calendar-description xml:lang="fr-CA"
   xmlns:C="urn:ietf:params:xml:ns:caldav"
>Calendrier de Mathilde Desruisseaux</C:calendar-description>
```

#### 5.2.2. CALDAV:calendar-timezone Property

Name: calendar-timezone

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

Purpose: Specifies a time zone on a calendar collection.

Conformance: This property SHOULD NOT be returned by a PROPFIND DAV: allprop request (as defined in Section 12.14.1 of [RFC2518]).

Description: The CALDAV:calendar-timezone property SHOULD be defined on all calendar collections to specify the time zone the server should rely on to resolve "date" values and "date with local time" values (i.e., floating time) to "date with UTC time" values. The server will require this information to determine if a calendar component scheduled with "date" values or "date with local time" values overlaps a CALDAV:time-range specified in a CALDAV: calendar-query REPORT. The server will also require this information to compute the proper FREEBUSY time period as "date with UTC time" in the VFREEBUSY component returned in a response to a CALDAV:free-busy-query REPORT request that takes into account calendar components scheduled with "date" values or "date with local time" values.

Definition:

```
      <!ELEMENT calendar-timezone (#PCDATA)>
      PCDATA value: an iCalendar object with exactly one VTIMEZONE
            component.
```

Example:

```
    <C:calendar-timezone
       xmlns:C="urn:ietf:params:xml:ns:caldav">BEGIN:VCALENDAR
   PRODID:-//Example Corp.//CalDAV Client//EN
   VERSION:2.0
   BEGIN:VTIMEZONE
   TZID:US-Eastern
   LAST-MODIFIED:19870101T000000Z
   BEGIN:STANDARD
   DTSTART:19671029T020000
   RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
   TZOFFSETFROM:-0400
   TZOFFSETTO:-0500
   TZNAME:Eastern Standard Time (US &amp; Canada)
   END:STANDARD
   BEGIN:DAYLIGHT
   DTSTART:19870405T020000
   RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
   TZOFFSETFROM:-0500
   TZOFFSETTO:-0400
   TZNAME:Eastern Daylight Time (US &amp; Canada)
   END:DAYLIGHT
   END:VTIMEZONE
   END:VCALENDAR
   </C:calendar-timezone>
```

### 5.2.3. CALDAV:supported-calendar-component-set Property

Name: supported-calendar-component-set

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

Purpose: Specifies the calendar component types (e.g., VEVENT, VTODO, etc.) that calendar object resources may contain in the calendar collection.

Conformance: This property MUST be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]).

Description: The CALDAV:supported-calendar-component-set property MAY be defined on any calendar collection to specify restrictions on the calendar component types that calendar object resources may contain in a calendar collection.  Since this property is protected it cannot be changed by clients using a PROPPATCH request.  However, clients can initialize the value of this property when creating a new calendar collection with MKCALENDAR. The empty-element tag <C:comp name="VTIMEZONE"/> MUST only be specified if support for calendar object resources that only

contain VTIMEZONE components is provided or desired. Support for VTIMEZONE components in calendar object resources that contain VEVENT or VTODO components is always assumed.

Definition:

```
      <!ELEMENT supported-calendar-component-set (comp*)>
```

Example:

```
      <C:supported-calendar-component-set
          xmlns:C="urn:ietf:params:xml:ns:caldav">
        <C:comp name="VEVENT"/>
        <C:comp name="VTODO"/>
      </C:supported-calendar-component-set>
```

### 5.2.4. CALDAV:supported-calendar-data Property

Name: supported-calendar-data

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

Purpose: Specifies restrictions on a calendar collection.

Conformance: This property MUST be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]).

Description: The CALDAV:supported-calendar-data property MAY be defined on any calendar collection to specify the media type supported for the calendar object resources contained in a given calendar collection (e.g., iCalendar version 2.0).

Definition:

```
      <!ELEMENT supported-calendar-data (calendar-data*)>
```

Example:

```
      <C:supported-calendar-data
         xmlns:C="urn:ietf:params:xml:ns:caldav">
        <C:calendar-data content-type="text/calendar" version="2.0"/>
      </C:supported-calendar-data>
```

### 5.2.5. Additional Precondition for PROPPATCH

This specification creates an additional Precondition for the PROPPATCH method. The precondition is:

(CALDAV:valid-calendar-data): The time zone specified in CALDAV:calendar-timezone property MUST be a valid iCalendar object containing a single valid VTIMEZONE component.

### 5.3. Creating Resources

The creation of calendar collections and calendar object resources may be initiated by either a CalDAV client or by the CalDAV server. For example, a server might come pre-configured with a user's calendar collection, or the CalDAV client might request the server to create a new calendar collection for a given user. Servers might populate events as calendar objects inside a calendar collection, or clients might request the server to create events. Either way, both client and server MUST comply with the requirements in this document, and MUST understand objects appearing in calendar collections or according to the data model defined here.

#### 5.3.1. MKCALENDAR Method

An HTTP request using the MKCALENDAR method creates a new calendar collection resource. A server MAY restrict calendar collection creation to particular collections.

Support for MKCALENDAR on the server is only RECOMMENDED and not REQUIRED because some calendar stores only support one calendar per user (or principal) and those are typically pre-created for each account. However, servers and clients are strongly encouraged to support MKCALENDAR whenever possible to allow users to create multiple calendar collections to better help organize their data.

Clients SHOULD use the DAV:displayname property for a human-readable name of the calendar. Clients can either specify the value of the DAV:displayname property in the request body of the MKCALENDAR request, or alternatively issue a PROPPATCH request to change the DAV:displayname property to the appropriate value immediately after issuing the MKCALENDAR request. Clients SHOULD NOT set the DAV:displayname property to be the same as any other calendar collection at the same URI "level". When displaying calendar collections to users, clients SHOULD check the DAV:displayname property and use that value as the name of the calendar. In the event that the DAV:displayname property is empty, the client MAY use the last part of the calendar collection URI as the name, however that path segment may be "opaque" and not represent any meaningful human-readable text.

If a MKCALENDAR request fails, the server state preceding the request MUST be restored.

Marshalling:

If a request body is included, it MUST be a CALDAV:mkcalendar XML element. Instruction processing MUST occur in the order instructions are received (i.e., from top to bottom). Instructions MUST either all be executed or none executed. Thus if any error occurs during processing, all executed instructions MUST be undone and a proper error result returned. Instruction processing details can be found in the definition of the DAV:set instruction in Section 12.13 of [RFC2518].

```
<!ELEMENT mkcalendar (DAV:set)>
```

If a response body for a successful request is included, it MUST be a CALDAV:mkcalendar-response XML element.

```
<!ELEMENT mkcalendar-response ANY>
```

The response MUST include a Cache-Control:no-cache header.

Preconditions:

(DAV:resource-must-be-null): A resource MUST NOT exist at the Request-URI;

(CALDAV:calendar-collection-location-ok): The Request-URI MUST identify a location where a calendar collection can be created;

(CALDAV:valid-calendar-data): The time zone specified in the CALDAV:calendar-timezone property MUST be a valid iCalendar object containing a single valid VTIMEZONE component;

(DAV:needs-privilege): The DAV:bind privilege MUST be granted to the current user on the parent collection of the Request-URI.

Postconditions:

(CALDAV:initialize-calendar-collection): A new calendar collection exists at the Request-URI. The DAV:resourcetype of the calendar collection MUST contain both DAV:collection and CALDAV:calendar XML elements.

#### 5.3.1.1. Status Codes

The following are examples of response codes one would expect to get in a response to a MKCALENDAR request. Note that this list is by no means exhaustive.

201 (Created) - The calendar collection resource was created in its entirety;

207 (Multi-Status) - The calendar collection resource was not created since one or more DAV:set instructions specified in the request body could not be processed successfully. The following are examples of response codes one would expect to be used in a 207 (Multi-Status) response in this situation:

403 (Forbidden) - The client, for reasons the server chooses not to specify, cannot alter one of the properties;

409 (Conflict) - The client has provided a value whose semantics are not appropriate for the property. This includes trying to set read-only properties;

424 (Failed Dependency) - The DAV:set instruction on the specified resource would have succeeded if it were not for the failure of another DAV:set instruction specified in the request body;

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; and

507 (Insufficient Storage) - The server did not have sufficient space to record the property;

403 (Forbidden) - This indicates at least one of two conditions: 1) the server does not allow the creation of calendar collections at the given location in its namespace, or 2) the parent collection of the Request-URI exists but cannot accept members;

409 (Conflict) - A collection cannot be made at the Request-URI until one or more intermediate collections have been created;

415 (Unsupported Media Type) - The server does not support the request type of the body; and

507 (Insufficient Storage) - The resource does not have sufficient space to record the state of the resource after the execution of this method.

#### 5.3.1.2. Example: Successful MKCALENDAR request

This example creates a calendar collection called /home/lisa/calendars/events/ on the server cal.example.com with specific values for the properties DAV:displayname, CALDAV:calendar-description, CALDAV:supported-calendar-component-set, and CALDAV:calendar-timezone.

```
>> Request <<

MKCALENDAR /home/lisa/calendars/events/ HTTP/1.1
Host: cal.example.com
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:mkcalendar xmlns:D="DAV:"
              xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:set>
    <D:prop>
      <D:displayname>Lisa's Events</D:displayname>
      <C:calendar-description xml:lang="en"
>Calendar restricted to events.</C:calendar-description>
      <C:supported-calendar-component-set>
        <C:comp name="VEVENT"/>
      </C:supported-calendar-component-set>
      <C:calendar-timezone><![CDATA[BEGIN:VCALENDAR
PRODID:-//Example Corp.//CalDAV Client//EN
VERSION:2.0
BEGIN:VTIMEZONE
TZID:US-Eastern
LAST-MODIFIED:19870101T000000Z
BEGIN:STANDARD
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
TZNAME:Eastern Standard Time (US & Canada)
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
TZNAME:Eastern Daylight Time (US & Canada)
END:DAYLIGHT
END:VTIMEZONE
END:VCALENDAR
]]></C:calendar-timezone>
    </D:prop>
  </D:set>
</C:mkcalendar>
```

```
   >> Response <<

   HTTP/1.1 201 Created
   Cache-Control: no-cache
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Length: 0
```

### 5.3.2. Creating Calendar Object Resources

Clients populate calendar collections with calendar object resources. The URL for each calendar object resource is entirely arbitrary, and does not need to bear a specific relationship to the calendar object resource's iCalendar properties or other metadata. New calendar object resources MUST be created with a PUT request targeted at an unmapped URI. A PUT request targeted at a mapped URI updates an existing calendar object resource.

When servers create new resources, it's not hard for the server to choose an unmapped URI. It's slightly tougher for clients, because a client might not want to examine all resources in the collection, and might not want to lock the entire collection to ensure that a new resource isn't created with a name collision. However, there is an HTTP feature to mitigate this. If the client intends to create a new non-collection resource, such as a new VEVENT, the client SHOULD use the HTTP request header "If-None-Match: *" on the PUT request. The Request-URI on the PUT request MUST include the target collection, where the resource is to be created, plus the name of the resource in the last path segment. The "If-None-Match: *" request header ensures that the client will not inadvertently overwrite an existing resource, if the last path segment turned out to already be used.

