



CalWSRest - Restful Web service protocol for calendaring Version 0.1

Draft 01

5 August 2010

Specification URIs:

This Version:

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].html](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].html)
[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].odt](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].odt)
[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].pdf](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].pdf)

Previous Version:

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].html](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].html)
[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].odt](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].odt)
[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].pdf](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].pdf)

Latest Version:

[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].html](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].html)
[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].odt](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].odt)
[http://docs.oasis-open.org/\[tc-short-name\]/\[additional path/filename\].pdf](http://docs.oasis-open.org/[tc-short-name]/[additional path/filename].pdf)

Technical Committee:

CalConnect TC-XML

Chair(s):

[Chair name]

Editor(s):

Michael A Douglass

Related Work:

This specification is related to:

- <https://datatracker.ietf.org/idtracker/draft-daboo-et-al-icalendar-in-xml>

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/wscal/calws>

Abstract:

This document describes a RESTful web service for calendar access and update.

Status:

This document was last revised or approved by the [TC name | membership of OASIS] on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at [http://www.oasis-open.org/committees/\[specific location\]/](http://www.oasis-open.org/committees/[specific location]/).

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page ([http://www.oasis-open.org/committees/\[specific location\]/ipr.php](http://www.oasis-open.org/committees/[specific location]/ipr.php)).

The non-normative errata page for this specification is located at [http://www.oasis-open.org/committees/\[specific location\]/](http://www.oasis-open.org/committees/[specific location]/).

Notices

Copyright © OASIS® 2008. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication 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 OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", [insert specific trademarked names, abbreviations, etc. here] are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1 Introduction.....	6
1.1 Terminology.....	6
1.2 Normative References.....	6
1.3 Non-normative References.....	6
2 Issues not addressed by this specification.....	8
2.1 Access Control.....	8
2.2 Provisioning.....	8
2.3 Copy/Move.....	8
2.4 Creating Collections.....	8
2.5 Retrieving collections.....	8
3 CalWS Glossary.....	9
3.1 Hrefs.....	9
3.2 Calendar Object Resource.....	9
3.3 Calendar Collection.....	9
3.4 Scheduling Calendar Collection.....	9
3.5 Principal Home.....	9
4 Overview of the CalWS protocol.....	10
4.1 HTTP Methods.....	10
4.2 Properties.....	10
4.3 Operations.....	10
4.4 Calendar Object Resources.....	10
4.5 Timezone information.....	10
5 Error conditions.....	11
6 Properties and link relations.....	12
6.1 Property URIs.....	12
6.2 CalWS/timezone-service relation.....	12
6.3 CalWS/principal-home relation.....	12
6.4 CalWS/current-principal-freebusy relation.....	12
6.5 CalWS/principal-freebusy relation.....	12
6.6 CalWS/child-collection relation.....	12
7 Retrieving Collection and Service Properties.....	13
7.1 Request parameters.....	13
7.2 Responses:.....	13
7.3 Example - retrieving server properties:.....	13
8 Creating Calendar Object Resources.....	15
8.1 Request parameters.....	15
8.2 Responses:.....	15
8.3 Preconditions for Calendar Object Creation.....	15
8.4 Example - successful POST:.....	16

8.5 Example - unsuccessful POST:.....	16
9 Retrieving resources.....	17
9.1 Request parameters.....	17
9.2 Responses:.....	17
10 Updating resources.....	18
10.1 Responses:.....	18
10.2 Example - successful update:.....	18
10.3 Example - unsuccessful update:.....	18
11 Deletion of resources.....	19
11.1 Delete for Collections.....	19
11.2 Responses:.....	19
12 Querying calendar resources.....	20
12.1 Limiting data returned.....	20
12.2 Example: time range limited retrieval.....	20
13 Free-busy queries.....	24
13.1 URL Query Parameters.....	24
13.2 URL parameters - notes.....	25
13.3 HTTP Operations.....	26
13.4 Response Codes.....	26
13.5 Examples.....	26
# Conformance.....	29

1 Introduction

The CalWS protocol is built upon and makes the same assumptions about structure as the CalDAV protocol defined in [RFC 4791] and related specifications. It does NOT require nor assume the WebDAV nor CalDAV protocol but does make use of some of the same elements and structures in the CalDAV XML namespace..

