CalWSSOAP - SOAP Web service protocol for calendaring

Version 1.0

13 September 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].odt

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].odt

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].odt

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-soap

Declared Properties and Relations Namespaces

Properties and extended relation types are prefixed with the URL"

http://docs.oasis-open.org/ns/wscal/calwsrel

Abstract:

This document describes a SOAP 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/committeees/[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.

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

Notices

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 7
1.1 Terminology 7
1.2 Normative References 7
1.3 Non-normative References 8
2 Issues not addressed by this specification 9
2.1 Access Control 9
2.2 Provisioning 9
2.3 Copy/Move 9
2.4 Creating Collections 9
2.5 Retrieving collections 9
2.6 Setting service and resource properties 9
3 CalWS Glossary 10
3.1 Hrefs 10
3.2 Calendar Object Resource 10
3.3 Calendar Collection 10
3.4 Scheduling Calendar Collection 10
3.5 Principal Home 10
3.6 Etoken 10
4 Overview of the CalWS protocol 11
4.1 Discovery 11
4.2 Properties 11
4.3 Operations 11
4.4 Calendar Object Resources 12
4.5 Timezone information 12
4.6 Error conditions 12
5 CalWs-SOAP Messages 13
5.1 Common Elements and types 13
6 Properties and link relations 15
6.1 Property and relation-type URIs 15
6.2 supported-features property 15
6.3 max-attendees-per-instance 15
6.4 max-date-time 15
6.5 max-instances 15
6.6 max-resource-size 15
6.7 min-date-time 16
6.8 description 16
6.9 timezone-service relation 16
6.10 principal-home relation 16
6.11 current-principal-freebusy relation 16

6.12 principal-freebusy relation....16
6.13 child-collection relation....16
6.14 created link property....17
6.15 last-modified property....17
6.16 displayname property....17
6.17 timezone property....17
6.18 owner property....17
6.19 collection link property....18
6.20 calendar-collection link property....18
6.21 CalWS:privilege-set XML element....18
6.22 CalWS:supported-calendar-component-set XML element....18
7 Retrieving Collection and Service Properties....19
7.1 Example - retrieving server properties:....19
8 Creating Calendar Object Resources....21
8.1 Preconditions for Calendar Object Creation....21
8.2 Example - successful addItem:....22
8.3 Example - unsuccessful addItem:....23
9 Retrieving resources....24
9.1 Example - successful fetchItem:....24
9.2 Example - unsuccessful fetchItem:....25
10 Updating resources....26
10.1 Example - successful update:....27
10.2 Example - unsuccessful update:....27
11 Deletion of resources....28
11.1 Example - successful deleteItem:....28
11.2 Example - unsuccessful deleteItem:....28
12 Querying calendar resources....30
12.1 Calendar Query common types....30
12.2 CompFilterType....30
12.3 PropFilterType....31
12.4 ParamFilterType....31
12.5 CalendarQueryType elements....32
12.6 Specifying data to be returned....32
12.7 Pre/postconditions for calendar queries....33
12.8 Example: time range limited retrieval....33
13 Free-busy queries....36
13.1 Element values....36
13.2 Examples....37
14 CalWS XML Elements....39
14.1 description XML Element....39
14.2 error XML Element....39
14.3 CalWS:href XML Element....39

14.4 CalWS:target-exists XML Element 39
14.5 CalWS:not-calendar-data XML Element 39
14.6 CalWS:invalid-calendar-data XML Element 40
14.7 CalWS:invalid-calendar-object-resource XML Element 40
14.8 CalWS:unsupported-calendar-component XML Element 40
14.9 CalWS:uid-conflict XML Element 40
14.10 CalWS:invalid-calendar-collection-location XML Element 40
14.11 CalWS:exceeds-max-resource-size XML Element 40
14.12 CalWS:before-min-date-time XML Element 40
14.13 CalWS:after-max-date-time XML Element 41
14.14 CalWS:too-many-instances XML Element 41
14.15 CalWS:too-many-attendees-per-instance XML Element 41
14.16 CalWS:privilege-set 41
14.17 CalWS:privilege 41
14.18 CalWS:read 41
14.19 CalWS:write 42
Conformance 43

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.

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 and freebusy 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 3339] Klyne g., Newman C., *Date and Time on the Internet: Timestamps* http://tools.ietf.org/html/rfc3339

[RFC 4790] Newman, et al. *Internet Application Protocol Collation Registry.* http://www.ietf.org/rfc/rfc4790.txt.

[RFC 4791] Daboo, et al. *Calendaring Extensions to WebDAV (CalDAV).* http://www.ietf.org/rfc/rfc4791.txt.

[draft caldav-sched] Desruisseaux, et al. *CalDAV Scheduling extensions to WebDAV* http://tools.ietf.org/html/draft-desruisseaux-caldav-sched-08

[RFC 4918] L. Dusseault, *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)* http://tools.ietf.org/html/rfc4918

[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

[draft-timezones] C. Daboo, M. Douglass*: Timzone Service Protocol* http://tools.ietf.org/html/draft-douglass-timezone-service

[FreeBusy Read URL] E York. *Freebusy read URL*
http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf

[SOAP11] Simple Object Access Protocol (SOAP) 1.1, 8 May 2000
http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[Web-Linking] M. Nottingham *Web linking*
http://tools.ietf.org/html/draft-nottingham-http-link-header

[WS-Addr] W3C Recommendation,Web Services Addressing 1.0 - Core, and Web Services Addressing 1.0 - SOAP Binding, 9 May 2006
http://www.w3.org/2002/ws/addr/

[WT-I-Basic] Basic Profile Version 1.1, 10 April 2006
http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html

[WS-I-Bind] Web Services-Interoperability Organization (WS-I) Simple SOAP Binding Profile Version 1.0, 24 August 2004
http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html

[WSDL11**]** Web Services Description Language (WSDL) 1.1, 15 March 2001
http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[XRD-1.0] E. Hammer, W. Norris, *Extensible Resource Descriptor (XRD) Version 1.0*
http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.html

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 version of the 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 MUST 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.

2.6 Setting service and resource properties.

These operations are not defined in this version of the specification. In the future it will be possible to define or set the properties for the service or resources within the service.