```
>> Request <<

PUT /home/lisa/calendars/events/qwue23489.ics HTTP/1.1
If-None-Match: *
Host: cal.example.com
Content-Type: text/calendar
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VEVENT
UID:20010712T182145Z-123401@example.com
DTSTAMP:20010712T182145Z
DTSTART:20010714T170000Z
DTEND:20010715T040000Z
SUMMARY:Bastille Day Party
END:VEVENT
END:VCALENDAR

>> Response <<

HTTP/1.1 201 Created
Content-Length: 0
Date: Fri, 11 Nov 2005 09:32:12 GMT
ETag: "123456789-000-111"
```

The request to change an existing event is the same, but with a specific ETag in the "If-Match" header, rather than the "If-None-Match" header.

As indicated in Section 3.10 of [RFC2445], the URL of calendar object resources containing (an arbitrary set of) calendaring and scheduling information may be suffixed by ".ics", and the URL of calendar object resources containing free or busy time information may be suffixed by ".ifb".

#### 5.3.2.1. Additional Preconditions for PUT, COPY and MOVE

This specification creates additional Preconditions for PUT, COPY and MOVE methods. These preconditions apply:

When a PUT operation of a calendar object resource into a calendar collection occurs.

When a COPY or MOVE operation of a calendar object resource into a calendar collection occurs.

The new preconditions are:

(CALDAV:supported-calendar-data): The resource submitted in the PUT request, or targeted by a COPY or MOVE request MUST be a supported media type (i.e., iCalendar) for calendar object resources;

(CALDAV:valid-calendar-data): The resource submitted in the PUT request, or targeted by a COPY or MOVE request MUST be valid data for the media type being specified (i.e., MUST contain valid iCalendar data);

(CALDAV:valid-calendar-object-resource): The resource submitted in the PUT request, or targeted by a COPY or MOVE request MUST obey all restrictions specified in Section 4.1 (e.g., calendar object resources MUST NOT contain more than one type of calendar component, calendar object resources MUST NOT specify the iCalendar METHOD property, etc.);

(CALDAV:supported-calendar-component): The resource submitted in the PUT request, or targeted by a COPY or MOVE request MUST contain a type of calendar component that is supported in the targeted calendar collection;

(CALDAV:no-uid-conflict): The resource submitted in the PUT request, or targeted by a COPY or MOVE request MUST NOT specify an iCalendar UID property value already in use in the targeted calendar collection or overwrite an existing calendar object resource with one that has a different UID property value. Servers SHOULD report the URL of the resource that is already making use of the same UID property value in the DAV:resource element;

```
          <!ELEMENT no-uid-conflict (DAV:resource)>
```

### 5.3.3. Non-standard components, properties and parameters

iCalendar provides a "standard mechanism for doing non-standard things". This extension support allows implementers to make use of non-standard components, properties and parameters whose names are prefixed with the text "X-".

Servers MUST support the use of non-standard components, properties and parameters in calendar object resources stored via the PUT method.

Servers MAY reject any non-standard components, properties and parameters that have specific values in calendar object resources

stored via the PUT method. This allows the server to enforce rules for its own "private" values that it may use.

### 5.3.4. Calendar Object Resource Entity Tag

The DAV:getetag property MUST be defined and set to a strong entity tag on all calendar object resources.

A response to a GET request targeted at a calendar object resource MUST contain an ETag response header field indicating the current value of the strong entity tag of the calendar object resource.

A response to a PUT request MAY contain an ETag response header field indicating the current value of the entity tag for the calendar object resource just created or modified.

As required by [RFC2616], a response to a PUT request with a strong entity tag MUST mean that the server will return on a subsequent GET request a calendar object resource that is equivalent by octet equality.

A response to a PUT request MUST NOT contain an ETag response header field if the server will return on a subsequent GET request a calendar object resource that is not equivalent by octet equality to the submitted calendar object resource. In this case, the client SHOULD retrieve the new entity (and ETag) as a basis for further changes, rather than use the entity it had sent with the PUT request.

# 6. Calendaring Access Control

## 6.1. Calendaring Privilege

CalDAV servers MUST support and adhere to the requirements of WebDAV ACL [RFC3744]. WebDAV ACL provides a framework for an extensible set of privileges that can be applied to WebDAV collections and ordinary resources. CalDAV servers MUST also support the calendaring privilege defined in this section.

### 6.1.1. CALDAV:read-free-busy Privilege

Calendar users often wish to allow other users to see their busy time information, without viewing the other details of the calendar components (location, summary, attendees). This allows a significant amount of privacy while still allowing other users to schedule meetings at times when the user is likely to be free.

The CALDAV:read-free-busy privilege controls which calendar

collections and calendar object resources are examined when a CALDAV:free-busy-query REPORT request is processed (see Section 7.9). This privilege can be granted on calendar collections or calendar object resources. Servers MUST support this privilege on all calendar collections and calendar object resources.

```
        <!ELEMENT read-free-busy EMPTY>
```

The CALDAV:read-free-busy privilege MUST be aggregated in the DAV:read privilege. Servers MUST allow the CALDAV:read-free-busy to be granted without the DAV:read privilege being granted.

Clients should note that when only the CALDAV:read-free-busy privilege has been granted on a resource, this does not imply access to GET, HEAD, OPTIONS and PROPFIND on the resource -- those operations are governed by the DAV:read privilege.

### 6.2. Additional Principal Property

This section defines an additional property for WebDAV principal resources as defined in [RFC3744].

#### 6.2.1. CALDAV:calendar-home-set Property

Name: calendar-home-set

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

Purpose: Identifies the URL of any WebDAV collections that contain calendar collections owned by the associated principal resource.

Conformance: This property MAY be protected and SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in Section 12.14.1 of [RFC2518]). Support for this property is RECOMMENDED.

Description: The CALDAV:calendar-home-set property is meant to allow users to easily find the calendar collections owned by the principal. Typically, users will group all the calendar collections that they own under a common collection. This property specify the URL of collections that either are calendar collections or ordinary collections that have child or descendant calendar collections owned by the principal.

Definition:

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

Example:

```
    <C:calendar-home-set xmlns:D="DAV:"
                         xmlns:C="urn:ietf:params:xml:ns:caldav">
      <D:href>http://cal.example.com/home/bernard/calendars/</D:href>
    </C:calendar-home-set>
```

## 7. Calendaring Reports

This section defines the REPORTs that CalDAV servers MUST support on calendar collections and calendar object resources.

CalDAV servers MUST advertise support for these REPORTs on all calendar collections and calendar object resources with the DAV: supported-report-set property defined in Section 3.1.5 of [RFC3253]. CalDAV servers MAY also advertise support for these REPORTs on ordinary collections.

Some of these REPORTs allow calendar data (from possibly multiple resources) to be returned.

### 7.1. REPORT Method

The REPORT method (defined in Section 3.6 of [RFC3253]) provides an extensible mechanism for obtaining information about one or more resources. Unlike the PROPFIND method, which returns the value of one or more named properties, the REPORT method can involve more complex processing. REPORT is valuable in cases where the server has access to all of the information needed to perform the complex request (such as a query), and where it would require multiple requests for the client to retrieve the information needed to perform the same request.

CalDAV servers MUST support the DAV:expand-property REPORT defined in Section 3.8 of [RFC3253].

### 7.2. Ordinary collections

Servers MAY support the REPORTs defined in this document on ordinary collections, that is, collections that are not calendar collections. In computing responses to the REPORTs defined in this document, servers MUST only consider calendar object resources contained in calendar collections, subject also to the value of the Depth request header.

### 7.3. Date and floating time

iCalendar provides a way to specify DATE and DATE-TIME values that are not bound to any time zone in particular, hereafter called "floating date" and "floating time" respectively. These values are used to represent the same day, hour, minute and second value regardless of which time zone is being observed. For instance, the DATE value "20051111", represents November 11th, 2005 in no specific time zone, while the DATE-TIME value "20051111T111100" represents November 11th, 2005 at 11:11 AM in no specific time zone.

CalDAV servers may need to convert "floating date" and "floating time" values in date with UTC time values in the processing of calendaring REPORT requests.

For the CALDAV:calendar-query REPORT, CalDAV servers MUST rely on the value of the CALDAV:timezone XML element, if specified as part of the request body, to perform the proper conversion of "floating date" and "floating time" values to date with UTC time values. If the CALDAV: timezone XML element is not specified in the request body, CalDAV servers MUST rely on the value of the CALDAV:calendar-timezone property, if defined, else the CalDAV servers MAY rely on the time zone of their choice.

For the CALDAV:free-busy-query REPORT, CalDAV servers MUST rely on the value of the CALDAV:calendar-timezone property, if defined, to compute the proper FREEBUSY time period value as date with UTC time, for calendar components scheduled with "floating date" or "floating time". If the CALDAV:calendar-timezone property is not defined, CalDAV servers MAY rely on the time zone of their choice.

### 7.4. Time range filtering

Some of the reports defined in this section can be targeted at calendar object resources within a specific time range. To determine whether a calendar object resource matches the time range filter element, the start and end times for the particular type of object are determined and then compared to the requested time range. If the start and end overlap the requested time range, then the calendar object resource matches the filter element. The rules defined in [RFC2445] for determining the actual start and end times of calendar components MUST be used, along with the rules for determining overlap specified in Section 9.8 of this document.

When such time range filtering is used, special consideration must be given to recurring calendar components such as VEVENT and VTODO components. The server MUST expand recurring components to determine whether any recurrence instances overlap the specified time range.

If one or more recurrence instances overlap the time range, then the calendar object resource matches the filter element.

### 7.5. Returned calendar components

In addition, CalDAV provides three ways to determine which components of a calendar object resource are returned from the recurrence set. The three options are:

1. Return all the calendar components contained in the calendar object resources. This includes the component that defines the recurrence set, referred to as the "master component", as well as the components that define exceptions to the recurrence set, referred to as the "overridden components". According to the rules defined in Section 3.2 all recurrence instances of a recurring component will always be contained in the same calendar object resource.

2. Return the "master component" and only the "overridden components" that currently or originally overlap the specified time range. This avoids the need for clients to process "overridden components" outside of the time range they are interested in. See Section 9.5.6.

3. Return "expanded" calendar components that represent only those recurrence instances in the recurrence set that overlap the specified time range. This avoids the need for clients to do any recurrence processing themselves as the server does the expansion for them and provides the list of instances. See Section 9.5.5.

### 7.6. Non-standard components, properties and parameters

Servers MUST support the use of non-standard component, property or parameter names in the CALDAV:calendar-data XML element in calendaring REPORT requests to allow clients to request that non-standard components, properties and parameters be returned in the calendar data provided in the response.

Servers MAY support the use of non-standard component, property or parameter names in the CALDAV:comp-filter, CALDAV:prop-filter and CALDAV:param-filter XML elements specified in the CALDAV:filter XML element of calendaring REPORT requests.

Servers MUST fail with the CALDAV:supported-filter precondition if a calendaring REPORT request uses a CALDAV:comp-filter, CALDAV:prop-filter or CALDAV:param-filter XML element that makes reference to a non-standard component, property or parameter name which the server does not support queries on.

## 7.7. CALDAV:calendar-query Report

The CALDAV:calendar-query REPORT performs a search for all calendar object resources that match a specified filter. The response of this REPORT will contain all the WebDAV properties and calendar object resource data specified in the request. In the case of the CALDAV:calendar-data XML element, one can explicitly specify the calendar components and properties that should be returned in the calendar object resource data that matches the filter.

The format of this REPORT is modeled on the PROPFIND method. The request and response bodies of the CALDAV:calendar-query REPORT use XML elements that are also used by PROPFIND. In particular the request can include XML elements to request WebDAV properties to be returned. When that occurs the response should follow the same behavior as PROPFIND with respect to the DAV:multistatus response elements used to return specific property results. For instance, a request to retrieve the value of a property which does not exist is an error and MUST be noted with a response XML element which contains a 404 (Not Found) status value.

Support for the CALDAV:calendar-query REPORT is REQUIRED.

Marshalling:

The request body MUST be a CALDAV:calendar-query XML element as defined in Section 9.4.

The response body for a successful request MUST be a DAV:multistatus XML element (i.e., the response uses the same format as the response for PROPFIND). In the case where there are no response elements, the returned DAV:multistatus XML element is empty.

The response body for a successful CALDAV:calendar-query REPORT request MUST contain a DAV:response element for each iCalendar object that matched the search filter. Calendar data is being returned in the CALDAV:calendar-data XML element inside the DAV:propstat XML element.

Preconditions:

(CALDAV:supported-calendar-data): The attributes "content-type" and "version" of the CALDAV:calendar-data XML elements specify a media type supported by the server for calendar object resources.

(CALDAV:valid-filter): The CALDAV:filter XML element specified in the REPORT request MUST be valid. For instance, a CALDAV:filter

cannot nest a <C:comp name="VEVENT"> element in a <C:comp name="VTODO"> element, or a CALDAV:filter cannot nest a <C:time-range start="..." end="..."> element in a <C:prop name="SUMMARY"> element.

(CALDAV:supported-filter): The CALDAV:comp-filter, CALDAV:prop-filter and CALDAV:param-filter XML elements used in the CALDAV:filter XML element in the REPORT request only makes reference to components, properties and parameters on which queries are supported by the server. Servers SHOULD report the CALDAV:comp-filter, CALDAV:prop-filter or CALDAV:param-filter for which it does not provide support.