Calendar resources, for example events and tasks are stored as named resources (files) inside special collections (folders) known as "**Calendar Collections**".

This specification can be looked upon as a layer built on top of CalDAV and defines the basic operations which allow creation, retrieval, update and deletion. In addition, query operations are defined to allow efficient, partial retrieval of calendar data.

This does not mean that a CalWS service must be built on CalDAV, merely that a degree of conformity is established such that services built in that manner do not have a significant mismatch. It is assumed that some CalWS services will be built without any CalDAV support.

1.1 Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 [RFC 2119].

1.2 Normative References

- [RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- [RFC 2616] Fielding, et al, *Hypertext Transfer Protocol -- HTTP/1.1* <http://tools.ietf.org/html/rfc2616>
- [RFC 4791] Daboo, et al. *Calendar Extensions to WebDAV (CalDAV)*. IETF RFC 2119, March 2007. <http://www.ietf.org/rfc/rfc4791.txt>.
- [RFC 5545] B. Desruisseaux, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* <http://tools.ietf.org/html/rfc5545>
- [RFC 5546] C. Daboo. *iCalendar Transport-Independent Interoperability Protocol (iTIP)* <http://tools.ietf.org/html/rfc5546>
- [draft-xcal] C. Daboo, M. Douglass, S. Lees *xCal: The XML format for iCalendar* <https://datatracker.ietf.org/idtracker/draft-daboo-et-al-icalendar-in-xml>
- [FreeBusy Read URL] E York. *Freebusy read URL* <http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf>

1.3 Non-normative References

- [Reference] [reference citation]
- [Reference] [reference citation]

NOTE: The proper format for a citation to an OASIS Technical Committee's work (whether Normative or Non-Normative) is:

OASIS

Stage (Committee Draft 01, Committee Draft 02, Committee Specification 01, etc. or Standard)

Title (*italicized* or in quotation marks)

Approval Date (Month YYYY)

URI of the actual Authoritative Specification (namespace is not acceptable as the content changes over time)

For example:

[EDXL-HAVE] OASIS Standard, "Emergency Data Exchange Language (EDXL) Hospital AVailability Exchange (HAVE) Version 1.0", November 2008.
http://docs.oasis-open.org/emergency/edxl-have/os/emergency_edxl_have-1.0-spec-os.doc

2 Issues not addressed by this specification.

A number of issues are not addressed by this specification, either because they should be addressed elsewhere or will be addressed at some later date.

2.1 Access Control

It is assumed that the targeted server will set an appropriate level of access based on authentication. This specification will not attempt to address the issues of sharing or ACLs.

2.2 Provisioning

The protocol will not provide any explicit provisioning operations. If it is possible to authenticate or address a principals calendar resources then they will be automatically created if necessary or appropriate

2.3 Copy/Move

These operations are not yet defined for this version of the CalWS protocol. Both operations raise a number of issues. In particular implementing a move operation through a series of retrievals, insertions and deletions may cause undesirable side-effects. Both these operations will be defined in a later version of this specification.

2.4 Creating Collections

We will not address the issue of creating collections within the address space. The initial set is created by provisioning.

2.5 Retrieving collections

This operation is currently undefined. A GET on a collection may fail or return a complete calendar object representing the collection.

3 CalWS Glossary

3.1 Hrefs

An href is a URI reference to a resource, for example

```
"http://example.org/user/fred/calendar/event1.ics".
```

The URL above reflects a possible structure for a calendar server.

3.2 Calendar Object Resource

A calendar object resource is an event, meeting or a task. Attachments are resources but NOT calendar object resources. An event or task with overrides is a single calendar resource entity.

3.3 Calendar Collection

A folder only allowed to contain calendar object resources.

3.4 Scheduling Calendar Collection

A folder only allowed to contain calendar resources which is also used for scheduling operations. Scheduling events placed in such a collection will trigger implicit scheduling activity on the server.

3.5 Principal Home

The collection under which all the resources for a given principal are stored. For example, for principal "fred" the principal home might be "/user/fred/"

4 Overview of the CalWS protocol

The protocol is an HTTP based RESTful protocol using a limited set of methods. Each request may be followed by a response containing status information.