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. All URLs should be absolute or path-absolute following the rules defined in [RFC 4918] Section 8.3.

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/"

3.6 Etoken

This is an opaque token returned to identify the current change status of an entity. Whenever an entity is changed the token will take on a new value. An unchanged etoken value DOES NOT imply byte-for-byte equality with the stored entity. The service may choose to modify properties under its control, for example last-modification times. However, an entity with an unchanged etoken can be safely updated by a client holding that etoken.

4 Overview of the CalWS protocol

CalWs operations and data elements are defined in this specification. Many of the operations result in the transmission of data as defined in [RFC 5545].

SOAP 1.1 messages consist of three elements: an envelope, header data, and a message body. CalWs request-response elements MUST be enclosed within the SOAP message body. CalWs SOAP messages MUST conform to [WT-I-Basic] and [WS-I-Bind]. A single CalWs SOAP message MUST contain only one service request or a single service response).

The basic process for using SOAP for CalWs operations is:

A system entity acting as a CalWs requester transmits a CalWs request element within the body of a SOAP message to a system entity acting as a CalWs responder. The CalWs requester MUST NOT include more than one CalWs request per SOAP message or include any additional XML elements in the SOAP body.

The CalWs responder MUST return either a CalWs response element within the body of another SOAP message or generate a SOAP fault. The CalWs responder MUST NOT include more than one CalWs response per SOAP message or include any additional XML elements in the SOAP body. If a CalWs responder cannot, for some reason, process a CalWs request, it MUST generate a SOAP fault. (SOAP 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)

4.1 Discovery

CalWs implementers (service providers) MUST provide WSDL WSDL11 to describe their implementations. This WSDL MAY or may not be made public via a standard discovery mechanism (such as UDDI) or other method.

In addition, it is REQUIRED that the CalWs implementation include the Properties operation to provide dynamic information regarding CalWs capabilities, options, etc. that are supported.

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 the execution of a properties operation specifying the target resource. 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 single calendar object
- Multiget of one or more calendar objects
- 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 a timezone service (which may be the same service acting as a timezone server) through the server properties object. The timezone service operations are defined in [draft-timezones].

4.6 Error conditions

Each operation on the calendar system has a number of pre-conditions and post-conditions that apply. If any of these are violated the response message will have a status code indicating an error occurred and will contain an error response element providing details.

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 message.

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.

Example: error with CalDAV error condition

```
<?xml version="1.0" encoding="utf-8"
      xmlns:CW="http://docs.oasis-open.org/ns/wscal/calws-soap"
      xmlns:C="urn:ietf:params:xml:ns:caldav" ?>
<CW:error>
  <C:supported-filter>
    <C:prop-filter name="X-ABC-GUID"/>
  </C:supported-filter>
  <CW:description>Unknown property </CW:description>
</CW:error>
```

5 CalWs-SOAP Messages.

This section describes the common elements and structure of CalWs-SOAP messages. The conventions followed are shown in Table 1

Header	Description	Values	Meaning
Field	Name of the field.		Prefixed with / to indicate a child-relationship Prefixed with # to indicate an attribute
Type	XML schema type		
#	Cardinality of the field	1	One occurrence
		0..1	Zero or one occurrence
		0..*	Zero or more occurrences
		1..*	One or more occurrences
?	Presence	Y	Always required
		N	Optional
		C	Conditional - dependent on the message or other conditions
Description	A short description		

Table 1: Field column descriptions

5.1 Common Elements and types

The following tables define the base types for requests and responses. All CalWs-SOAP messages and responses are based on these types.

All requests must include an href which specifies the target for the request. There is also an id attribute which will be copied into the response to help identify it.

Field	Type	#	?	Description
href	string	1	Y	Required in each request to identify the target of the message.
#id	int	1	N	Useful for tying responses to requests.

Table 2: BaseRequestType elements

A response may include an error response element of type ErrorResponseType. This element will be returned in response messages when some form of processing error occurs and provides further information on the error beyond the basic status code.

Field	Type	#	?	Description
?	ErrorCodeType	1	Y	One of the error code elements defined below
description	string	0..1	N	Optional descriptive message

Table 3: ErrorResponseType elements

ErrorCodeType

The following table defines the error codes that may be returned as an element of ErrorCodeType.

Field	Type	#	?	Description
target-exists	ErrorCodeType	1	Y	
not-calendar-data	ErrorCodeType	1	Y	
invalid-calendar-data	ErrorCodeType	1	Y	
invalid-calendar-object-resource	ErrorCodeType	1	Y	
unsupported-calendar-component	ErrorCodeType	1	Y	
invalid-calendar-collection-location	ErrorCodeType	1	Y	
exceeds-max-resource-size	ErrorCodeType	1	Y	
before-min-date-time	ErrorCodeType	1	Y	
after-max-date-time	ErrorCodeType	1	Y	
too-many-instances	ErrorCodeType	1	Y	
too-many-attendees-per-instance	ErrorCodeType	1	Y	

Table 4: ErrorCodeType definitions

BaseResponseType

Field	Type	#	?	Description
#id	int	1	N	Copied over from the request
status	StatusType	1	Y	Give the overall status of the response
message	string	0..1	N	Optional explanatory message
errorResponse	ErrorCodeType	0..1	N	Required for a status of Error.

Table 5: BaseResponseType elements

6 Properties and link relations

6.1 Property and relation-type URIs

In the XRD entity returned properties and related services and entities are defined by absolute URIs which correspond to the extended relation type defined in [Web-Linking] Section 4.2. These URIs do NOT correspond to any real entity on the server and clients should not attempt to retrieve any data at that target.

Certain of these property URIs correspond to CalDAV preconditions. Each URL is prefixed by the CalWS relations and properties namespace http://docs.oasis-open.org/ns/wscal/calws. Those properties which correspond to CalDAV properties have the additional path element "**caldav/**", for example

```
http://docs.oasis-open.org/ns/wscal/calws/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://docs.oasis-open.org/ns/wscal/calws**.**

6.2 supported-features property.

http://docs.oasis-open.org/ns/wscal/calws/supported-features

This property defines the features supported by the target. All resources contained and managed by the service should return this property. The value is a comma separated list containing one or more of the following

- calendar-access - the service supports all MUST requirements in this specification

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/supported-features"
          >calendar-access</Property>
```

6.3 max-attendees-per-instance

http://docs.oasis-open.org/ns/wscal/calws/max-attendees-per-instance

An integer value defining the maximum number of attendees allowed per event or task.