```
      <!ELEMENT supported-filter (comp-filter*,
                                  prop-filter*,
                                  param-filter*)>
```

(CALDAV:valid-calendar-data): The time zone specified in the REPORT request MUST be a valid iCalendar object containing a single valid VTIMEZONE component.

Postconditions:

(DAV:number-of-matches-within-limits): The number of matching calendar object resources must fall within server-specific, predefined limits. For example, this condition might be triggered if a search specification would cause the return of an extremely large number of responses.

### 7.7.1. Example: Partial retrieval of events by time range

In this example, the client requests the server to return specific components and properties of the VEVENT components that overlap the time range from January 4th, 2006 at 00:00:00 AM UTC to January 5th, 2006 at 00:00:00 AM UTC. In addition the DAV:getetag property is also requested and returned as part of the response. Note that the first calendar object returned is a recurring event whose first instance lies outside of the requested time range, but whose third instance does overlap the time range. Note that due to the CALDAV:calendar-data element restrictions, the DTSTAMP property in VEVENT components has not been returned, and only the only property returned in the VCALENDAR object is VERSION.

```
   >> Request <<

   REPORT /bernard/work/ HTTP/1.1
   Host: cal.example.com
   Depth: 1
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <C:calendar-query xmlns:D="DAV:"
                 xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:prop>
       <D:getetag/>
       <C:calendar-data>
         <C:comp name="VCALENDAR">
           <C:prop name="VERSION"/>
           <C:comp name="VEVENT">
             <C:prop name="SUMMARY"/>
             <C:prop name="UID"/>
             <C:prop name="DTSTART"/>
             <C:prop name="DTEND"/>
             <C:prop name="DURATION"/>
             <C:prop name="RRULE"/>
             <C:prop name="RDATE"/>
             <C:prop name="EXRULE"/>
             <C:prop name="EXDATE"/>
             <C:prop name="RECURRENCE-ID"/>
           </C:comp>
           <C:comp name="VTIMEZONE"/>
         </C:comp>
       </C:calendar-data/>
     </D:prop>
     <C:filter>
       <C:comp-filter name="VCALENDAR">
         <C:comp-filter name="VEVENT">
           <C:time-range start="20060104T000000Z"
                         end="20060105T000000Z"/>
         </C:comp-filter>
       </C:comp-filter>
     </C:filter>
   </C:calendar-query>

   >> Response <<

   HTTP/1.1 207 Multi-Status
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx
```

```
   <?xml version="1.0" encoding="utf-8" ?>
   <D:multistatus xmlns:D="DAV:"
              xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd2.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd2"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      DTSTART;TZID=US/Eastern:20060102T120000
      DURATION:PT1H
      RRULE:FREQ=DAILY;COUNT=5
      SUMMARY:Event #2
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      BEGIN:VEVENT
      DTSTART;TZID=US/Eastern:20060104T140000
      DURATION:PT1H
      RECURRENCE-ID;TZID=US/Eastern:20060104T120000
      SUMMARY:Event #2 bis
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      BEGIN:VEVENT
      DTSTART;TZID=US/Eastern:20060106T140000
      DURATION:PT1H
      RECURRENCE-ID;TZID=US/Eastern:20060106T120000
      SUMMARY:Event #2 bis bis
      UID:00959BC664CA650E933C892C@example.com
```

```
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd3"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      ATTENDEE;PARTSTAT=ACCEPTED;ROLE=CHAIR:mailto:cyrus@example.com
      ATTENDEE;PARTSTAT=NEEDS-ACTION:mailto:lisa@example.com
      DTSTAMP:20060206T001220Z
      DTSTART;TZID=US/Eastern:20060104T100000
      DURATION:PT1H
      LAST-MODIFIED:20060206T001330Z
      ORGANIZER:mailto:cyrus@example.com
      SEQUENCE:1
      STATUS:TENTATIVE
      SUMMARY:Event #3
      UID:DC6C50A017428C5216A2F1CD@example.com
      X-ABC-GUID:E1CX5Dr-0007ym-Hz@example.com
      END:VEVENT
      END:VCALENDAR
```

```
          </C:calendar-data>
        </D:prop>
        <D:status>HTTP/1.1 200 OK</D:status>
      </D:propstat>
    </D:response>
  </D:multistatus>
```

### 7.7.2. Example: Partial retrieval of recurring events

In this example, the client requests the server to return VEVENT components that overlap the time range from January 3rd, 2006 at 00:00:00 AM UTC to January 5th, 2006 at 00:00:00 AM UTC. Use of the CALDAV:limit-recurrence-set element causes the server to only return overridden recurrence components that overlap the time range specified in that element, or that affect other instances that overlap the time range (e.g., in the case of a "THISANDFUTURE" behavior). In this example the first overridden component in the matching resource is returned but the second one is not.

>> Request <<

```
REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <C:calendar-data>
      <C:limit-recurrence-set start="20060103T000000Z"
                              end="20060105T000000Z"/>
    </C:calendar-data>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VEVENT">
        <C:time-range start="20060103T000000Z"
                      end="20060105T000000Z"/>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>
```

>> Response <<

```
   HTTP/1.1 207 Multi-Status
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <D:multistatus xmlns:D="DAV:"
              xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd2.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd2"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART;TZID=US/Eastern:20060102T120000
      DURATION:PT1H
      RRULE:FREQ=DAILY;COUNT=5
      SUMMARY:Event #2
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART;TZID=US/Eastern:20060104T140000
      DURATION:PT1H
      RECURRENCE-ID;TZID=US/Eastern:20060104T120000
      SUMMARY:Event #2 bis
```

```
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd3"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      ATTENDEE;PARTSTAT=ACCEPTED;ROLE=CHAIR:mailto:cyrus@example.com
      ATTENDEE;PARTSTAT=NEEDS-ACTION:mailto:lisa@example.com
      DTSTAMP:20060206T001220Z
      DTSTART;TZID=US/Eastern:20060104T100000
      DURATION:PT1H
      LAST-MODIFIED:20060206T001330Z
      ORGANIZER:mailto:cyrus@example.com
      SEQUENCE:1
      STATUS:TENTATIVE
      SUMMARY:Event #3
      UID:DC6C50A017428C5216A2F1CD@example.com
      X-ABC-GUID:E1CX5Dr-0007ym-Hz@example.com
      END:VEVENT
```

```
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
   </D:multistatus>
```

### 7.7.3. Example: Expanded retrieval of recurring events

In this example, the client requests the server to return VEVENT components that overlap the time range from January 2nd, 2006 at 00:00:00 AM UTC to January 5th, 2006 at 00:00:00 AM UTC and to return recurring calendar components expanded into individual recurrence instance calendar components. Use of the CALDAV:expand element causes the server to only return overridden recurrence instances that overlap the time range specified in that element.

```
   >> Request <<

   REPORT /bernard/work/ HTTP/1.1
   Host: cal.example.com
   Depth: 1
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <C:calendar-query xmlns:D="DAV:"
                     xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:prop>
       <C:calendar-data>
         <C:expand start="20060103T000000Z"
                                  end="20060105T000000Z"/>
       </C:calendar-data>
     </D:prop>
     <C:filter>
       <C:comp-filter name="VCALENDAR">
         <C:comp-filter name="VEVENT">
           <C:time-range start="20060103T000000Z"
                         end="20060105T000000Z"/>
         </C:comp-filter>
       </C:comp-filter>
     </C:filter>
   </C:calendar-query>

   >> Response <<

   HTTP/1.1 207 Multi-Status
```

```
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <D:multistatus xmlns:D="DAV:"
              xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd2.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd2"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART:20060103T170000
      DURATION:PT1H
      RECURRENCE-ID:20060103T170000
      SUMMARY:Event #2
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART:20060104T190000
      DURATION:PT1H
      RECURRENCE-ID:20060104T170000
      SUMMARY:Event #2 bis
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd3"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VEVENT
      ATTENDEE;PARTSTAT=ACCEPTED;ROLE=CHAIR:mailto:cyrus@example.com
      ATTENDEE;PARTSTAT=NEEDS-ACTION:mailto:lisa@example.com
```

```
      DTSTAMP:20060206T001220Z
      DTSTART:20060104T150000
      DURATION:PT1H
      LAST-MODIFIED:20060206T001330Z
      ORGANIZER:mailto:cyrus@example.com
      SEQUENCE:1
      STATUS:TENTATIVE
      SUMMARY:Event #3
      UID:DC6C50A017428C5216A2F1CD@example.com
      X-ABC-GUID:E1CX5Dr-0007ym-Hz@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
   </D:multistatus>
```

### 7.7.4. Example: Partial retrieval of stored free busy components

In this example, the client requests the server to return the VFREEBUSY components that have free busy information that overlap the time range from January 2nd, 2006 at 00:00:00 AM UTC (inclusively) to January 3rd, 2006 at 00:00:00 AM UTC (exclusively). Use of the CALDAV:limit-freebusy-set element causes the server to only return the FREEBUSY property values that overlap the time range specified in that element. Note that this is not an example of discovering when the calendar owner is busy.

```
>> Request <<

REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:D="DAV:"
              xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <C:calendar-data>
      <C:limit-freebusy-set start="20060102T000000Z"
                              end="20060103T000000Z"/>
    </C:calendar-data>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VFREEBUSY">
        <C:time-range start="20060102T000000Z"
                        end="20060103T000000Z"/>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>
```

```
   >> Response <<

   HTTP/1.1 207 Multi-Status
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <D:multistatus xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd6.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd6"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VFREEBUSY
      ORGANIZER;CN="Bernard Desruisseaux":mailto:bernard@example.com
      UID:76ef34-54a3d2@example.com
      DTSTAMP:20050530T123421Z
      DTSTART:20060101T100000Z
      DTEND:20060108T100000Z
      FREEBUSY;FBTYPE=BUSY-TENTATIVE:20060102T100000Z/20060102T120000Z
      END:VFREEBUSY
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
   </D:multistatus>
```

### 7.7.5. Example: Retrieval of to-dos by alarm time range

In this example, the client requests the server to return the VTODO components that have an alarm trigger scheduled in the specified time range.

```
   >> Request <<

   REPORT /bernard/work/ HTTP/1.1
   Host: cal.example.com
   Depth: 1
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <C:calendar-query xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:prop xmlns:D="DAV:">
       <D:getetag/>
       <C:calendar-data/>
     </D:prop>
     <C:filter>
       <C:comp-filter name="VCALENDAR">
         <C:comp-filter name="VTODO">
           <C:comp-filter name="VALARM">
             <C:time-range start="20060106T100000Z"
                                 end="20060107T100000Z"/>
           </C:comp-filter>
         </C:comp-filter>
       </C:comp-filter>
     </C:filter>
   </C:calendar-query>
```

```
>> Response <<

HTTP/1.1 207 Multi-Status
Date: Fri, 11 Nov 2005 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:"
               xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:response>
    <D:href>http://cal.example.com/bernard/work/abcd4.ics</D:href>
    <D:propstat>
      <D:prop>
        <D:getetag>"fffff-abcd4"</D:getetag>
        <C:calendar-data>BEGIN:VCALENDAR
   VERSION:2.0
   PRODID:-//Example Corp.//CalDAV Client//EN
   BEGIN:VTODO
   DTSTAMP:20060205T235300Z
   DUE;TZID=US/Eastern:20060106T120000
   LAST-MODIFIED:20060205T235308Z
   SEQUENCE:1
   STATUS:NEEDS-ACTION
   SUMMARY:Task #2
   UID:E10BA47467C5C69BB74E8720@example.com
   BEGIN:VALARM
   ACTION:AUDIO
   TRIGGER;RELATED=START:-PT10M
   END:VALARM
   END:VTODO
   END:VCALENDAR
        </C:calendar-data>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

### 7.7.6. Example: Retrieval of event by UID

In this example, the client requests the server to return the VEVENT component that has the UID property set to "DC6C50A017428C5216A2F1CD@example.com".

```
   >> Request <<

   REPORT /bernard/work/ HTTP/1.1
   Host: cal.example.com
   Depth: 1
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <C:calendar-query xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:prop xmlns:D="DAV:">
       <D:getetag/>
       <C:calendar-data/>
     </D:prop>
     <C:filter>
       <C:comp-filter name="VCALENDAR">
         <C:comp-filter name="VEVENT">
           <C:prop-filter name="UID">
             <C:text-match caseless="no">
              DC6C50A017428C5216A2F1CD@example.com</C:text-match>
           </C:prop-filter>
         </C:comp-filter>
       </C:comp-filter>
     </C:filter>
   </C:calendar-query>

   >> Response <<

   HTTP/1.1 207 Multi-Status
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <D:multistatus xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd3"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
```

```
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      ATTENDEE;PARTSTAT=ACCEPTED;ROLE=CHAIR:mailto:cyrus@example.com
      ATTENDEE;PARTSTAT=NEEDS-ACTION:mailto:lisa@example.com
      DTSTAMP:20060206T001220Z
      DTSTART;TZID=US/Eastern:20060104T100000
      DURATION:PT1H
      LAST-MODIFIED:20060206T001330Z
      ORGANIZER:mailto:cyrus@example.com
      SEQUENCE:1
      STATUS:TENTATIVE
      SUMMARY:Event #3
      UID:DC6C50A017428C5216A2F1CD@example.com
      X-ABC-GUID:E1CX5Dr-0007ym-Hz@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
   </D:multistatus>
```