4.1 HTTP Methods

The following methods are specified in the protocol description, PUT, POST, GET, DELETE. To avoid various issues with certain methods being blocked clients may use the X-HTTP-Method-Override: header to specify the intended operation. Servers should behave as if the named method was used.

```
POST /user/fred/calendar/ HTTP/1.1
...
X-HTTP-Method-Override: PUT
```

4.2 Properties

A service or resource will have a number of properties which describe the current state of that service or resource. These properties are accessed through a GET on the target resource or service with an ACCEPT header specifying application/xrd+xml. See Retrieving Collection and Service Properties below

4.3 Operations

The following operations are defined by this specification:

- Retrieval and update of service and resource properties
- Creation of a calendar object
- Retrieval of a calendar object
- Update of a calendar object
- Deletion of a calendar object
- Query
- Free-busy query

4.4 Calendar Object Resources

The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC 4791] section 4.2. An additional constraint for CalWS is that no timezone specifications are transferred.

4.5 Timezone information

It is assumed that the client and server each have access to a full set of up to date timezone information. Timezones will be referenced by a timezone identifier from the full set of Olson data together with a set of well-known aliases defined [where?]. CalWS services may advertise themselves as timezone servers through the server properties object.

5 Error conditions

Each operation on the calendar system has a number of pre-conditions and post-conditions that apply.

A "precondition" for 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. Any violation of these conditions will result in an error response in the form of a CalWS XML error element containing the violated condition and an optional description. For example

```
<?xml version="1.0" encoding="utf-8"
  xmlns:D="DAV:"
  xmlns:C="urn:ietf:params:xml:ns:caldav" ?>
<D:error>
  <C:supported-filter>
    <C:prop-filter name="X-ABC-GUID"/>
  </C:supported-filter>
  <D:description>Unknown property </C:description>
</D:error>
```

Each method specification defines the preconditions that must be satisfied before the method can succeed. A number of postconditions are generally specified which define the state that must exist after the execution of the operation. Preconditions and postconditions are defined as error elements in the CalWS XML namespace.

6 Properties and link relations

6.1 Property URIs

In the XRD entity returned properties and related services and entities are defined by absolute URIs. Certain property URIs correspond to CalDAV preconditions. Each URL is prefixed by the namespace **http://calws.example.net**. Those properties which correspond to CalDAV properties have the additional path element "**caldav**", for example

```
http://calws.example.net/caldav/supported-calendar-data
```

corresponds to

```
CalDAV:supported-calendar-data
```

In addition to those CalDAV properties, the CalWS specification defines a number of other properties and link relations with the URI prefix of **http://calws.example.net/calws** (**CalWS** in the description below)

6.2 CalWS/timezone-service relation.

The location of a timezone service used to retrieve timezone information and specifications. This may be an absolute URL referencing some other service or a relative URL if the current server also provides a timezone service.

6.3 CalWS/principal-home relation.

Provides the URL to the user home for the currently authenticated principal.

6.4 CalWS/current-principal-freebusy relation.

Provides the URL to use as a target for freebusy requests for the current authenticated principal.

6.5 CalWS/principal-freebusy relation.

Provides the URL to use as a target for freebusy requests for a different principal.

6.6 CalWS/child-collection relation.

Provides the URL to the user home for the currently authenticated principal. The link element contains a type element and a number of properties.

type: A comma-separated list of collection types from

collection: a plain collection

calendar: a calendar collection

scheduling: a scheduling calendar collection

inbox: an inbox

Link properties:

http://calws.example.net/calws/created: Create date for the collection

7 Retrieving Collection and Service Properties

Given the URL of a CalWS service a client retrieves the service XRD document through a GET on the service URL with an ACCEPT header specifying application/xrd+xml.

To retrieve the collection properties is identical to obtaining service properties, that is, execute a GET on the target URL with an ACCEPT header specifying application/xrd+xml.

All of the properties defined in CalDAV [RFC 4791] section 5.2 are defined for a collection or service. The service properties define the global limits and defaults. Any defined properties defined on collections within the service hierarchy override those service defaults. The service may choose to prevent such overriding of defaults and limits when appropriate.