6.4 max-date-time

http://docs.oasis-open.org/ns/wscal/calws/max-date-time

Defines the maximum date/time allowed on an event or task

6.5 max-instances

http://docs.oasis-open.org/ns/wscal/calws/max-instances

An integer value defining the maximum number of instances allowed per event or task

6.6 max-resource-size

http://docs.oasis-open.org/ns/wscal/calws/max-resource-size

An integer value defining the maximum size of a resource in octets that the server is willing to accept when a calendar object resource is stored in a calendar collection.

6.7 min-date-time

http://docs.oasis-open.org/ns/wscal/calws/min-date-time

Provides a DATE-TIME value indicating the earliest date and time (in UTC) that the server is willing to accept for any DATE or DATE-TIME value in a calendar object resource stored in a calendar collection.

6.8 description

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

Provides some descriptive text for the targeted collection.

6.9 timezone-service relation.

http://docs.oasis-open.org/ns/wscal/calws/timezone-service

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.

```
<Link rel="http://docs.oasis-open.org/ns/wscal/calws/calws/timezone-service"
          href="http://example.com/tz" />
```

6.10 principal-home relation.

http://docs.oasis-open.org/ns/wscal/calws/principal-home

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

```
<Link rel="http://docs.oasis-open.org/ns/wscal/calws/principal-home"
          href="http://example.com/user/fred" />
```

6.11 current-principal-freebusy relation.

http://docs.oasis-open.org/ns/wscal/calws/current-principal-freebusy

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

```
<Link rel="http://docs.oasis-open.org/ns/wscal/calws/current-principal-freebusy"
          href="http://example.com/freebusy/user/fred" />
```

6.12 principal-freebusy relation.

http://docs.oasis-open.org/ns/wscal/calws/principal-freebusy

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

```
<Link rel="http://docs.oasis-open.org/ns/wscal/calws/principal-freebusy"
          href="http://example.com/freebusy" />
```

6.13 child-collection relation.

http://docs.oasis-open.org/ns/wscal/calws/child-collection

Provides information about a child collections for the target. The href attribute gives the URI of the collection. The element should only have CalWS child elements giving the type of the collection, that is the CalWS:collection link property and the CalWS-calendar-collection link property. This allows clients to determine the structure of a hierarchical system by targeting each of the child collections in turn.

The xrd:title child element of the link element provides a description for the child-collection.

```
<Link rel="http://http://docs.oasis-open.org/ns/wscal/calws/child-collection"
      href="http://example.com/calws/user/fred/calendar">
  <Title xml:lang="en">Calendar</Title>
  <Property type="http://docs.oasis-open.org/ns/wscal/calws/collection"
            xsi:nil="true" />
  <Property type="http://docs.oasis-open.org/ns/wscal/calws/calendar-collection"
            xsi:nil="true" />
</Link>
```

6.14 created link property

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

Appears within a link relation describing collections or entities. The value is a date-time as defined in [RFC 3339] Section 5.6

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/created"
          >1985-04-12T23:20:50.52Z</Property>
```

6.15 last-modified property

http://docs.oasis-open.org/ns/wscal/calws/last-modified

Appears within an xrd object describing collections or entities. The value is the same format as would appear in the Last-Modified header and is defined in [RFC 2616] Section 3.3.1

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/last-modified"
          >Mon, 12 Jan 1998 09:25:56 GMT</Property>
```

6.16 displayname property

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

Appears within an xrd object describing collections or entities. The value is a localized name for the entity or collection.

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/displayname"
          >My Calendar</Property>
```

6.17 timezone property

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

Appears within an xrd object describing collections. The value is a text timezone identifier.

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/timezone"
          >America/New_York</Property>
```

6.18 owner property

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

Appears within an xrd object describing collections or entities. The value is a server specific uri.

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/owner"
          >/principals/users/mike</Property>
```

6.19 collection link property

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

Appears within a link relation describing collections or entities. The property takes no value and indicates that this child element is a collection.

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/collection"
          xsi:nil="true" />
```

6.20 calendar-collection link property

http://docs.oasis-open.org/ns/wscal/calws/calendar-collection

Appears within a link relation describing collections or entities. The property takes no value and indicates that this child element is a calendar collection.

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/calendar-collection"
          xsi:nil="true" />
```

6.21 CalWS:privilege-set XML element

http://docs.oasis-open.org/ns/wscal/calws:privilege-set

Appears within a link relation describing collections or entities and specifies the set of privileges allowed to the current authenticated principal for that collection or entity.

```
<!ELEMENT calws:privilege-set (calws:privilege*)>
<!ELEMENT calws:privilege ANY>
```

Each privilege element defines a privilege or access right. The following set is currently defined

- CalWS: Read - current principal has read access
- CalWS: Write - current principal has write access

```
<calWS:privilege-set>
  <calWS:privilege><calWS:read></calWS:privilege>
  <calWS:privilege><calWS:write></calWS:privilege>
</calWS:privilege-set>
```

6.22 CalWS:supported-calendar-component-set XML element

http://docs.oasis-open.org/ns/wscal/calws:supported-calendar-component-set

Appears within a link relation and specifies the set of component types allowed in the targeted collection.

The elements within the returned supported-calendar-component-set element are any component element from the xcal:IcalendarType specification.

7 Retrieving Collection and Service Properties

Properties, related services and locations are obtained from the service or from service resources in the form of an XRD document as defined by [XRD-1.0].

The CalWs-SOAP getProperties request is used to fetch properties. The href can target the service with a path of "/" or any entity within the service.

The service properties define the global limits and defaults. Any 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. The tables below show the fiedls for request and response.

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request. "/" for the service.

Table 6: GetPropertiesType fields

Field	Type	#	?	Description
XRD	XRD	0..1	C	Returned fro an OK response. Type is an XRD object populated with properties and references defined in Section ?

Table 7: GetPropertiesResponseType fields

7.1 Example - retrieving server properties:

```
>>Request

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:getProperties xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
        xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
        xmlns:ns4="urn:ietf:params:xml:ns:caldav"
        xmlns:ns5="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/</ns2:href>
    </ns2:getProperties>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>>Response

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:getPropertiesResponse
          xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
          xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
          xmlns:ns4="urn:ietf:params:xml:ns:caldav"
         xmlns:ns5="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns5:XRD>
        <ns5:Subject>/ucaldav/</ns5:Subject>
        <ns5:Property
            type="http://docs.oasis-open.org/ns/wscal/calws/last-modified"
                >Tue, 29 Mar 2011 17:47:25 +0000</ns5:Property>
        <ns5:Property
            type="http://docs.oasis-open.org/ns/wscal/calws/owner"
                >/ucaldav/principals/users/public-user/</ns5:Property>
        <ns5:Property
            type="http://docs.oasis-open.org/ns/wscal/calws/max-resource-size"
```

```
               >100000</ns5:Property>
        <ns5:Property type="http://docs.oasis-open.org/ns/wscal/calws/collection"/>
        <ns5:Property
            type="http://docs.oasis-open.org/ns/wscal/calws/max-instances"
               >1000</ns5:Property>
        <ns5:Property
           type="http://docs.oasis-open.org/ns/wscal/calws/principal-home"
                >/ucaldav/user/douglm/</ns5:Property>
      </ns5:XRD>
    </ns2:getPropertiesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

8 Creating Calendar Object Resources

Creating calendar object resources is carried out by using a CalWs-SOAP addItem request targeted at the parent collection and containing the resource to be created. The response will contain the href of the newly created object.

The icalendar entity in the request MUST contain only a single calendaring entity with any related overrides.

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request.
icalendar	xcal:IcalendarType	1	Y	The entity to be created

Table 8: AddItemType fields

The service will respond with an AddItemResponseType giving either the href and etoken of the new entity or an error response.

Field	Type	#	?	Description
href	string	0..1	N	Href of the new entity for a successful request.
etoken	string	0..1	N	Etoken for the new entity

Table 9: AddItemResponseType additional fields

8.1 Preconditions for Calendar Object Creation

- **CalWS:target-exists**: The entity already exists.
- **CalWS:not-calendar-data:** The resource submitted MUST be a supported media type (i.e., iCalendar) for calendar object resources;
- **CalWS:invalid-calendar-data:** The resource submitted 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 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 request MUST contain a type of calendar component that is supported in the targeted calendar collection;
- **CalWS:uid-conflict:** The resource submitted in the 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:exceeds-max-resource-size:** The resource submitted in the 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 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 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 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 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;

8.2 Example - successful addItem:

```
>>Request

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:addItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                 xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar</ns2:href>
      <ns3:icalendar>
        <ns3:vcalendar>
          <ns3:components>
            <ns3:vevent>
              <ns3:properties>
                <ns3:uid>
                  <ns3:text>1302064354993</ns3:text>
                </ns3:uid>
                <ns3:summary>
                  <ns3:text>try this</ns3:text>
                </ns3:summary>
                <ns3:dtstart>
                  <ns3:date-time>20110406T150000Z</ns3:date-time>
                </ns3:dtstart>
                <ns3:dtend>
                  <ns3:date-time>20110406T160000Z</ns3:date-time>
                </ns3:dtend>
              </ns3:properties>
            </ns3:vevent>
          </ns3:components>
        </ns3:vcalendar>
      </ns3:icalendar>
    </ns2:addItem>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>>Response

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:addItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                         xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                         xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>OK</ns2:status>
      <ns2:href>/user/douglm/calendar/1302064354993.ics</ns2:href>
      <ns2:etoken>"20110406T155741Z-0"</ns2:etoken>
    </ns2:addItemResponse>
```

```
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

8.3 Example - unsuccessful addItem:

```
   TBD
```

9 Retrieving resources

Fetching calendar object resources is carried out by using a CalWs-SOAP fetchItem request with an href specifying the entity to be fetched. The response will contain the calendaring entity with any related overrides.

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request.

Table 10: FetchItemType fields

The service will respond with a FetchItemResponseType containing either the etoken and the entity or an error response.

Field	Type	#	?	Description
etoken	string	0..1	N	The etoken for the fetched entity
icalendar	xcal:IcalendarType	0..1	N	The entity to be created

Table 11: FetchItemResponseType additional fields

9.1 Example - successful fetchItem:

```
>>Request

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                   xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                   xmlns:ns5="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
    </ns2:fetchItem>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>>Response

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                           xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                           xmlns:ns5="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>OK</ns2:status>
      <ns2:etoken>"20110406T155741Z-0"</ns2:etoken>
      <ns3:icalendar>
        <ns3:vcalendar>
          <ns3:properties>
            <ns3:prodid>
              <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
            </ns3:prodid>
            <ns3:version>
              <ns3:text>2.0</ns3:text>
            </ns3:version>
          </ns3:properties>
          <ns3:components>
            <ns3:vevent>
              <ns3:properties>
                <ns3:created>
```

```
                <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
              </ns3:created>
              <ns3:dtend>
                <ns3:date-time>20110406T160000Z</ns3:date-time>
              </ns3:dtend>
              <ns3:dtstamp>
                <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
              </ns3:dtstamp>
              <ns3:dtstart>
                <ns3:date-time>20110406T150000Z</ns3:date-time>
              </ns3:dtstart>
              <ns3:last-modified>
                <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
              </ns3:last-modified>
              <ns3:summary>
                <ns3:text>try this</ns3:text>
              </ns3:summary>
              <ns3:uid>
                <ns3:text>1302105461170</ns3:text>
              </ns3:uid>
            </ns3:properties>
          </ns3:vevent>
        </ns3:components>
      </ns3:vcalendar>
    </ns3:icalendar>
  </ns2:fetchItemResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

9.2 Example - unsuccessful fetchItem:

```
>>Request

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                   xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                   xmlns:ns5="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
    </ns2:fetchItem>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>>Response

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                           xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                           xmlns:ns5="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>Error</ns2:status>
      <ns2:errorResponse>
        <ns2:targetDoesNotExist/>
      </ns2:errorResponse>
    </ns2:fetchItemResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10 Updating resources

Calendar entity updates apply changes to a data model which has the form:

- An iCalendar element contains...
- a single vCalendar element which contains...
- one or more calendaring components, event, task etc each of which contain...
- zero or more components, alarms etc or one or more properties each of which contains...
- zero or more parameters and one or more values.

This we have a nested structure which does recurse to a limited extent and looks like