### 7.7.7. Example: Retrieval of events by PARTSTAT

In this example, the client requests the server to return the VEVENT components that have the ATTENDEE property with the value "mailto:lisa@example.com" and for which the PARTSTAT parameter is set to "NEEDS-ACTION".

```
   >> Request <<

   REPORT /bernard/work/ HTTP/1.1
   Host: cal.example.com
   Depth: 1
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <C:calendar-query xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:prop xmlns:D="DAV:">
       <D:getetag/>
       <C:calendar-data/>
     </D:prop>
     <C:filter>
       <C:comp-filter name="VCALENDAR">
         <C:comp-filter name="VEVENT">
           <C:prop-filter name="ATTENDEE">
             <C:text-match
               caseless="yes">mailto:lisa@example.com</C:text-match>
             <C:param-filter name="PARTSTAT">
               <C:text-match caseless="yes">NEEDS-ACTION</C:text-match>
             </C:param-filter>
           </C:prop-filter>
         </C:comp-filter>
       </C:comp-filter>
     </C:filter>
   </C:calendar-query>

   >> Response <<

   HTTP/1.1 207 Multi-Status
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <D:multistatus xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd3"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
    VERSION:2.0
    PRODID:-//Example Corp.//CalDAV Client//EN
    BEGIN:VTIMEZONE
```

```
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      ATTENDEE;PARTSTAT=ACCEPTED;ROLE=CHAIR:mailto:cyrus@example.com
      ATTENDEE;PARTSTAT=NEEDS-ACTION:mailto:lisa@example.com
      DTSTAMP:20060206T001220Z
      DTSTART;TZID=US/Eastern:20060104T100000
      DURATION:PT1H
      LAST-MODIFIED:20060206T001330Z
      ORGANIZER:mailto:cyrus@example.com
      SEQUENCE:1
      STATUS:TENTATIVE
      SUMMARY:Event #3
      UID:DC6C50A017428C5216A2F1CD@example.com
      X-ABC-GUID:E1CX5Dr-0007ym-Hz@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
   </D:multistatus>
```

### 7.7.8. Example: Retrieval of events only

In this example, the client requests the server to return all VEVENT components.

```
>> Request <<

REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop xmlns:D="DAV:">
    <D:getetag/>
    <C:calendar-data/>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VEVENT"/>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>

>> Response <<

HTTP/1.1 207 Multi-Status
Date: Fri, 11 Nov 2005 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:"
               xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:response>
    <D:href>http://cal.example.com/bernard/work/abcd1.ics</D:href>
    <D:propstat>
      <D:prop>
        <D:getetag>"fffff-abcd1"</D:getetag>
        <C:calendar-data>BEGIN:VCALENDAR
   VERSION:2.0
   PRODID:-//Example Corp.//CalDAV Client//EN
   BEGIN:VTIMEZONE
   LAST-MODIFIED:20040110T032845Z
   TZID:US/Eastern
   BEGIN:DAYLIGHT
   DTSTART:20000404T020000
   RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
   TZNAME:EDT
   TZOFFSETFROM:-0500
   TZOFFSETTO:-0400
```

```
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      DTSTAMP:20060206T001102Z
      DTSTART;TZID=US/Eastern:20060102T100000
      DURATION:PT1H
      SUMMARY:Event #1
      Description:Go Steelers!
      UID:74855313FA803DA593CD579A@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd2.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd2"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
```

```
      END:VTIMEZONE
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART;TZID=US/Eastern:20060102T120000
      DURATION:PT1H
      RRULE:FREQ=DAILY;COUNT=5
      SUMMARY:Event #2
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART;TZID=US/Eastern:20060104T140000
      DURATION:PT1H
      RECURRENCE-ID;TZID=US/Eastern:20060104T120000
      SUMMARY:Event #2 bis
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART;TZID=US/Eastern:20060106T140000
      DURATION:PT1H
      RECURRENCE-ID;TZID=US/Eastern:20060106T120000
      SUMMARY:Event #2 bis bis
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd3"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
```

```
   END:DAYLIGHT
   BEGIN:STANDARD
   DTSTART:20001026T020000
   RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
   TZNAME:EST
   TZOFFSETFROM:-0400
   TZOFFSETTO:-0500
   END:STANDARD
   END:VTIMEZONE
   BEGIN:VEVENT
   ATTENDEE;PARTSTAT=ACCEPTED;ROLE=CHAIR:mailto:cyrus@example.com
   ATTENDEE;PARTSTAT=NEEDS-ACTION:mailto:lisa@example.com
   DTSTAMP:20060206T001220Z
   DTSTART;TZID=US/Eastern:20060104T100000
   DURATION:PT1H
   LAST-MODIFIED:20060206T001330Z
   ORGANIZER:mailto:cyrus@example.com
   SEQUENCE:1
   STATUS:TENTATIVE
   SUMMARY:Event #3
   UID:DC6C50A017428C5216A2F1CD@example.com
   X-ABC-GUID:E1CX5Dr-0007ym-Hz@example.com
   END:VEVENT
   END:VCALENDAR
        </C:calendar-data>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

### 7.7.9. Example: Attempt to query unsupported property

In this example, the client requests the server to return all VEVENT components that include an "X-ABC-GUID" property with a value matching "ABC".  However, the server does not support querying that non-standard property and instead returns and error response.

```
>> Request <<

REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop xmlns:D="DAV:">
    <D:getetag/>
    <C:calendar-data/>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VEVENT">
        <C:prop-filter name="X-ABC-GUID">
          <C:text-match>ABC</C:text-match>
        </C:prop-filter>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>

>> Response <<

HTTP/1.1 400 Bad Request
Date: Fri, 11 Nov 2005 09:32:12 GMT
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:error>
  <C:supported-filter>
    <C:prop-filter name="X-ABC-GUID"/>
  </C:supported-filter>
</D:error>
```

### 7.8. CALDAV:calendar-multiget Report

The CALDAV:calendar-multiget REPORT is used to retrieve specific calendar object resources from within a collection, if the Request-URI is a collection, or to retrieve a specific calendar object resource, if the Request-URI is a calendar object resource. This REPORT is similar to the CALDAV:calendar-query REPORT (see Section 7.7), except that it takes a list of DAV:href elements instead of a CALDAV:filter element to determine which calendar object

resources to return.

Support for the calendar-multiget REPORT is REQUIRED.

Marshalling:

The request body MUST be a CALDAV:calendar-multiget XML element (see Section 9.9). If the Request-URI is a collection resource, then the DAV:href elements MUST refer to resources within that collection, and they MAY refer to resources at any depth within the collection. As a result the "Depth" header MUST be ignored by the server and SHOULD NOT be sent by the client. If the Request-URI refers to a non-collection resource, then there MUST be a single DAV:href element that is equivalent to the Request-URI.

The response body for a successful request MUST be a DAV: multistatus XML element.

The response body for a successful CALDAV:calendar-multiget REPORT request MUST contain a DAV:response element for each calendar object resource referenced by the provided set of DAV:href elements. Calendar data is being returned in the CALDAV:calendar-data element inside the DAV:prop element.

In the case of an error accessing any of the provided DAV:href resources, the server MUST return the appropriate error status code in the DAV:status element of the corresponding DAV:response element.

Preconditions:

(CALDAV:supported-calendar-data): The attributes "content-type" and "version" of the CALDAV:calendar-data XML elements specify a media type supported by the server for calendar object resources.

Postconditions:

None.

### 7.8.1. Example: Successful CALDAV:calendar-multiget Report

In this example, the client requests the server to return specific properties of the VEVENT components referenced by specific URIs. In addition the DAV:getetag property is also requested and returned as part of the response. Note that in this example, the resource at http://cal.example.com/bernard/work/mtg1.ics does not exist, resulting in an error status response.

```
   >> Request <<

   REPORT /bernard/work/ HTTP/1.1
   Host: cal.example.com
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <C:calendar-multiget xmlns:D="DAV:"
                    xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:prop>
       <D:getetag/>
       <C:calendar-data/>
     </D:prop>
     <D:href>/bernard/work/abcd1.ics</D:href>
     <D:href>/bernard/work/mtg1.ics</D:href>
   </C:calendar-multiget>

   >> Response <<

   HTTP/1.1 207 Multi-Status
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <D:multistatus xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:response>
       <D:href>http://cal.example.com/bernard/work/abcd1.ics</D:href>
       <D:propstat>
         <D:prop>
           <D:getetag>"fffff-abcd1"</D:getetag>
           <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
```

```
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      DTSTAMP:20060206T001102Z
      DTSTART;TZID=US/Eastern:20060102T100000
      DURATION:PT1H
      SUMMARY:Event #1
      Description:Go Steelers!
      UID:74855313FA803DA593CD579A@example.com
      END:VEVENT
      END:VCALENDAR
           </C:calendar-data>
         </D:prop>
         <D:status>HTTP/1.1 200 OK</D:status>
       </D:propstat>
     </D:response>
     <D:response>
       <D:href>http://cal.example.com/bernard/work/mtg1.ics</D:href>
       <D:status>HTTP/1.1 404 Not Found</D:status>
     </D:response>
   </D:multistatus>
```

## 7.9. CALDAV:free-busy-query Report

The CALDAV:free-busy-query REPORT generates a VFREEBUSY component containing free busy information for all the calendar object resources targeted by the request and which have the CALDAV:read-free-busy or DAV:read privilege granted to the current user.

Only VEVENT components without a TRANSP property or with the TRANSP property set to "OPAQUE", and VFREEBUSY components SHOULD be considered to generate the free busy time information.

In the case of VEVENT components, the free or busy time type (FBTYPE) of the FREEBUSY properties in the returned VFREEBUSY component SHOULD be derived from the value of the TRANSP and STATUS properties as outlined in the table below:

| VEVENT | | VFREEBUSY |
|---|---|---|
| TRANSP | STATUS | FBTYPE |
| OPAQUE (default) | CONFIRMED (default) | BUSY |
| | CANCELLED | FREE |
| | TENTATIVE | BUSY-TENTATIVE |
| | x-name | BUSY or x-name |
| TRANSPARENT | CONFIRMED<br>CANCELLED<br>TENTATIVE<br>x-name | FREE |

Duplicate busy time periods with the same FBTYPE parameter value SHOULD NOT be specified in the returned VFREEBUSY component. Servers SHOULD coalesce consecutive or overlapping busy time period of the same type. Busy time periods with different FBTYPE parameter values MAY overlap.

Support for the CALDAV:free-busy-query REPORT is REQUIRED.

Marshalling:

The request body MUST be a CALDAV:free-busy-query XML element (see Section 9.10, which MUST contain exactly one CALDAV:time-range XML element, as defined in Section 9.8.

The request MAY include a Depth header. If no Depth header is included, Depth:0 is assumed.

The response body for a successful request MUST be an iCalendar object that contains exactly one VFREEBUSY component that describes the busy time intervals for the calendar object resources containing VEVENT or VFREEBUSY components that satisfy the Depth value and for which the current user is at least granted the CALDAV:read-free-busy privilege. If no calendar object resources are found to satisfy these conditions a VFREEBUSY component with no FREEBUSY property MUST be returned. This REPORT only returns busy time information. Free time information can be inferred from the returned busy time information.

If the current user is not granted the CALDAV:read-free-busy or DAV:read privileges on the Request-URI, the CALDAV:free-busy-query REPORT request MUST fail and return a 404 (Not Found) status value. This restriction will prevent users from discovering URLs of resources for which they are only granted the CALDAV:read-free-busy privilege.

The CALDAV:free-busy-query REPORT request can only be run against a collection (either a regular collection or a calendar collection). An attempt to run the report on a calendar object resource MUST fail and return a 403 (Forbidden) status value.

Preconditions:

None.

Postconditions:

(DAV:number-of-matches-within-limits): The number of matching calendar object resources must fall within server-specific, predefined limits. For example, this postcondition might fail if the specified CALDAV:time-range would cause an extremely large number calendar object resources to be considered to compute the response.

### 7.9.1. Example: Successful CALDAV:free-busy-query Report

In this example, the client requests the server to return free busy information on the calendar collection /bernard/work/, between 9:00 AM and 5:00 PM EST (2:00 PM and 10:00 PM UTC) on the 4th January 2006. The server responds indicating two busy time intervals of one hour, one of which is tentative.

>> Request <<