7.1 Request parameters

- None

7.2 Responses:

- 200: OK
- 403: Forbidden
- 404: Not found

7.3 Example - retrieving server properties:

```
>>Request
GET / HTTP/1.1
Host: example.com
ACCEPT:application/xrd+xml

>>Response
<XRD xmlns="http://docs.oasis-open.org/ns/xri/xrd-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Expires>1970-01-01T00:00:00Z</Expires>
  <Subject>http://example.com/calws</Subject>

  <Link rel="http://calws.example.net/calws/timezone-service"
    href="http://example.com/tz" />

  <Link rel="http://calws.example.net/calws/principal-home"
    type="collection"
    href="http://example.com/calws/user/fred">
    <Title xml:lang="en">Fred's calendar home</Title>
    <Property type="http://calws.example.net/calws/created"
      >1970-01-01</Property>
  </Link>

  <Link rel="http://calws.example.net/calws/child-collection"
    type="calendar,scheduling"
    href="http://example.com/calws/user/fred/calendar">
    <Title xml:lang="en">Calendar</Title>
    <Property type="http://calws.example.net/created/1.0"
      >1970-01-01</Property>
  </Link>

  <Link rel="http://calws.example.net/calws/child-collection"
    type="calendar-inbox"
    href="http://example.com/calws/user/fred/inbox">
    <Title xml:lang="en">Inbox</Title>
```

```
<Property type="http://calws.example.net/created/1.0"
  >1970-01-01</Property>
</Link>

<Property type="http://calws.example.net/max-instances/1.0"
  >1000</Property>

<Property type="http://calws.example.net/max-attendees-per-instance/1.0"
  >100</Property>
...
</XRD>
```

8 Creating Calendar Object Resources

Creating calendar object resources is carried out by a POST on the parent collection. The body of the request will contain the resource being created. The request parameter "action=create" indicates this POST is a create. The location header of the response gives the URL of the newly created object.

8.1 Request parameters

- action=create

8.2 Responses:

- 201: created
- 403: Forbidden - no access

8.3 Preconditions for Calendar Object Creation

- **CalWS:target-exists:** The target of a PUT must exist. Use POST to create entities and PUT to update them.
- **CalWS:not-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;
- **CalWS:invalid-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);
- **CalWS:invalid-calendar-object-resource:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST obey all restrictions specified in Calendar Object Resources (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.);
- **CalWS:unsupported-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;
- **CalWS: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 CalWS:href element
<!ELEMENT uid-conflict (CalWS:href)>
- **CalWS:invalid-calendar-collection-location:** In a COPY or MOVE request, when the Request-URI is a calendar collection, the Destination-URI MUST identify a location where a calendar collection can be created;
- **CalWS:exceeds-max-resource-size:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have an octet size less than or equal to the value of the CalDAV:max-resource-size property value on the calendar collection where the resource will be stored;
- **CalWS:before-min-date-time:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have all of its iCalendar DATE or DATE-TIME property values (for each recurring instance) greater than or equal to the value of the CalDAV:min-date-time property value on the calendar collection where the resource will be stored;

- **CalWS:after-max-date-time:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have all of its iCalendar DATE or DATE-TIME property values (for each recurring instance) less than the value of the CalDAV:max-date-time property value on the calendar collection where the resource will be stored;
- **CalWS:too-many-instances:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST generate a number of recurring instances less than or equal to the value of the CalDAV: max-instances property value on the calendar collection where the resource will be stored;
- **CalWS:too-many-attendees-per-instance:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have a number of ATTENDEE properties on any one instance less than or equal to the value of the CalDAV:max-attendees-per-instance property value on the calendar collection where the resource will be stored;

8.4 Example - successful POST:

```
>>Request

POST /user/fred/calendar/?action=create HTTP/1.1
Host: example.com
Content-Type: application/xml+calendar; charset="utf-8"
Content-Length: ?

<?xml version="1.0" encoding="utf-8" ?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    ...
  </vcalendar>
</icalendar>

>>Response

HTTP/1.1 201 Created
Location: http://example.com/user/fred/calendar/event1.ics
```

8.5 Example - unsuccessful POST:

```
>>Request

POST /user/fred/readcalendar/?action=create HTTP/1.1
Host: example.com
Content-Type: text/text; charset="utf-8"
Content-Length: ?

This is not an xml calendar object

>>Response

HTTP/1.1 403 Forbidden
<?xml version="1.0" encoding="utf-8"
  xmlns:D="DAV:"
  xmlns:C="urn:ietf:params:xml:ns:caldav" ?>
<D:error>
  <C:supported-calendar-data/>
  <D:description>Not an icalendar object</C:description>
</D:error>
```

9 Retrieving resources

A simple GET on the href will return a named resource. If that resource is a recurring event or task with overrides, the entire set will be returned.