```
<icalendar>
  <vcalendar>
    <components>
      <vevent>
        <properties>
          <uid>
            <text>1302064354993-a</text>
          </uid>
          <summary>
            <text>try this</text>
          </summary>
          <dtstart>
            <date-time>2011-07-18T15:00:00Z</date-time>
          </dtstart>
          <dtend>
            <date-time>2011-07-18T16:00:00Z</date-time>
          </dtend>
        </properties>
      </vevent>
    </components>
  </vcalendar>
</icalendar>
```

The update approach described here only allows for updating a single calendar entity, though that entity may consist of more than one component, for example an override to a repeating event.

Resources are updated with the CalWs-SOAP updateItem request. The request contains the href of the entity to be updated, the current etoken for that entity and the updates. The updates take the form if nested selections of an element from the current level in the data. The outermost selection is always for a vcalendar element - we ignore the icalendar element. Nested within that outer selection is one for the components element followed by selections on the entity, event, task etc and so on.

Only 3 kinds of update may be applied at any point:

- Add - components, properties or parameters
- Remove - components, properties or parameters
- Change - property or parameter values

Preconditions as specified in Preconditions for Calendar Object Creation are applicable. The response will indicate success or failure of the update. If the etoken value does not match that held by the service a mismatchEtoken error status will be returned. The client should re-fetch the entity to refresh its cache and then retry the update based on the new entity values and etoken.

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request.
etoken	string	1	Y	The etoken held by the client for that entity
select	ComponentSelectionType	1..*	Y	Must select vcalendar

Table 12: UpdateItemType fields

The ComponentsSelectionType contains three repeating child elements. The first allows for selection of nested components which can then be updated. The next allows addition of entire components and the last allows for the removal of components.

Field	Type	#	?	Description
component	ComponentSelectionType	0..1	N	Used to match against a component in the target
add	ComponentReferenceType	0..1	N	Supplies components to add
remove	ComponentReferenceType	0..1	N	Species components to remove

Table 13: ComponentsSelectionType fields

The PropertiesSelectionType follows the same pattern, selecting properties to update, add or remove.

Field	Type	#	?	Description
property	PropertySelectionType	0..1	N	Used to match against a property in the target
add	PropertyReferenceType	0..1	N	Supplies properties to add
remove	PropertyReferenceType	0..1	N	Species properties to remove

Table 14: PropertiesSelectionType fields

To complete that pattern there is also a ParametersSelectionType used to select property parameters for update or removal and to supply new parameters.

Field	Type	#	?	Description
parameter	ParameterSelectionType	0..1	N	Used to match against a parameter in the target
add	ParameterReferenceType	0..1	N	Supplies parameters to add
remove	ParameterReferenceType	0..1	N	Species parameters to remove

Table 15: ParametersSelectionType fields

Each of these refers to a reference type. These either provide a complete entity for addition or identify the entity for removal. The three reference types are:

Field	Type	#	?	Description
Any valid iCalendar component name	xcal:BaseComponentType	1	Y	Either a complete component or sufficient to identify it.

Table 16: ComponentReferenceType fields

Field	Type	#	?	Description
Any valid iCalendar property name	xcal:BasePropertyType	1	Y	Either a complete property or sufficient to identify it or provide a new value, depending on usage.

Table 17: PropertyReferenceType fields

Field	Type	#	?	Description
Any valid iCalendar parameter name	xcal:BaseParameterType	1	Y	Either a complete parameter or sufficient to identify it or provide a new value, depending on usage.

Table 18: ParameterReferenceType fields

To complete the picture we have three selection types for component, property and parameter. Each of these identifies the entity to be updated, possible selections of the sub-elements and a possible change to values.

ComponentSelectionType contains three child elements. The first is any valid icalendar component element which is to be matched at the current level.

The optional properties selection allows selection and possible updates to the properties of the component. An iCalendar properties element cannot take a value so the only updates possible are addition and removal of properties. Nested properties may be selected for updates.

The optional components selection allows selection and possible updates to the nested icalendar components element of the component. An iCalendar components element cannot take a value so the only updates possible are addition and removal of components. Nested components may be selected for updates.

Field	Type	#	?	Description
Any valid iCalendar component name	xcal:VcalendarType xcal:BaseComponentType xcal:VAlarmType	1	Y	Used to match against an element in the target
properties	PropertiesSelectionType	0..1	N	To match the properties element
components	ComponentsSelectionType	0..1	N	To match the components element

Table 19: ComponentSelectionType fields

PropertySelectionType contains three child elements. The first is any valid icalendar property element which is to be matched at the current level.

The optional parameters selection allows selection and possible updates to the parameters of the property.

The optional change element allows a change to the value of the property. The new value is specified by supplying an iCalendar property with the desired value(s). Any parameters will be ignored.

Field	Type	#	?	Description
Any valid iCalendar property name	xcal:BasePropertyType	1	Y	Used to match against an element in the target
parameters	ParametersSelectionType	0..1	N	To match the parameters element
change	PropertyReferenceType	0..1	N	To provide a new value

Table 20: PropertySelectionType fields

Lastly, there is the ParameterSelectionType which contains two child elements. The first is any valid icalendar parameter element which is to be matched at the current level.

The optional change element allows a change to the value of the parameter. The new value is specified by supplying an iCalendar parameter with the desired value(s).

Field	Type	#	?	Description
Any valid iCalendar parameter name	xcal:BaseParameter Type	1	Y	Used to match against an element in the target
change	ParameterReferenceType	0..1	N	To provide a new value

Table 21: ParameterSelectionType fields

For a successful update the service will respond with a UpdateItemResponseType containing the status and the new etoken.

Field	Type	#	?	Description
etoken	string	0..1	N	The new etoken for the updated entity

Table 22: UpdateItemResponseType additional fields

The etoken value should be used to replace the value held by the client.

10.1 Etokens and concurrent updates

The etoken is used to allow a service to determine whether or not it is safe to carry out an update requested by the client. The etoken should be opaque to the client but will probably in fact be a structured value. Calendaring transactions have some special characteristics which make it desirable to allow certain non-conflicting updates to take place while other changes are taking place. For example, meeting requests with a large number of attendees can be frequently updated by the server as a result of attendee participation status changes. If we use an unstructured etoken to represent all changes this can make it very difficult to update an event while those participation status changes are being made.

If, on the other hand, the token has a section indicating that only participation status changes have been made, then other changes can take place. For a reference on implementing such a token see "Avoiding Conflicts when Updating Scheduling Object Resources" in [draft caldav-sched]. This describes the use of a schedule-tag.

10.2 Example - successful update:

The event to be updated is represented by the following XML.