```
REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:free-busy-query xmlns:C="urn:ietf:params:xml:ns:caldav">
  <C:time-range start="20060104T140000Z"
                end="20060105T220000Z"/>
</C:free-busy-query>
```

```
   >> Response <<

   HTTP/1.1 200 OK
   Date: Fri, 11 Nov 2005 09:32:12 GMT
   Content-Type: text/calendar
   Content-Length: xxxx

   BEGIN:VCALENDAR
   VERSION:2.0
   PRODID:-//Example Corp.//CalDAV Server//EN
   BEGIN:VFREEBUSY
   DTSTAMP:20050125T090000Z
   DTSTART:20060104T140000Z
   DTEND:20060105T220000Z
   FREEBUSY;FBTYPE=BUSY-TENTATIVE:20060104T150000Z/PT1H
   FREEBUSY:20060104T190000Z/PT1H
   END:VFREEBUSY
   END:VCALENDAR
```

## 8. Guidelines

### 8.1. Client-to-client Interoperability

There are a number of actions clients can take which will be legal (the server will not return errors) but which can degrade interoperability with other client implementations accessing the same data. For example, a recurrence rule could be replaced with a set of recurrence dates, a single recurring event could be replaced with a set of independent resources to represent each recurrence, or the start/end time values can be translated from the original time zone to another time zone. Although this advice amounts to iCalendar interoperability best practices and is not limited only to CalDAV usage, interoperability problems are likely to be more evident in CalDAV use cases.

### 8.2. Synchronization Operations

WebDAV already provides functionality required to synchronize a collection or set of collections, make changes offline, and a simple way to resolve conflicts when reconnected. ETags are the key to making this work, but these are not required of all WebDAV servers. Since offline functionality is more important to calendar applications than to some other WebDAV applications, CalDAV servers MUST support ETags as specified in Section 5.3.4.

### 8.2.1. Use of Reports

#### 8.2.1.1. Restrict the Time Range

The REPORTs provided in CalDAV can be used by clients to optimize their performance in terms of network bandwidth usage, and resource consumption on the local client machine. Both are certainly major considerations for mobile or handheld devices with limited capacity, but they are also relevant to desktop client applications in cases where the calendar collections contain large amounts of data.

Typically clients present calendar data to users in views that span a finite time interval, so whenever possible clients should only retrieve calendar components from the server using CALDAV:calendar-query REPORT combined with a CALDAV:time-range element to limit the set of returned components to just those needed to populate the current view.

#### 8.2.1.2. Synchronize by Time Range

Typically in a calendar, historical data (events, to-dos etc. that have completed prior to the current date) do not change, though they may be deleted. As a result, a client can speed up the synchronization process by only considering data for the present time and the future up to a reasonable limit (e.g., one week, one month). If the user then tries to examine a portion of the calendar outside of the range that has been synchronized, the client can perform another synchronization operation on the new time interval being examined. This "just-in-time" synchronization can minimize bandwidth for common user interaction behaviors.

#### 8.2.1.3. Synchronization Process

If a client wants to support calendar data synchronization, as opposed to downloading calendar data each time it is needed, it needs to cache the calendar object resource's URI and ETag along with the actual calendar data. While the URI remains static for the lifetime of the calendar object resource, the ETag will change with each successive change to the calendar object resource. Thus to synchronize a local data cache with the server, the client can first fetch the URI/ETag pairs for the time interval being considered, and compare those results with the cached data. Any cached component whose ETag differs from that on the server needs to be refreshed.

In order to properly detect the changes between the server and client data, the client will need to keep a record of which calendar object resources have been created, changed or deleted since the last synchronization operation so that it can reconcile those changes with

the data on the server.

Here's an example of how to do that:

The client issues a CALDAV:calendar-query REPORT request for a specific time range, and asks for only the DAV:getetag property to be returned:

```
REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <D:getetag/>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:comp-filter name="VEVENT">
        <C:time-range start="20040902T000000Z"
                        end="20040903T000000Z"/>
      </C:comp-filter>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>
```

The client then uses the results to determine which calendar object resources have changed, been created or deleted on the server and how those relate to locally cached calendar object resources that may have changed, been created or deleted.  If the client determines that there are calendar object resources on the server that need to be fetched, the client issues a CALDAV:calendar-multiget REPORT request to fetch their calendar data:

```
   REPORT /bernard/work/ HTTP/1.1
   Host: cal.example.com
   Depth: 1
   Content-Type: application/xml; charset="utf-8"
   Content-Length: xxxx

   <?xml version="1.0" encoding="utf-8" ?>
   <C:calendar-multiget xmlns:D="DAV:"
                        xmlns:C="urn:ietf:params:xml:ns:caldav">
     <D:prop>
       <D:getetag/>
       <C:calendar-data/>
     </D:prop>
     <D:href>/bernard/work/abcd1.ics</D:href>
     <D:href>/bernard/work/mtg1.ics</D:href>
   </C:calendar-multiget>
```

### 8.2.2. Restrict the Properties Returned

Clients may not need all the calendar properties of a calendar object resource when presenting information to the user. Since some calendar property values can be large (e.g., ATTACH or ATTENDEE) clients can choose to restrict the calendar properties to be returned in a calendaring REPORT request to those it knows it will use.

However, if a client needs to make a change to a calendar object resource, it can only change the entire calendar object resource via a PUT request. There is currently no way to incrementally make a change to a set of calendar properties of a calendar object resource. As a result the client will have to get the entire calendar object resource that is being changed.

### 8.3. Use of Locking

WebDAV locks can be used to prevent two clients modifying the same resource from either overwriting each others' changes (though that problem can also be solved by using ETags) or wasting time making changes that will conflict with another set of changes. In a multi-user calendar system, an interactive calendar client could lock an event while the user is editing the event, and unlock the event when the user finishes or cancels. Locks can also be used to prevent changes while data is being reorganized. For example, a calendar client might lock two calendar collections prior to moving a bunch of calendar resources from one to another.

Clients are responsible for requesting a lock timeout period that is appropriate to the use case. When the user explicitly decides to reserve a resource and prevent other changes, a long timeout might be

appropriate, but in cases when the client automatically decides to lock the resource the timeout should be short (and the client can always refresh the lock should it need to). A short lock timeout means that if the client is unable to remove the lock, the other calendar users aren't prevented from making changes.

### 8.4. Finding calendars

Much of the time a calendar client (or agent) will discover a new calendar's location by being provided directly with the URL. E.g., a user will type his or her own calendar location into client configuration information, or copy and paste a URL from email into the calendar application. The client need only confirm that the URL points to a resource which is a calendar collection. The client may also be able to browse WebDAV collections to find calendar collections.

The choice of HTTP URLs means that calendar object resources are backward compatible with existing software, but does have the disadvantage that existing software does not usually know to look at the OPTIONS response to that URL to determine what can be done with it. This is somewhat of a barrier for WebDAV usage as well as with CalDAV usage. This specification does not offer a way through this other than making the information available in the OPTIONS response should this be requested.

For calendar sharing and scheduling use cases, one might wish to find the calendar belonging to another user. If the other user has a calendar in the same repository, that calendar can be found by using the principal namespace required by WebDAV ACL support. For other cases, the authors have no universal solution but implementors can consider whether to use vCard [RFC2426] or LDAP [RFC2251] standards together with calendar attributes [RFC2739].

Because CalDAV requires servers to support WebDAV ACL [RFC3744] including principal namespaces, and with the addition of the CALDAV:calendar-home-set property, there are a couple options for CalDAV clients to find one's own calendar or another user's calendar.

In this case, a DAV:principal-match REPORT is used to find a named property (the CALDAV:calendar-home-set) on the Principal-URL of the current user. Using this, a WebDAV client can learn "who am I" and "where are my calendars". The REPORT request body looks like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<D:principal-match xmlns:D="DAV:">
  <D:self/>
  <D:prop>
    <C:calendar-home-set
       xmlns:C="urn:ietf:params:xml:ns:caldav"/>
  </D:prop>
</D:principal-match>
```

To find other users calendars, the DAV:principal-property-search REPORT can be used to filter on some properties and return others. To search for a calendar owned by a user named "Laurie", the REPORT request body would look like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<D:principal-property-search xmlns:D="DAV:">
  <D:property-search>
    <D:prop>
      <D:displayname/>
    </D:prop>
    <D:match>Laurie</D:match>
  </D:property-search>
  <D:prop>
    <C:calendar-home-set
       xmlns:C="urn:ietf:params:xml:ns:caldav"/>
    <D:displayname/>
  </D:prop>
</D:principal-property-search>
```

The server performs a case-sensitive or caseless search for a matching string subset of "Laurie" within the DAV:displayname property. Thus, the server might return "Laurie Dusseault", "Laurier Desruisseaux" or "Wilfrid Laurier" all as matching DAV:displayname values, and the calendars for each of these.

### 8.5. Storing and Using Attachments

CalDAV clients MAY create attachments in calendar components either as inline or external. This section contains some guidelines on creating and managing attachments.

#### 8.5.1. Inline attachments

CalDAV clients MUST support inline attachments as specified in iCalendar [RFC2445]. CalDAV servers MUST support inline attachments, so clients can rely on being able to create attachments this way. On the other hand, inline attachments have some drawbacks:

- Servers MAY impose limitations on the size of calendar object resources (i.e., refusing PUT requests of very large iCalendar objects).

- Servers MAY impose storage quota limitations on calendar collections (See [I-D.ietf-webdav-quota]).

- Any change to a calendar object resource containing an attachment requires the entire attachment to be re-uploaded.

- Clients synchronizing a changed calendar object resource have to download the entire calendar object resource even if the attachment is unchanged.

### 8.5.2. External attachments

CalDAV clients MUST support external attachments: if the client accesses any calendar object resource it MUST be capable of also accessing the external attachment if one exists. An external attachment could be:

- In a collection in the calendar collection containing the calendar object resource;

- Somewhere else in the same repository that hosts the calendar collection; or

- On an HTTP or FTP server elsewhere.

CalDAV servers MAY provide support for child collections in calendar collections. CalDAV servers MAY allow the MKCOL method to create child collections in calendar collections. Child collections of calendar collections MAY contain any type of resource except calendar collections which they MUST NOT contain. Some CalDAV servers won't allow child collections in calendar collections, and it may be possible on such a server to discover other locations where attachments can be stored.

Clients are entirely responsible for maintaining reference consistency with calendar components that link to external attachments. A client deleting a calendar component with an external attachment might therefore also delete the attachment if that's appropriate, however appropriateness can be very hard to determine. A new component might easily reference some pre-existing Web resource which is intended to have independent existence from the calendar component (the "attachment" could be a major proposal to be discussed in a meeting, for instance). Best practices will probably emerge and should probably be documented but for now clients should be wary of

engaging in aggressive "cleanup" of external attachments. A client could involve the user in making decisions about removing unreferenced documents, or a client could be conservative in only deleting attachments it had created.

Also, clients are responsible for consistency of permissions when using external attachments. One reason for servers to support the storage of attachments within child collections of calendar collections is that ACL inheritance might make it easier to grant the same permissions to attachments that are granted on the calendar collection. Otherwise, it can be very difficult to keep permissions synchronized. With attachments stored on separate repositories, it can be impossible to keep permissions consistent -- the two repositories may not support the same permissions or have the same set of principals. Some systems have used tickets or other anonymous access control mechanisms to provide partially satisfactory solutions to these kinds of problems.

### 8.6. Storing and Using Alarms

Note that all CalDAV calendar collections (including those which the user might treat as public or group calendars) can contain alarm information on events and to-dos. Users can synchronize a calendar between multiple devices and decide to have alarms execute on a different device than the device that created the alarm. Not all alarm action types are completely interoperable (e.g., those which name a sound file to play).

> When the action is "AUDIO", and the client is configured to execute the alarm, the client SHOULD play the suggested sound if it's available or play another sound, but SHOULD NOT rewrite the alarm just to replace the suggested sound with a sound that's locally available.
>
> When the action is "DISPLAY", and the client is configured to execute the alarm, the client SHOULD execute a display alarm by displaying either according to the suggested description or some reasonable replacement, but SHOULD NOT rewrite the alarm for its own convenience.
>
> When the action is "EMAIL", and the client is incapable of sending email, it SHOULD ignore the alarm but MUST continue to synchronize the alarm itself.
>
> This specification makes no recommendations about executing alarm of type PROCEDURE except to note that clients are advised to take care to avoid creating security holes by executing these.

Non-interoperable alarm information (e.g., should somebody define a color to be used in a display alarm) should be put in non-standard properties inside the VALARM component in order to keep the basic alarm usable on all devices.

Clients that allow changes to calendar object resources MUST synchronize the alarm data that already exists in the resources. Clients MAY execute alarms that are downloaded in this fashion, possibly based on user preference. If a client is only doing read operations on a calendar and there is no risk of losing alarm information, then the client MAY discard alarm information.

This specification makes no attempt to provide multi-user alarms on group calendars or to find out who an alarm is intended for. Addressing those issues might require extensions to iCalendar, for example to store alarms per-user or indicate which user a VALARM was intended for. In the meantime, clients might maximize interoperability by generally not uploading alarm information to public, group or resource calendars.

# 9. XML Element Definitions

## 9.1. CALDAV:calendar XML Element

Name: calendar

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

Purpose: Specifies the resource type of a calendar collection.

Description: See Section 4.2.

Definition:

```
      <!ELEMENT calendar EMPTY>
```

## 9.2. CALDAV:mkcalendar XML Element

Name: mkcalendar

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

Purpose: Specifies a request that lists the WebDAV property values to be set for a calendar collection resource.

Description: See Section 5.3.1.

Definition:

```
      <!ELEMENT mkcalendar (DAV:set)>
```

### 9.3. CALDAV:mkcalendar-response XML Element

Name: mkcalendar-response

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

Purpose: Specifies a response body for a successful MKCALENDAR request.

Description: See Section 5.3.1.

Definition:

```
      <!ELEMENT mkcalendar-response ANY>
```

### 9.4. CALDAV:calendar-query XML Element

Name: calendar-query

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

Purpose: Defines a REPORT for querying calendar object resources.

Description: See Section 7.7.

Definition:

```
      <!ELEMENT calendar-query ((DAV:allprop |
                                 DAV:propname |
                                 DAV:prop)?, filter, timezone?)>
```

### 9.5. CALDAV:calendar-data XML Element

Name: calendar-data

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

Purpose: Used to (1) specify a supported media type for calendar object resources when nested in the CALDAV:supported-calendar-data property; (2) specify which parts of a calendar object resource should be returned by a given calendaring REPORT; and (3) specify the content of a calendar object resource in a response to a

calendaring REPORT.

Description: When nested in the CALDAV:supported-calendar-data property, the CALDAV:calendar-data XML element specifies a media type supported by the CalDAV server for calendar object resources.

When used in a calendaring REPORT request, the CALDAV:calendar-data XML element specifies which parts of calendar object resources need to be returned in the response.  If the CALDAV:calendar-data XML element doesn't contain any CALDAV:comp element, calendar object resources will be returned in their entirety.

Finally, when used in a calendaring REPORT response, the CALDAV:calendar-data XML element specifies the content of a calendar object resource.  Given that XML parsers normalizes the two-character sequence CRLF (US-ASCII decimal 13 and US-ASCII decimal 10) to a single LF character (US-ASCII decimal 10), the CR character (US-ASCII decimal 13) MAY be omitted in calendar object resources specified in the CALDAV:calendar-data XML element. Furthermore, calendar object resources specified in the CALDAV:calendar-data XML element MAY be invalid per their media type specification if the CALDAV:calendar-data XML element part of the calendaring REPORT request did not specify required properties (e.g., UID, DTSTAMP, etc.) or specified a CALDAV:prop XML element with the "novalue" attribute set to "yes".

Note: The CALDAV:calendar-data XML element is specified in requests and responses inside the DAV:prop XML element as if it were a WebDAV property.  However, the CALDAV:calendar-data XML element is not a WebDAV property and as such it is not returned in PROPFIND responses nor used in PROPPATCH requests.

Definition:

```
<!ELEMENT calendar-data ((comp?, (expand |
                                  limit-recurrence-set)?,
                                  limit-freebusy-set?) |
                         #PCDATA)?>
PCDATA value: iCalendar object

<!ATTLIST calendar-data content-type CDATA "text/calendar">
                        version CDATA "2.0">
content-type value: a MIME media type
version value: a version string
```

### 9.5.1. CALDAV:comp XML Element

Name: comp

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

Purpose: Defines which component types to return.

Description: The name value is a calendar component name (e.g., "VEVENT").

Definition:

```
      <!ELEMENT comp (((allprop | prop*), allcomp) |
                      ((allprop | prop*), comp*))>

      <!ATTLIST comp name CDATA #REQUIRED>
      name value: a calendar component name
```

Note: The CALDAV:prop and CALDAV:allprop elements have the same name as the DAV:prop and DAV:allprop elements defined in [RFC2518]. However, the CALDAV:prop and CALDAV:allprop element are defined in the "urn:ietf:params:xml:ns:caldav" namespace instead of the "DAV:" namespace.

### 9.5.2. CALDAV:allcomp XML Element

Name: allcomp

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

Purpose: Specifies that all components shall be returned.

Description: The CALDAV:allcomp XML element can be used when the client wants all types of components returned by a calendaring REPORT request.

Definition:

```
      <!ELEMENT allcomp EMPTY>
```

### 9.5.3. CALDAV:allprop XML Element

Name: allprop

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

Purpose: Specifies that all properties shall be returned.

Description: The CALDAV:allprop XML element can be used when the client wants all properties of components returned by a calendaring REPORT request.

Definition:

```
      <!ELEMENT allprop EMPTY>
```

Note: The CALDAV:allprop element has the same name as the DAV:allprop element defined in [RFC2518]. However, the CALDAV:allprop element is defined in the "urn:ietf:params:xml:ns:caldav" namespace instead of the "DAV:" namespace.

### 9.5.4. CALDAV:prop XML Element

Name: prop

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

Purpose: Defines which properties to return in the response.

Description: The "name" attribute specifies the name of the calendar property to return (e.g., "ATTENDEE"). The "novalue" attribute can be used by clients to request that the actual value of the property not be returned (if the "novalue" attribute is set to "yes"). In that case the server will return just the iCalendar property name and any iCalendar parameters and a trailing ":" without the subsequent value data.

Definition:

```
      <!ELEMENT prop EMPTY>

      <!ATTLIST prop name CDATA #REQUIRED
                  novalue (yes | no) "no">
      name value: a calendar property name
      novalue value: "yes" or "no"
```

Note: The CALDAV:prop element has the same name as the DAV:prop element defined in [RFC2518]. However, the CALDAV:prop element is defined in the "urn:ietf:params:xml:ns:caldav" namespace instead of the "DAV:" namespace.

### 9.5.5. CALDAV:expand XML Element

Name: expand

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

Purpose: Forces the server to expand recurring components into individual recurrence instances.

Description: The CALDAV:expand XML element specifies that for a given calendaring REPORT request the server MUST expand the recurrence set into calendar components that define exactly one recurrence instance and MUST return only those whose scheduled time intersect a specified time range. The "start" attribute specifies the inclusive start of the time range, and the "end" attribute specifies the non-inclusive end of the time range. Both attributes are specified as date with UTC time value. The server MUST use the same logic as defined for CALDAV:time-range to determine if a recurrence instance intersects the specified time range. Recurring components, other than the initial instance, MUST include a RECURRENCE-ID property indicating which instance they refer to. The returned calendar components MUST NOT use recurrence properties (i.e., EXDATE, EXRULE, RDATE and RRULE) and MUST NOT have reference to or include VTIMEZONE components. Date and local time with reference to time zone information MUST be converted into date with UTC time.

Definition:

```
<!ELEMENT expand EMPTY>

<!ATTLIST expand start CDATA #REQUIRED
                 end   CDATA #REQUIRED>
start value: an iCalendar "date with UTC time"
end value: an iCalendar "date with UTC time"
```

### 9.5.6. CALDAV:limit-recurrence-set XML Element

Name: limit-recurrence-set

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

Purpose: Specifies a time range to limit the set of "overridden components" returned by the server.

Description: The CALDAV:limit-recurrence-set XML element specifies that for a given calendaring REPORT request the server MUST only return the "master component" as well as the "overridden components" that specify one or more recurrence instances whose current scheduled time or original scheduled time intersect a specified time range. The "start" attribute specifies the inclusive start of the time range, and the "end" attribute specifies the non-inclusive end of the time range. Both attributes are specified as date with UTC time value. The server MUST use the same logic as defined for CALDAV:time-range to determine if the current or original scheduled time of an "overridden" recurrence instance intersect the specified time range. Overridden components that have a RANGE parameter on their RECURRENCE-ID property may specify one or more instances in the recurrence set, and some of those instances may fall within the specified time range, or may have originally fallen within the specified time range prior to being overridden. If that is the case, the overridden component MUST be included in the results as it has a direct impact on the interpretation of instances within the specified time range.

Definition:

```
<!ELEMENT limit-recurrence-set EMPTY>

<!ATTLIST limit-recurrence-set start CDATA #REQUIRED
                               end   CDATA #REQUIRED>
start value: an iCalendar "date with UTC time"
end value: an iCalendar "date with UTC time"
```

### 9.5.7. CALDAV:limit-freebusy-set XML Element

Name: limit-freebusy-set

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

Purpose: Specifies a time range to limit the set of FREEBUSY values returned by the server.

Description: The CALDAV:limit-freebusy-set XML element specifies that for a given calendaring REPORT request the server MUST only return the FREEBUSY property values of a VFREEBUSY component that intersect a specified time range. The "start" attribute specifies the inclusive start of the time range, and the "end" attribute specifies the non-inclusive end of the time range. Both attributes are specified as "date with UTC time" value. The server MUST use the same logic as defined for CALDAV:time-range to determine if a FREEBUSY property value intersect the specified

time range.

Definition:

```
<!ELEMENT limit-freebusy-set EMPTY>

<!ATTLIST limit-freebusy-set start CDATA #REQUIRED
                             end   CDATA #REQUIRED>
start value: an iCalendar "date with UTC time"
end value: an iCalendar "date with UTC time"
```

## 9.6. CALDAV:filter XML Element

Name: filter

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

Purpose: Specifies a filter to limit the set of calendar components returned by the server.

Description: The CALDAV:filter XML element specifies the search filter used to limit the calendar components returned by a calendaring REPORT request.

Definition:

```
<!ELEMENT filter (comp-filter)>
```

### 9.6.1. CALDAV:comp-filter XML Element

Name: comp-filter

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

Purpose: Specifies search criteria on calendar components.

Description: The CALDAV:comp-filter XML element specifies the queried calendar component type (e.g., "VEVENT"). A calendar object resource is said to match a CALDAV:comp-filter if it contains calendar components of the type specified by the "name" attribute, and that it contains at least one recurrence instance scheduled to overlap a given time range if a CALDAV:time-range XML element is specified, and that any CALDAV:prop-filter and CALDAV:comp-filter child elements also matches.

Definition:

```
      <!ELEMENT comp-filter (time-range?, prop-filter*,
                             comp-filter*)>

      <!ATTLIST comp-filter name CDATA #REQUIRED>
      name value: a calendar component name (e.g., "VEVENT")
```

### 9.6.2. CALDAV:prop-filter XML Element

Name: prop-filter

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

Purpose: Specifies search criteria on calendar properties.

Description: The CALDAV:prop-filter XML element specifies a search criteria on a specific calendar property (e.g., CATEGORIES) in the scope of a given CALDAV:comp-filter. A calendar component is said to match a CALDAV:prop-filter if it defines the property specified by the "name" attribute, and that it matches the CALDAV:time-range or CALDAV:text-match conditions if specified, and that any CALDAV: param-filter child elements also matches.

Definition:

```
      <!ELEMENT prop-filter ((time-range | text-match)?,
                            param-filter*)>

      <!ATTLIST prop-filter name CDATA #REQUIRED>
      name value: a calendar property name (e.g., "ATTENDEE")
```

### 9.6.3. CALDAV:param-filter XML Element

Name: param-filter

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

Purpose: Limits the search to specific parameter values.

Description: The CALDAV:param-filter XML element specifies a search criteria on a specific calendar property parameter (e.g., PARTSTAT) in the scope of a given CALDAV:prop-filter. A calendar property is said to match a CALDAV:param-filter if it defines the parameter specified by the "name" attribute, and that it matches the CALDAV:text-match condition if specified.

Definition:

```
      <!ELEMENT param-filter (text-match?)>

      <!ATTLIST param-filter name CDATA #REQUIRED>
      name value: a property parameter name (e.g., "PARTSTAT")
```

### 9.6.4. CALDAV:text-match XML Element

Name: text-match

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

Purpose: Specifies a substring match on a property or parameter value.

Description: The CALDAV:text-match XML element specifies text used for a substring match against the property or parameter value specified in a calendaring REPORT request. The "caseless" attribute indicates whether the match is case-sensitive (value set to "no") or case-insensitive (value set to "yes"). The default value is server-specified. Caseless matching SHOULD be implemented as defined in section 5.18 of the Unicode Standard ([UNICODE4]). Support for the "caseless" attribute is REQUIRED. A server MAY ignore the caseless attribute when applied to enumerated iCalendar property or parameter values, and default to caseless matching for those values, since they are defined as being case-insensitive in iCalendar.

Definition:

```
      <!ELEMENT text-match (#PCDATA)>
      PCDATA value: string

      <!ATTLIST text-match caseless (yes | no) #IMPLIED>
```

### 9.7. CALDAV:timezone XML Element

Name: timezone

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

Purpose: Specifies the time zone component to use when determining the results of a report.

Description: The CALDAV:timezone XML element specifies that for a given calendaring REPORT request the server MUST rely on the specified VTIMEZONE component instead of the CALDAV:calendar-timezone property of the calendar collection in which the calendar object resource is contained to resolve "date" values and "date with local time" values (i.e., floating time) to "date with UTC time" values. The server will require this information to determine if a calendar component scheduled with "date" values or "date with local time" values intersect a CALDAV:time-range specified in a CALDAV:calendar-query REPORT.

Definition:

```
<!ELEMENT timezone (#PCDATA)>
PCDATA value: an iCalendar object with exactly one VTIMEZONE
```

### 9.8. CALDAV:time-range XML Element

Name: time-range

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

Purpose: Specifies a time range to limit the set of calendar components returned by the server.

Description: The CALDAV:time-range XML element specifies that for a given calendaring REPORT request the server MUST only return the calendar object resources that, depending on the context, have a component or property or parameter whose value intersect a specified time range. The "start" attribute specifies the inclusive start of the time range, and the "end" attribute specifies the non-inclusive end of the time range. Both attributes are specified as "date with UTC time" value. While the "start" and "end" attributes are not required to allow time ranges opened at one end, at least one of them MUST be specified in the CALDAV:time-range element.

A VEVENT component overlaps a given time range if:

```
(DTSTART <= start AND DTEND > start) OR
(DTSTART <= start AND DTSTART+DURATION > start) OR
(DTSTART >= start AND DTSTART < end) OR
(DTEND   > start AND DTEND <= end)
```

A VEVENT component with no DTSTART and DTEND properties does not overlap any time range.

A VTODO component overlaps a given time range if:

```
(DTSTART <= start AND DUE >= start) OR
(DTSTART <= start AND DTSTART+DURATION > start) OR
(DTSTART >= start AND DTSTART < end) OR
(DUE     >= start AND DUE < end)
```

A VTODO component with no DTSTART and DUE properties does not overlap any time range.

A VJOURNAL component overlaps a given time range if:

```
DTSTART >= start AND DTSTART < end
```

A VJOURNAL component with no DTSTART property does not overlap any time range.

A VFREEBUSY component overlaps a given time range if for any of its FREEBUSY property value the following condition holds:

```
freebusy-period-start >= start AND freebusy-period-end < end
```

A VFREEBUSY component with no FREEBUSY property does not overlap any time range.

A VALARM component overlaps a given time range if:

```
trigger-time >= start AND trigger-time < end
```

A VALARM component can be defined such that it triggers repeatedly. Such a VALARM component is said to overlap a given time range if at least one of its trigger overlap the time range.

The calendar properties COMPLETED, CREATED, DTSTAMP and LAST-MODIFIED overlaps a given time range

```
date-time >= start AND date-time < end
```

The semantic of CALDAV:time-range is not defined for any other calendar properties.

Definition:

```
<!ELEMENT time-range EMPTY>

<!ATTLIST time-range start CDATA #IMPLIED
                     end   CDATA #IMPLIED>
start value: an iCalendar "date with UTC time"
```

      end value: an iCalendar "date with UTC time"

### 9.9. CALDAV:calendar-multiget XML Element

Name: calendar-multiget

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

Purpose: CalDAV REPORT used to retrieve specific calendar object resources.

Description: See Section 7.8.

Definition:

```
<!ELEMENT calendar-multiget ((DAV:allprop |
                             DAV:propname |
                             DAV:prop)?, DAV:href+)>
```

### 9.10. CALDAV:free-busy-query XML Element

Name: free-busy-query

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

Purpose: CalDAV REPORT used to generate a VFREEBUSY to determine busy time over a specific time range.

Description: See Section 7.9.

Definition:

```
<!ELEMENT free-busy-query (time-range)>
```

## 10. Internationalization Considerations

CalDAV allows internationalized strings to be stored and retrieved for the description of calendar collections (see Section 5.2.1).

## 11. Security Considerations

HTTP protocol transactions are sent in the clear over the network unless protection from snooping is negotiated. This can be accomplished by use of TLS as defined in [RFC2818]. In particular, HTTP Basic authentication MUST NOT be used unless TLS is in effect.

Servers MUST take adequate precautions to ensure malicious clients cannot consume excessive server resources (CPU, memory, disk, etc.) through carefully crafted reports. For example, a client could upload an event with a recurrence rule that specifies a recurring event occurring every second for the next 100 years which would result in approximately 3 x 10^9 instances! A REPORT that asks for recurrences to be expanded over that range would likely constitute a denial-of-service attack on the server.

When creating new resources (including calendar collections), clients MUST ensure that the resource name (the last path segment of the resource URI) assigned to the new resource does not expose any data from within the iCalendar resource itself and information about the nature of a calendar collection. This is required to ensure that the presence of a specific iCalendar component or nature of components in a collection cannot be inferred based on the name of a resource.

Security considerations described in iCalendar [RFC2445] and iTIP [RFC2446] are also applicable to CalDAV.

Beyond these, CalDAV does not raise any security considerations that are not present in HTTP [RFC2616] and WebDAV [RFC2518], [RFC3253], [RFC3744], as discussed in those documents.

## 12. IANA Consideration

This document uses one new URN to identify a new XML namespace. The URN conforms to a registry mechanism described in [RFC3688].

### 12.1. Namespace Registration

Registration request for the CalDAV namespace:

URI: urn:ietf:params:xml:ns:caldav

Registrant Contact: See the "Author's Address" section of this document.

XML: None. Namespace URIs do not represent an XML specification.

## 13. Acknowledgements

The authors would like to thank the following individuals for contributing their ideas and support for writing this specification: Michael Arick, Mario Bonin, Chris Bryant, Scott Carr, Mike Douglass, Helge Hess, Dan Mosedale, Kervin L. Pierre, Julian F. Reschke, Mike

Shaver, Simon Vaillancourt, Wilfredo Sanchez Vega and Jim Whitehead.

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

## 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.

[RFC2246] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[RFC2445] Dawson, F. and Stenerson, D., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", RFC 2445, November 1998.

[RFC2446] Silverberg, S., Mansour, S., Dawson, F., and R. Hopson, "iCalendar Transport-Independent Interoperability Protocol (iTIP) Scheduling Events, BusyTime, To-dos and Journal Entries", RFC 2446, November 1998.

[RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S., and D. Jensen, "HTTP Extensions for Distributed Authoring -- WEBDAV", RFC 2518, February 1999.

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

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

[RFC3253] Clemm, G., Amsden, J., Ellison, T., Kaler, C., and J. Whitehead, "Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)", RFC 3253, March 2002.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

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

[UNICODE4]
The Unicode Consortium, "The Unicode Standard, Version 4.0", Addison-Wesley, Boston, MA. ISBN 0-321-18578-1, August 2003, <http://www.unicode.org/versions/Unicode4.0.0/>.

[W3C.REC-xml-20040204]
Yergeau, F., Paoli, J., Sperberg-McQueen, C., Bray, T., and E. Maler, "Extensible Markup Language (XML) 1.0 (Third Edition)", W3C REC REC-xml-20040204, February 2004.

## 14.2. Informative References

[I-D.ietf-webdav-quota]
Korver, B. and L. Dusseault, "Quota and Size Properties for DAV Collections", draft-ietf-webdav-quota-07 (work in progress), April 2005.

[RFC2251] Wahl, M., Howes, T., and S. Kille, "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997.

[RFC2426] Dawson, F. and T. Howes, "vCard MIME Directory Profile", RFC 2426, September 1998.

[RFC2739] Small, T., Hennessy, D., and F. Dawson, "Calendar Attributes for vCard and LDAP", RFC 2739, January 2000.

## Appendix A. CalDAV Method Privilege Table (Normative)

The following table extends the WebDAV Method Privilege Table specified in Appendix B of [RFC3744].

| METHOD | PRIVILEGES |
|---|---|
| MKCALENDAR | DAV:bind |
| REPORT | DAV:read or CALDAV:read-free-busy (on all referenced resources) |

## Appendix B. Calendar collections used in the examples

This appendix shows the calendar object resources contained in the calendar collection queried in the examples throughout this document.

The content of the calendar collection is being shown as it would be

returned by a CALDAV:calendar-query REPORT request designed to return all the calendar data in the collection:

>> Request <<

```
REPORT /bernard/work/ HTTP/1.1
Host: cal.example.com
Depth: 1
Content-Type: application/xml; charset="utf-8"
Content-Length: xxxx
```

>> Response <<

```
<?xml version="1.0" encoding="utf-8" ?>
<C:calendar-query xmlns:D="DAV:"
                  xmlns:C="urn:ietf:params:xml:ns:caldav">
  <D:prop>
    <D:getetag/>
    <C:calendar-data/>
  </D:prop>
  <C:filter>
    <C:comp-filter name="VCALENDAR">
      <C:allprop/>
      <C:allcomp/>
    </C:comp-filter>
  </C:filter>
</C:calendar-query>

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:"
               xmlns:C="urn:ietf:params:xml:ns:caldav">

  <D:response>
    <D:href>http://cal.example.com/bernard/work/abcd1.ics</D:href>
    <D:propstat>
      <D:prop>
        <D:getetag>"fffff-abcd1"</D:getetag>
        <C:calendar-data>BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VTIMEZONE
LAST-MODIFIED:20040110T032845Z
TZID:US/Eastern
BEGIN:DAYLIGHT
DTSTART:20000404T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZNAME:EDT
TZOFFSETFROM:-0500
```

```
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      DTSTAMP:20060206T001102Z
      DTSTART;TZID=US/Eastern:20060102T100000
      DURATION:PT1H
      SUMMARY:Event #1
      Description:Go Steelers!
      UID:74855313FA803DA593CD579A@example.com
      END:VEVENT
      END:VCALENDAR
      </C:calendar-data>
            </D:prop>
            <D:status>HTTP/1.1 200 OK</D:status>
          </D:propstat>
        </D:response>

        <D:response>
          <D:href>http://cal.example.com/bernard/work/abcd2.ics</D:href>
          <D:propstat>
            <D:prop>
              <D:getetag>"fffff-abcd2"</D:getetag>
              <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
      TZOFFSETFROM:-0400
```

```
      TZOFFSETTO:-0500
      END:STANDARD
      END:VTIMEZONE
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART;TZID=US/Eastern:20060102T120000
      DURATION:PT1H
      RRULE:FREQ=DAILY;COUNT=5
      SUMMARY:Event #2
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      BEGIN:VEVENT
      DTSTAMP:20060206T001121Z
      DTSTART;TZID=US/Eastern:20060104T140000
      DURATION:PT1H
      RECURRENCE-ID;TZID=US/Eastern:20060104T120000
      SUMMARY:Event #2 bis
      UID:00959BC664CA650E933C892C@example.com
      END:VEVENT
      END:VCALENDAR
      </C:calendar-data>
            </D:prop>
            <D:status>HTTP/1.1 200 OK</D:status>
          </D:propstat>
        </D:response>