9.1 Request parameters

- format=xml - return as an XML object - the default

9.2 Responses:

- 200: OK
- 403: Forbidden - no access

10 Updating resources

Resources are updated with the PUT method targeted at the resource href. The body of the request contains a complete new resource which effectively replaces the targeted resource. To allow for optimistic locking of the resource we use the if-match header.

Preconditions as specified in Preconditions for Calendar Object Creation are applicable.

10.1 Responses:

- 200: OK
- 304: Not modified - entity was modified by some other request
- 403: Forbidden - no access, does not exist etc. See error response

10.2 Example - successful update:

```
>>Request

PUT /user/fred/calendar/event1.ics HTTP/1.1
Host: example.com
Content-Type: application/xml+calendar; charset="utf-8"
Content-Length: ?

<?xml version="1.0" encoding="utf-8" ?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    ...
  </vcalendar>
</icalendar>

>>Response

HTTP/1.1 200 OK
```

10.3 Example - unsuccessful update:

```
>>Request

PUT /user/fred/readcalendar/event1.ics HTTP/1.1
Host: example.com
Content-Type: text/text; charset="utf-8"
Content-Length: ?

<?xml version="1.0" encoding="utf-8" ?>
<icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
  <vcalendar>
    ...
  </vcalendar>
</icalendar>

>>Response

HTTP/1.1 403 Forbidden
<?xml version="1.0" encoding="utf-8"
  xmlns:D="DAV:"
  xmlns:CW="http://docs.oasis-open.org/ns/wscal/calws" ?>
<D:error>
  <CW:target-exists/>
  <D:description>Target of update must exist</C:description>
</D:error>
```

11 Deletion of resources

Delete is defined in [RFC 2616] Section 9.7. In addition to conditions defined in that specification, servers must remove any references from the deleted resource to other resources. Resources are deleted with the DELETE method targeted at the resource URL. After a successful completion of a deletion a GET on that URL must result in a 404 - Not Found status.

11.1 Delete for Collections

Delete for collections may or may not be supported by the server. Certain collections are considered undeletable. On a successful deletion of a collection all contained resources to any depth must also be deleted.

11.2 Responses:

- 200: OK
- 403: Forbidden - no access
- 404: Not Found

12 Querying calendar resources

Querying provides a mechanism by which information can be obtained from the service through possibly complex queries. A list of icalendar properties can be specified to limit the amount of information returned to the client. A query takes the parts

- Limitations on the data returned
- Selection of the data
- Optional timezone id for floating time calculations.

The current specification uses CalDAV multiget and calendar-query XML bodies as specified in [RFC 4791] with certain limitations and differences.

1. The POST method is used for all requests, the action being identified by the outer element.
2. While CalDAV servers generally only support [RFC 5545] and assume that as the default, the delivery format for CalWS will, by default, be [draft-xcal].
3. The CalDAV query allows the specification of a number of DAV properties. Specification of these properties, with the exception of DAV:getetag, is considered an error in CalWS.
4. The CalDAV:proppnames element is invalid

With those differences, the CalDAV specification is the normative reference for this operation.

12.1 Limiting data returned

This is achieved by specifying one of the following

- CalDAV:allprop return all properties (some properties are specified as not being part of the allprop set so are not returned)
- CalDAV:prop An element which contains a list of properties to be returned . May only contain DAV:getetag and CalDAV:calendar-data

Of particular interest, and complexity, is the calendar-data property which can contain a time range to limit the range of recurrences returned and/or a list of calendar properties to return.

12.2 Example: time range limited retrieval

This example shows the time-range limited retrieval from a calendar which results in 2 events, one a recurring event and one a simple non-recurring event.

```
>> Request <<

POST /user/fred/calendar/ HTTP/1.1
Host: calws.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 content-type="application/xml+calendar" >
      <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>
</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: Sat, 11 Nov 2006 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 content-type="application/xml+calendar" >
          <xc:icalendar
            xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
            <xc:vcalendar>
              <xc:properties>
                <xc:calscale><text>GREGORIAN</text></xc:calscale>
                <xc:prodid>
                  <xc:text>-//Example Inc.//Example Calendar//EN</xc:text>
                </xc:prodid>
                <xc:version><xc:text>2.0</xc:text></xc:version>
              </xc:properties>
              <xc:components>
                <xc:vevent>
                  <xc:properties>
                    <xc:dtstart>
                      <xc:parameters>
                        <xc:tzid>US/Eastern<xc:tzid>
                      <xc:parameters>
                        <xc:date-time>20060102T120000</xc:date-time>
                      </xc:dtstart>
                    <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
                    <xc:summary>
                      <xc:text>Event #2</xc:text>
                    </xc:summary>
                    <xc:uid>
                      <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
                    </xc:uid>
                    <xc:rrule>
                      <xc:recur>
                        <xc:freq>DAILY</xc:freq>