```
      <ns3:icalendar>
        <ns3:vcalendar>
          <ns3:components>
            <ns3:vevent>
```

```
            <ns3:properties>
              <ns3:uid>
                <ns3:text>1302064354993-a</ns3:text>
              </ns3:uid>
              <ns3:summary>
                <ns3:text>try this</ns3:text>
              </ns3:summary>
              <ns3:dtstart>
                <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
              </ns3:dtstart>
              <ns3:dtend>
                <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
              </ns3:dtend>
            </ns3:properties>
          </ns3:vevent>
        </ns3:components>
      </ns3:vcalendar>
    </ns3:icalendar>
```

In the following example we make the following changes to the above event:

- Change the summary
- Change the dtstart - add a tzid and change the value to local time
- Add some categories

We first select an event by specifying the uid value and then, from that event, we select the properties, then select and change the appropriate properties.

```
>>Request

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:updateItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                    xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                    xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar/1302064354993-a.ics</ns2:href>
      <ns2:etoken>"20110802T032608Z-0"   null</ns2:etoken>
      <ns2:select>
        <ns3:vcalendar/>
        <ns2:components>
          <ns2:component>
            <ns3:vevent>
              <ns3:properties>
                <ns3:uid>
                  <ns3:text>1302064354993-a</ns3:text>
                </ns3:uid>
              </ns3:properties>
            </ns3:vevent>
            <ns2:properties>
              <ns2:property>
                <ns3:dtstart>
                  <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
                </ns3:dtstart>
                <ns2:parameters>
                  <ns2:add>
                    <ns3:tzid>
                      <ns3:text>America/New_York</ns3:text>
                    </ns3:tzid>
                  </ns2:add>
                </ns2:parameters>
                <ns2:change>
                  <ns3:dtstart>
                    <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
                  </ns3:dtstart>
                </ns2:change>
              </ns2:property>
```

```
              <ns2:property>
                <ns3:summary>
                  <ns3:text>try this</ns3:text>
                </ns3:summary>
                <ns2:change>
                  <ns3:summary>
                    <ns3:text>A changed summary - again and again and again</ns3:text>
                  </ns3:summary>
                </ns2:change>
              </ns2:property>
              <ns2:add>
                <ns3:categories>
                  <ns3:text>newcategory-2</ns3:text>
                  <ns3:text>resources</ns3:text>
                  <ns3:text>paper</ns3:text>
                </ns3:categories>
              </ns2:add>
            </ns2:properties>
          </ns2:component>
        </ns2:components>
      </ns2:select>
    </ns2:updateItem>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>>Response

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:updateItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                            xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                            xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0"
                            id="0">
      <ns2:status>OK</ns2:status>
    </ns2:updateItemResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

10.3 Other updates:

Based on the example above we present some XML fragments for different kinds of update. These include:

- Addition of properties
- Removal of properties
- Addition of parameters to properties
- Removal of parameters from properties
- Changing parameter values.

The examples all start with the selection of the vevent properties element. First we have the XML for the addition of a tzid to the start date/time. Here we select the dtstart, then the parameters element then add a tzid parameter and change the value of the date and time

```
            <ns2:properties>
              <ns2:property>
                <ns3:dtstart>
                  <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
                </ns3:dtstart>
                <ns2:parameters>
                  <ns2:add>
                    <ns3:tzid>
                      <ns3:text>America/New_York</ns3:text>
```

```
              </ns3:tzid>
            </ns2:add>
          </ns2:parameters>
          <ns2:change>
            <ns3:dtstart>
              <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
            </ns3:dtstart>
          </ns2:change>
        </ns2:property>
      </ns2:properties>
```

In this example we add two categories to the event.

```
      <ns2:properties>
        <ns2:add>
          <ns3:categories>
            <ns3:text>paper</ns3:text>
          </ns3:categories>
        </ns2:add>
        <ns2:add>
          <ns3:categories>
            <ns3:text>resources</ns3:text>
          </ns3:categories>
        </ns2:add>
      </ns2:properties>
```

In this example we add a duration and remove the dtend.

```
      <ns2:properties>
        <ns2:add>
          <ns3:duration>
            <ns3:duration>PT1H</ns3:duration>
          </ns3:duration>
        </ns2:add>
        <ns2:remove>
          <ns3:dtend>
            <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
          </ns3:dtend>
        </ns2:remove>
      </ns2:properties>
```

In this example we change the dtstart timezone identifier.

```
      <ns2:properties>
        <ns2:property>
          <ns3:dtstart>
            <ns3:parameters>
              <ns3:tzid>
                <ns3:text>America/New_York</ns3:text>
              </ns3:tzid>
            </ns3:parameters>
            <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
          </ns3:dtstart>
          <ns2:parameters>
            <ns2:parameter>
              <ns3:tzid>
                <ns3:text>America/New_York</ns3:text>
              </ns3:tzid>
              <ns2:change>
                <ns3:tzid>
                  <ns3:text>America/Montreal</ns3:text>
                </ns3:tzid>
              </ns2:change>
            </ns2:parameter>
          </ns2:parameters>
        </ns2:property>
      </ns2:properties>
```

10.4 Creating an update message.

The update can be created in many ways but the most common approach is to build the update while modifications take place or to create one as the result of comparing old and new versions. It appears that comparing XML for differences is difficult. However, we can take advantage of the structure of calendaring entities to simplify the process. There are implementations available which take the diff approach to producing an update stream.