        <D:response>
          <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
          <D:propstat>
            <D:prop>
              <D:getetag>"fffff-abcd3"</D:getetag>
              <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTIMEZONE
      LAST-MODIFIED:20040110T032845Z
      TZID:US/Eastern
      BEGIN:DAYLIGHT
      DTSTART:20000404T020000
      RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
      TZNAME:EDT
      TZOFFSETFROM:-0500
      TZOFFSETTO:-0400
      END:DAYLIGHT
      BEGIN:STANDARD
      DTSTART:20001026T020000
      RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
      TZNAME:EST
```

```
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
END:STANDARD
END:VTIMEZONE
BEGIN:VEVENT
ATTENDEE;PARTSTAT=ACCEPTED;ROLE=CHAIR:mailto:cyrus@example.com
ATTENDEE;PARTSTAT=NEEDS-ACTION:mailto:lisa@example.com
DTSTAMP:20060206T001220Z
DTSTART;TZID=US/Eastern:20060104T100000
DURATION:PT1H
LAST-MODIFIED:20060206T001330Z
ORGANIZER:mailto:cyrus@example.com
SEQUENCE:1
STATUS:TENTATIVE
SUMMARY:Event #3
UID:DC6C50A017428C5216A2F1CD@example.com
END:VEVENT
END:VCALENDAR
</C:calendar-data>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>

  <D:response>
    <D:href>http://cal.example.com/bernard/work/abcd4.ics</D:href>
    <D:propstat>
      <D:prop>
        <D:getetag>"fffff-abcd4"</D:getetag>
        <C:calendar-data>BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VTODO
DTSTAMP:20060205T235335Z
DUE;VALUE=DATE:20060104
STATUS:NEEDS-ACTION
SUMMARY:Task #1
UID:DDDEEB7915FA61233B861457@example.com
BEGIN:VALARM
ACTION:AUDIO
TRIGGER;RELATED=START:-PT10M
END:VALARM
END:VTODO
END:VCALENDAR
</C:calendar-data>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
```

```
        </D:response>

        <D:response>
          <D:href>http://cal.example.com/bernard/work/abcd5.ics</D:href>
          <D:propstat>
            <D:prop>
              <D:getetag>"fffff-abcd5"</D:getetag>
              <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VTODO
      DTSTAMP:20060205T235300Z
      DUE;VALUE=DATE:20060106
      LAST-MODIFIED:20060205T235308Z
      SEQUENCE:1
      STATUS:NEEDS-ACTION
      SUMMARY:Task #2
      UID:E10BA47467C5C69BB74E8720@example.com
      BEGIN:VALARM
      ACTION:AUDIO
      TRIGGER;RELATED=START:-PT10M
      END:VALARM
      END:VTODO
      END:VCALENDAR
      </C:calendar-data>
            </D:prop>
            <D:status>HTTP/1.1 200 OK</D:status>
          </D:propstat>
        </D:response>

        <D:response>
          <D:href>http://cal.example.com/bernard/work/abcd6.ics</D:href>
          <D:propstat>
            <D:prop>
              <D:getetag>"fffff-abcd6"</D:getetag>
              <C:calendar-data>BEGIN:VCALENDAR
      VERSION:2.0
      PRODID:-//Example Corp.//CalDAV Client//EN
      BEGIN:VFREEBUSY
      ORGANIZER;CN="Bernard Desruisseaux":mailto:bernard@example.com
      UID:76ef34-54a3d2@example.com
      DTSTAMP:20050530T123421Z
      DTSTART:20060101T000000Z
      DTEND:20060108T000000Z
      FREEBUSY:20050531T230000Z/20050601T010000Z
      FREEBUSY;FBTYPE=BUSY-TENTATIVE:20060102T100000Z/20060102T120000Z
      FREEBUSY:20060103T100000Z/20060103T120000Z
      FREEBUSY:20060104T100000Z/20060104T120000Z
```

```
      FREEBUSY;FBTYPE=BUSY-UNAVAILABLE:20060105T100000Z/20060105T120000Z
      FREEBUSY:20060106T100000Z/20060106T120000Z
      END:VFREEBUSY
      END:VCALENDAR
      </C:calendar-data>
            </D:prop>
            <D:status>HTTP/1.1 200 OK</D:status>
          </D:propstat>
        </D:response>

      </D:multistatus>
```

## Appendix C. Changes

### C.1. Changes in -10

a. Added new section about support for X- items when storing data.

b. Added new precondition to allow servers to reject queries on unsupported X- items, and a new example.

c. Added new text about always supporting X- in calendar-data.

d. Created new section for PUT, COPY and MOVE preconditions.

e. Report examples re-done with full listing of calendar data in Appendix.

f. Removed description of using UID, SUMMARY etc as resource name.

g. Indicate that calendar object resource may contain only overridden components.

h. Add security consideration about not expose details in resource names.

i. Add constraint that free-busy-query can only be run on a collection.

j. Add preconditions for calendar-timezone property/elements in MKCALENDAR, PROPPATCH and calendar-query REPORT.

k. Fix principal-match example.

## C.2. Changes in -09

a. Numerous editorial changes.

b. Removed the CALDAV:is-defined XML element.

c. Removed section on privilege aggregation.

d. Renamed the CALDAV:expand-recurrence-set XML element to CALDAV: expand and clarified the server behavior.

e. Renamed the CALDAV:calendar-component-restriction-set XML element to CALDAV:supported-calendar-component-set.

f. Renamed the CALDAV:calendar-restrictions XML element to CALDAV: supported-calendar-data.

g. Renamed some preconditions as "success conditions" instead of "failure causes". For instance, the precondition CALDAV: calendar-collection-location-bad has been renamed to CALDAV: calendar-collection-location-ok.

h. Reordered some sections.

i. Clarified the definition of CALDAV:time-range to specify that a repeating VALARM component is said to intersect a given time range if at least one of its trigger intersect the time range.

j. Clarified that calendar object resources stored in calendar collections MUST NOT specify the iCalendar METHOD property.

k. Clarified that CALDAV:calendar-data XML element is not a WebDAV property even though it is specified in the DAV:prop XML element in both calendaring REPORT requests and responses.

l. Clarified CALDAV:limit-recurrence-set with respect to the RANGE parameter on the RECURRENCE-ID property.

m. Changed the CALDAV:free-busy-query XML element to contain exactly one CALDAV:time-range XML element.

n. Changed many ELEMENT and ATTLIST declarations to comply with DTD syntax.

o. Changed XML element CALDAV:calendar-query to allow new XML element CALDAV:timezone.

p. Changed the XML elements CALDAV:time-range, CALDAV:expand and CALDAV:limit-recurrence-set to only allow DATE-TIME with UTC time values for the "start" and "end" attributes.

q. Changed description of CALDAV:limit-recurrence-set to specify that re-scheduled "overridden" recurrence instances whose original scheduled time used to overlap the time range specified by the "start" and "end" attribute should always be returned in a REPORT response.

r. Changed the description of the value of CALDAV:calendar-data XML element to specify that the CR character (US-ASCII decimal 13) MAY be omitted in the iCalendar object specified in this XML element.

s. Added specific requirements for entity tags support.

t. Added more preconditions.

u. Added further guidelines about finding calendars.

v. Added XML element CALDAV:limit-freebusy-set to limit the set of FREEBUSY property values returned in VFREEBUSY components.

w. Added property CALDAV:calendar-timezone on calendar collections.

x. Added XML element CALDAV:timezone to override the CALDAV: calendar-timezone property for a given CALDAV:calendar-query REPORT request.

y. Added text on the conversion of "floating date" and "floating time" values to date with UTC time values.

z. Completed internationalization considerations section.

aa. Completed security considerations section.

### C.3. Changes in -08

a. Removed statement that said that client SHOULD always request DAV:getetag in calendar REPORTs.

b. Removed redefiniton of DAV:response.

c. Removed XML elements CALDAV:calendar-data-only.

d. Removed resource type CALDAV:calendar-home.

e. Moved the CALDAV:calendar-data element in the DAV:prop element in requests, and in the DAV:propstat element in responses.

f. Further defined the request body of MKCALENDAR to allow clients to set properties at calendar collection creation time.

g. Renamed CALDAV:calendar-home-URL to CALDAV:calendar-home-set

h. Clarified the fact that calendar collections may only contain calendar object resources and ordinary collections.

i. Clarified that calendar REPORTs should only be applied to calendar object resources contained in calendar collections.

j. Changed the CALDAV:calendar-component-restriction-set and CALDAV: calendar-restriction properties to always be protected.

k. Changed to use existing postcondition DAV:needs-privileges instead of a new CALDAV:insufficient-privilege postcondition.

l. Added example for limit-recurrence-set.

m. Added example for expand-recurrence-set.

n. Moved CALDAV:calendar-address-set in the calendar-schedule draft and renamed it to CALDAV:calendar-user-address-set.

o. Added guidelines on attachments and alarms.

### C.4. Changes in -07

a. Various editorial changes.

b. Added properties calendar-restrictions and calendar-component-restriction-set on calendar collections.

c. Added properties calendar-home-URL and calendar-address-set on principal resources.

d. Removed property calendar-URL on principal resources.

e. Added pre- and postconditions to reports.

f. Added new XML elements calendar-data-only and limit-recurrent-set.

g. Modified calendar-data XML element to support the attributes content-type and version.

h. Reorganised sections 3, 4, 5 & 6 into two sections and re-ordered sub-sections.

i. Added comment about client not setting a duplicate displayname.

j. Removed three CalDAV OPTIONS requests.

k. Changed "authenticated user" to "user" in various places.

l. Rewrote section on calendar object resource restrictions for better clarity.

### C.5. Changes in -06

a. Reworded section "Recurrence and the Data Model".

b. Removed timezone collection feature.

c. Removed ability for a server to return the Location header on a successful PUT request.

d. Clarified restrictions on calendar object resources contained in calendar collections.

e. Added preconditions on PUT in calendar collections.

f. Added informative "Guidelines" section, with information on locking and how to find calendar collections.

g. Moved "Sychronization Operations" section in the "Guidelines" section.

### C.6. Changes in -05

a. Removed a lot of non-normative text.

b. Removed property promotion/demotion requirements.

c. Removed calendar-owner and cal-scale properties.

d. Removed 'ical' prefix/text from element names.

e. Relaxed WebDAV Class 2 (locking) requirement to a MAY.

f. Relaxed MKCALENDAR requirement to a SHOULD.

g. Moved the XML Namespace section in the Introduction.

h. Added CALDAV: prefix to CalDAV XML elements in the text.

i. Added CALDAV:calendar-multiget report.

j. Added CALDAV:free-busy-query report.

k. Added CALDAV:calendar-description property.

l. Changed CALDAV:calendar-query-result element name to CALDAV: calendar-data

m. Added description and examples of handling timezones.

n. Added mandatory "start" and "end" attributes to the CALDAV: expand-recurrence-set element.

o. Added three CalDAV OPTIONS requests.

p. Grouped XML Element declarations in a separate section.

### C.7. Changes in -04

a. Added a note about the HTTP Location response header.

b. Added report calendar-query.

c. Removed reports calendar-property-search and calendar-time-range.

d. Removed section on CalDAV and timezones.

e. Added requirement to return ETag on creation.

f. Revised data model to remove sub-collections from calendar collection.

g. Added informative references section.

h. Removed dependencies on DASL.

### C.8. Changes in -03

a. Removed Calendar Containers (simplification that doesn't seem to remove much functionality)

b. Added MKCALENDAR to create calendars and all sub-collections

c. Added cal-scale property to calendars

### C.9. Changes in -02

Basically still adding major sections of content:

a. Defined new field values to the OPTIONS "DAV:" response header

b. Added new resource properties

c. Added new principal properties

d. Added new SCHEDULE method and related headers

e. Added new privileges for scheduling

### C.10. Changes in -01

a. Added section on privileges for calendaring, extending WebDAV ACL privilege set

b. Defined what to do with unrecognized properties in the bodies of iCalendar events, with respect to property promotion/demotion

## Authors' Addresses

Cyrus Daboo

Email: cyrus@daboo.name

Bernard Desruisseaux
Oracle Corporation
600 Blvd. de Maisonneuve West
Suite 1900
Montreal, QC  H3A 3J2
CA

Email: bernard.desruisseaux@oracle.com
URI:   http://www.oracle.com/

Lisa Dusseault
Open Source Application Foundation
2064 Edgewood Dr.
Palo Alto, CA  94303
US

Email: lisa@osafoundation.org
URI:   http://www.osafoundation.org/

## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

## Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Copyright Statement

## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.