```

```

        <xc:count>5</xc:count>
    </xc:recur>
</xc:rrule>
</xc:properties>
</xc:vevent>

<xc:vevent>
<xc:properties>
  <xc:dtstart>
    <xc:parameters>
      <xc:tzid>US/Eastern<xc:tzid>
    <xc:parameters>
      <xc:date-time>20060104T140000</xc:date-time>
    </xc:dtstart>
  <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
  <xc:summary>
    <xc:text>Event #2 bis</xc:text>
  </xc:summary>
  <xc:uid>
    <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
  </xc:uid>
  <xc:recurrence-id>
    <xc:parameters>
      <xc:tzid>US/Eastern<xc:tzid>
    <xc:parameters>
      <xc:date-time>20060104T120000</xc:date-time>
    </xc:recurrence-id>
  <xc:rrule>
    <xc:recur>
      <xc:freq>DAILY</xc:freq>
      <xc:count>5</xc:count>
    </xc:recur>
  </xc:rrule>
</xc:properties>
</xc:vevent>

<xc:vevent>
<xc:properties>
  <xc:dtstart>
    <xc:parameters>
      <xc:tzid>US/Eastern<xc:tzid>
    <xc:parameters>
      <xc:date-time>20060106T140000</xc:date-time>
    </xc:dtstart>
  <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
  <xc:summary>
    <xc:text>Event #2 bis bis</xc:text>
  </xc:summary>
  <xc:uid>
    <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
  </xc:uid>
  <xc:recurrence-id>
    <xc:parameters>
      <xc:tzid>US/Eastern<xc:tzid>
    <xc:parameters>
      <xc:date-time>20060106T120000</xc:date-time>
    </xc:recurrence-id>
  <xc:rrule>
    <xc:recur>
      <xc:freq>DAILY</xc:freq>
      <xc:count>5</xc:count>
    </xc:recur>
  </xc:rrule>
</xc:properties>
</xc:vevent>
</xc:components>
</xc:vcalendar>
</xc:icalendar>
  </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 content-type="application/xml+calendar" >
        <xcal:icalendar
          xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
<xc:vcalendar>
  <xc:properties>
    <xc:calscale><text>GREGORIAN</text></xc:calscale>
    <xc:prodid>
      <xc:text>-//Example Inc.//Example Calendar//EN</xc:text>
    </xc:prodid>
    <xc:version><xc:text>2.0</xc:text></xc:version>
  </xc:properties>
  <xc:components>
    <xc:vevent>
      <xc:properties>
        <xc:dtstart>
          <xc:parameters>
            <xc:tzid>US/Eastern<xc:tzid>
          <xc:parameters>
            <xc:date-time>20060104T100000</xc:date-time>
          </xc:dtstart>
          <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
          <xc:summary>
            <xc:text>Event #3</xc:text>
          </xc:summary>
          <xc:uid>
            <xc:text>DC6C50A017428C5216A2F1CD@example.com</xc:text>
          </xc:uid>
          <xc:rrule>
            <xc:recur>
              <xc:freq>DAILY</xc:freq>
              <xc:count>5</xc:count>
            </xc:recur>
          </xc:rrule>
        </xc:properties>
      </xc:vevent>
    </xc:components>
  </xc:vcalendar>
</xc:icalendar>
  </C:calendar-data>
</D:prop>
  <D:status>HTTP/1.1 200 OK</D:status>
</D:propstat>
</D:response>
</D:multistatus>

```

13 Free-busy queries

Freebusy queries are used to obtain freebusy information for a calendar-collection or principals. The result contains information only for events to which the current principal has sufficient access.

When targeted at a calendar collection the result is based only on the calendaring entities contained in that collection. When targeted at a principal freebusy URL the result will be based on all information which affect the principals freebusy status, for example availability.