There are some special cases to consider when comparing. Some properties are multi-valued and may themselves appear more than once. There is no semantic information implied by any grouping though parameters may need to be taken into account. These properties need to be normalized before comparison and when updating them we produce a change which treats each value as a single property.

These properties are

- categories
- exdate
- freebusy
- rdate

This normalization can take place before comparison.

Some properties are multi-valued and may only appear once. At the moment the only standard property is resource which may take a comma separated list. This should be treated as a single multi-valued property when comparing. The order is unimportant. Sorting the values may help.

Some properties may appear multiple times, for example comment. Comparison should take account of parameters. Ordering all properties appropriately allows for relatively simple comparison.

11 Deletion of resources

Deletion of calendar object resources is carried out by using a CalWs-SOAP deleteItem request with an href specifying the entity to be deleted. The deleteItem request is not valid when the href specifies a collection.

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request.

Table 23: DeleteItemType fields

The service will respond with a DeleteItemResponseType containing the status and a possible error response. There are no additional elements.

11.1 Example - successful deleteItem:

```
>>Request

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                    xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                    xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar/1302620814655.ics</ns2:href>
    </ns2:deleteItem>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>>Response

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                            xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                            xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>OK</ns2:status>
    </ns2:deleteItemResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

11.2 Example - unsuccessful deleteItem:

```
>>Request

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                    xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                    xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
    </ns2:deleteItem>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>>Response

<?xml version="1.0" encoding="UTF-8"?>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                            xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                            xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>Error</ns2:status>
      <ns2:errorResponse>
        <ns2:targetDoesNotExist/>
      </ns2:errorResponse>
    </ns2:deleteItemResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

12 Querying calendar resources

Querying provides a mechanism by which information can be obtained from the service through possibly complex queries. A skeleton icalendar entity can be provided 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.

12.1 Calendar Query common types

The UTCTimeRangeType is used in a number of places to define a time range within which components must appear or property values must lie. The values are UTC time-date, the start is inclusive and the end is exclusive.

Field	Type	#	?	Description
start	UTC date-time	1	Y	UTC inclusive start
end	UTC date-time	1	Y	UTC exclusive end

Table 24: UTCTimeRangeType elements

The TextMatchType is used to match text values in properties and parameters. The collation attribute species a collation as defined in [RFC 4790].

Servers are REQUIRED to support the "i;ascii-casemap" and "i;octet" collations which provide a basic case insensitive and case sensitive match respectively.

Elements of this type take a string value which is matched according to the attributes.

Field	Type	#	?	Description
#collation	String	0..1	N	Collation name from [RFC 4790]. "
#negate-condition	boolean	0..1	N	if "true" negates the condition

Table 25: TextMatchType attributes

12.2 CompFilterType

This type defines a search query for the calendar query operation. It specifies the component types to return, absence tests or basic matching operations on properties and time ranges.

The top level comp-filter element (which must match "vcalendar" may contain zero or more comp-filter elements to match events, tasks or other contained components. These in turn may contain further nested comp-filter elements to match further levels of nested components.

Each may also contain prop-filter elements to test for the absence of properties or to match values.

Only logical conjunctions are supported, that is, all elements must match for the full expression to match.

Field	Type	#	?	Description
#name	String	1	Y	Specifies the component name - must be "vcalendar" at the top level. May be "vevent", vtodo" for example at the next level. Value is case-insignificant.
is-not-defined	empty	0..1	N	Only this element or one or more of time-range, prop-filter or comp-filter may be present
time-range	UTCTimeRangeType	0..1	N	
comp-filter	CompFilterType	1	Y	Match against contained components
prop-filter	PropFilterType	0..n	N	Match against component properties

Table 26: CompFilterType elements

12.3 PropFilterType

The prop-filter element may test for the absence of a property or match values or specify zero or more ParamFilterType elements to match against parameters.

Only logical conjunctions are supported, that is, all elements must match for the full expression to match.

Field	Type	#	?	Description
#name	String	1	Y	Specifies the property name. Value is case-insignificant.
is-not-defined	empty	0..1	N	Only this element or optionally one of time-range or text-match followed by param-filter
time-range	UTCTimeRangeType	0..1	N	
text-match	TextMatchtype	0..1	N	
param-filter	ParamFilterType	0..n	N	Match against property parameters

Table 27: PropFilterType elements

12.4 ParamFilterType

The ParamFilterType element may test for the absence of a parameter or match a value.

Field	Type	#	?	Description
#name	String	1	Y	Specifies the property name. Value is case-insignificant.
is-not-defined	empty	0..1	N	Only this element or text-match
text-match	TextMatchtype	0..1	N	

Table 28: ParamFilterType elements

12.5 CalendarQueryType elements

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request. "/" for the service.
allprop	empty	0..1	N	If present specifies all properties should be returned One or none of allprop or icalendar
icalendar	IcalendarType	0..1	N	If present is a valueless icalendar skeleton entity defining which components and properties should be returned. If present allprop must NOT be present.
expand	ExpandType	0..1	N	A subclass of UTCTimeRangeType. Either expand or limitRecurrenceSet may be specified but not both. If specified recurring events are expanded and limited to the supplied time-range. All events times are converted to UTC. This option allows for simplified event handling for certain classes of client.
limitRecurrenceSet	LimitRecurrenceSetType	0..1	N	A subclass of UTCTimeRangeType. Either expand or limitRecurrenceSet may be specified but not both. If specified only overrides that fall within the specified time-range are returned. This helps to limit the size of the result-set when there are many overrides.
depth	String	0..1	N	Species depth for query. "1" => just targeted collection, "infinity" => query targeted and all sub-collections.
filter	FilterType	1	Y	Defines the search filter
/comp-filter	CompFilterType	1	Y	Defines the top-level component

Table 29: CalendarQueryType elements

12.6 Specifying data to be returned

This is achieved by specifying one of the following

- allprop: return all properties and calendar data. (some properties are specified as not being part of the allprop set so are not returned)
- Set the icalendar element. This is an icalendar valueless pattern entity which provides a map of the components and properties to be returned. Neither the pattern nor the returned result need to be valid icalendar entities in that required properties may be absent if unselected.

12.7 Pre/postconditions for calendar queries

The preconditions as defined in [RFC 4791] Section 7.8 apply here. CalDav errors may be reported by the service when preconditions or postconditions are violated.