The possible targets are:

- A calendar collection URL
- The XRD link with relation CalWS/current-principal-freebusy
- The XRD link with relation CalWS/principal-freebusy with a principal given in the request.

The query follows the specification defined in [FreeBusy Read URL] with certain limitations. As an authenticated user to the CalWS service scheduling read-freebusy privileges must have been granted. As an unauthenticated user equivalent access must have been granted to unauthenticated access.

Freebusy information is returned by default as xcalendar vfreebusy components, as defined by [draft-xcal]. Such a component is not meant to conform to the requirements of VFREEBUSY components in [RFC 5546]. The VFREEBUSY component SHOULD conform to section "4.6.4 Free/Busy Component" of [RFC 5545]. A client SHOULD ignore the ORGANIZER field.

Since a Freebusy query can only refer to a single user, a client will already know how to match the result component to a user. A server MUST only return a single vfreebusy component.

13.1 URL Query Parameters

None of these parameters are required except for the conditions noted below. Appropriate defaults will be supplied by the server.

start

Default: The default value is left up to the server. It may be the current day, start of the current month, etc.

Description: Specifies the start date for the Freebusy data. The server is free to ignore this value and return data in any time range. The client must check the data for the returned time range.

Format: A profile of an [RFC3339] Date/Time. Fractional time is not supported. The server MUST support the expanded version e.g.

```
2007-01-02T13:00:00-08:00
```

It is up to the server to interpret local date/times.

Example:

```
2007-02-03T15:30:00-0800  
2007-12-01T10:15:00Z
```

Notes: Specifying only a start date/time without specifying an end-date/time or period should be interpreted as in [RFC 5545]. The effective period should cover the remainder of that day.

Date-only values are disallowed as the server cannot determine the correct start of the day. Only UTC or date/time with offset values are permitted.

end

Default: Same as start

Description: Specifies the end date for the Freebusy data. The server is free to ignore this value.

Format: Same as start

Example: Same as start

period

Default: The default value is left up to the server. The recommended value is "P42D".

Description: Specifies the amount of Freebusy data to return. A client cannot specify both a period and an end date. Period is relative to the start parameter.

Format: A duration as defined in section 4.3.6 of [RFC 5545]

Example:

```
P42D
```

format

Default: "application/xml+calendar"

Description: Specifies the output format as a MIME type.

A server MUST support the default "application/xml+calendar" which will return an xml vfreebusy object. Support for other formats is optional.

Format: A MIME type

Example:

```
application/xml+calendar
```

account

Default: none

Description: Specifies the principal when the request is targeted at the XRD CalWS/principal-freebusy. Specification of this parameter is an error otherwise.

Format: Server specific

Example:

```
fred  
/principals/users/jim  
user1@example.com
```

13.2 URL parameters - notes

The server is free to ignore the start, end and period parameters. It is recommended that the server return at least 6 weeks of data from the current day.

A client MUST check the time range in the VFREEBUSY response as a server may return a different time range than the requested range.

13.3 HTTP Operations

The server SHOULD return an Etag response header for a successful GET request targeting a Freebusy read URL. Clients MAY use the Etag response header value to do subsequent "conditional" GET requests that will avoid re-sending the Freebusy data again if it has not changed.

13.4 Response Codes

Below are the typical status codes returned by a GET request targeting a Freebusy URL. Note that other HTTP status codes not listed here might also be returned by a server.

- 200 OK
- 302 Redirect
- 400 Start parameter could not be understood / End parameter could not be understood / Period parameter could not be understood
- 401 Unauthorized
- 403 Forbidden
- 404 The data for the requested principal is not currently available, but may be available later.
- 406 The requested format in the Format parameter is not supported.
- 410 The data for the requested principal is no longer available
- 500 General server error

13.5 Examples

The following are examples of URLs used to retrieve Freebusy data for a user:

```
http://www.example.com/freebusy/user1@example.com?
start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00

http://www.example.com/freebusy/user1@example.com?
start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00&
format=text/calendar

http://www.example.com/calendar/user1@example.com?
start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00&
format=text/json

http://www.example.com/freebusy/user1@example.com

http://www.example.com/freebusy?user=user1@example.com&token=xcsfdgetdh&
start=2008-01-01T00:00:00Z&end=2008-12-31T00:00:00Z

http://www.example.com/freebusy?user=user%201@example.com&
start=2008-01-01T00:00:00Z&end=2008-12-31T00:00:00Z
```

Some Request/Response Examples:

An URL with no query parameters:

```
>> Request <<
GET /freebusy/bernard/ HTTP/1.1
Host: www.example.com

>> Response <<
HTTP/1.1 200 OK
Content-Type: application/xml+calendar; charset="utf-8"
Content-Length: xxxx
<xc:icalendar xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
  <xc:vcalendar>
```

```

<xc:properties>
  <xc:calscale><text>GREGORIAN</text></xc:calscale>
  <xc:prodid>
    <xc:text>-//Example Inc.//Example Calendar//EN</xc:text>
  </xc:prodid>
  <xc:version><xc:text>2.0</xc:text></xc:version>
</xc:properties>
<xc:components>
  <xc:vfreebusy>
    <xc:properties>
      <xc:uid>
        <xc:text>76ef34-54a3d2@example.com</xc:text>
      </xc:uid>
      <xc:dtstart>
        <xc:date-time>20060101T000000Z</xc:date-time>
      </xc:dtstart>
      <xc:dtend>
        <xc:date-time>20060108T000000Z</xc:date-time>
      </xc:dtend>
      <xc:dtstamp>
        <xc:date-time>20050530T123421Z</xc:date-time>
      </xc:dtstamp>
      <xc:freebusy>
        <xc:period>20050531T230000Z/20050601T010000Z</xc:period>
      </xc:freebusy>
      <xc:freebusy>
        <xc:parameters>
          <xc:fbsyntax>BUSYTENTATIVE</xc:fbsyntax>
        </xc:parameters>
        <xc:period>20060102T100000Z/20060102T120000Z</xc:period>
      </xc:freebusy>
      <xc:freebusy>
        <xc:period>20060103T100000Z/20060103T120000Z</xc:period>
      </xc:freebusy>
      <xc:freebusy>
        <xc:period>20060104T100000Z/20060104T120000Z</xc:period>
      </xc:freebusy>
      <xc:freebusy>
        <xc:parameters>
          <xc:fbsyntax>BUSYUNAVAILABLE</xc:fbsyntax>
        </xc:parameters>
        <xc:period>20060105T100000Z/20060105T120000Z</xc:period>
      </xc:freebusy>
      <xc:freebusy>
        <xc:period>20060106T100000Z/20060106T120000Z</xc:period>
      </xc:freebusy>
    </xc:vfreebusy>
  </xc:components>
</xc:vcalendar>
</xc:icalendar>

```

An URL with start and end parameters:

```

>> Request <<
GET /freebusy/user1@example.com?start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00
HTTP/1.1
Host: www.example.com

>> Response <<
HTTP/1.1 200 OK
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx
BEGIN:VCALENDAR
VERSION:2.0
PROPID:-//Example Corp.//CalDAV Client//EN
BEGIN:VFREEBUSY
ORGANIZER:
UID:76ef34-54a3d3@example.com
DTSTAMP:20070905T100000Z
DTSTART:20070901T080000Z

```

```
DTEND:20070931T080000Z
FREEBUSY:20070903T230000Z/20070904T010000Z
FREEBUSY;FBTYPE=BUSYTENTATIVE:
20070906T100000Z/20070906T120000Z
FREEBUSY:20070908T100000Z/20070908T120000Z
FREEBUSY:20070909T100000Z/20070909T120000Z
FREEBUSY;FBTYPE=BUSYUNAVAILABLE:
20070915T100000Z/20070917T120000Z
FREEBUSY:20070920T100000Z/20070922T120000Z
END:VFREEBUSY
END:VCALENDAR
```

An URL for which the server does not have any data for that user:

```
>> Request <<
GET /freebusy/user1@example.com?start=2012-12-
01T00:00:00-08:00&end=2012-12-31T00:00:00-08:00
HTTP/1.1
Host: www.example.com
>> Response <<
HTTP/1.1 404 No data
```

Conformance

The last numbered section in the specification must be the Conformance section. Conformance Statements/Clauses go here.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged

Participants:

- [Participant name, affiliation | Individual member]
- [Participant name, affiliation | Individual member]
- [Participant name, affiliation | Individual member]

Appendix B. Non-Normative Text

Appendix C. Revision History