12.8 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 <<

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:calendarQuery xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                       xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar</ns2:href>
      <ns3:icalendar>
        <ns3:vcalendar>
          <ns3:components>
            <ns3:vevent>
              <ns3:properties>
                <ns3:summary/>
                <ns3:dtstart/>
                <ns3:dtend/>
                <ns3:duration/>
                <ns3:uid/>
                <ns3:recurrence-id/>
                <ns3:rrule/>
                <ns3:rdate/>
                <ns3:exdate/>
              </ns3:properties>
            </ns3:vevent>
          </ns3:components>
        </ns3:vcalendar>
      </ns3:icalendar>
      <ns2:filter>
        <ns2:comp-filter name="vcalendar">
          <ns2:comp-filter name="vevent">
            <ns2:time-range end="20110430T040000Z" start="20110401T040000Z"/>
          </ns2:comp-filter>
        </ns2:comp-filter>
      </ns2:filter>
    </ns2:calendarQuery>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>> Response <<

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:calendarQueryResponse
                       xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
                       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
                       xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>OK</ns2:status>
      <ns2:response>
        <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
        <ns2:etoken>"20110406T155741Z-0"</ns2:etoken>
        <ns2:propstat>
          <ns2:prop>
            <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
```

```
            <ns3:icalendar>
              <ns3:vcalendar>
                <ns3:properties>
                  <ns3:prodid>
                    <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
                  </ns3:prodid>
                  <ns3:version>
                    <ns3:text>2.0</ns3:text>
                  </ns3:version>
                </ns3:properties>
                <ns3:components>
                  <ns3:vevent>
                    <ns3:properties>
                      <ns3:dtend>
                        <ns3:date-time>20110406T160000Z</ns3:date-time>
                      </ns3:dtend>
                      <ns3:dtstart>
                        <ns3:date-time>20110406T150000Z</ns3:date-time>
                      </ns3:dtstart>
                      <ns3:summary>
                        <ns3:text>try this</ns3:text>
                      </ns3:summary>
                      <ns3:uid>
                        <ns3:text>1302105461170</ns3:text>
                      </ns3:uid>
                    </ns3:properties>
                  </ns3:vevent>
                </ns3:components>
              </ns3:vcalendar>
            </ns3:icalendar>
          </ns2:calendar-data>
        </ns2:prop>
        <ns2:status>HTTP/1.1 200 ok</ns2:status>
      </ns2:propstat>
    </ns2:response>
    <ns2:response>
      <ns2:href>/user/douglm/calendar/CAL-00f1fc61-2f021bca-012f-022947f8-
00000006.ics</ns2:href>
      <ns2:etoken>"20110405T140920Z-0"</ns2:etoken>
      <ns2:propstat>
        <ns2:prop>
          <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
            <ns3:icalendar>
              <ns3:vcalendar>
                <ns3:properties>
                  <ns3:prodid>
                    <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
                  </ns3:prodid>
                  <ns3:version>
                    <ns3:text>2.0</ns3:text>
                  </ns3:version>
                </ns3:properties>
                <ns3:components>
                  <ns3:vevent>
                    <ns3:properties>
                      <ns3:duration>
                        <ns3:duration>PT1H</ns3:duration>
                      </ns3:duration>
                      <ns3:dtstart>
                        <ns3:parameters>
                          <ns3:tzid>
                            <ns3:text>America/New_York</ns3:text>
                          </ns3:tzid>
                        </ns3:parameters>
                        <ns3:date-time>20110412T110000</ns3:date-time>
                      </ns3:dtstart>
                      <ns3:summary>
                        <ns3:text>Test recurring event</ns3:text>
                      </ns3:summary>
                      <ns3:uid>
```

```
                        <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
00000006demobedework@mysite.edu</ns3:text>
                      </ns3:uid>
                      <ns3:rrule>
                        <ns3:recur>
                          <ns3:freq>WEEKLY</ns3:freq>
                          <ns3:count>2</ns3:count>
                          <ns3:interval>1</ns3:interval>
                        </ns3:recur>
                      </ns3:rrule>
                    </ns3:properties>
                  </ns3:vevent>
                  <ns3:vevent>
                    <ns3:properties>
                      <ns3:recurrence-id>
                        <ns3:parameters>
                          <ns3:tzid>
                            <ns3:text>America/New_York</ns3:text>
                          </ns3:tzid>
                        </ns3:parameters>
                        <ns3:date-time>20110419T150000Z</ns3:date-time>
                      </ns3:recurrence-id>
                      <ns3:duration>
                        <ns3:duration>PT1H</ns3:duration>
                      </ns3:duration>
                      <ns3:dtstart>
                        <ns3:parameters>
                          <ns3:tzid>
                            <ns3:text>America/New_York</ns3:text>
                          </ns3:tzid>
                        </ns3:parameters>
                        <ns3:date-time>20110419T120000</ns3:date-time>
                      </ns3:dtstart>
                      <ns3:summary>
                        <ns3:text>Test recurring event</ns3:text>
                      </ns3:summary>
                      <ns3:uid>
                        <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
00000006demobedework@mysite.edu</ns3:text>
                      </ns3:uid>
                    </ns3:properties>
                  </ns3:vevent>
                </ns3:components>
              </ns3:vcalendar>
            </ns3:icalendar>
          </ns2:calendar-data>
        </ns2:prop>
        <ns2:status>HTTP/1.1 200 ok</ns2:status>
      </ns2:propstat>
    </ns2:response>
  </ns2:calendarQueryResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

13 Free-busy queries

Freebusy queries are used to obtain freebusy information for a principal. The result contains information only for events to which the current principal has sufficient access and may be affected by components and rules available only to the server (for instance office hours availability).

These queries are carried out by using a CalWs-SOAP freebusyReport request with an href specifying a principal. The freebusyReport request is not valid when the href specifies any entity other than a principal.

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 users.

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 Element values

Three values are provided: href; start; end. Only the hre is required. The start and end are in XML UTC date/time format and are interpreted as follows:

start

Default: If omitted 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: An XML UTC date-time

Example:

```
2011-12-01T10:15:00Z
```

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.

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

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 response as a server may return a different time range than the requested range.

13.2 Examples

The following is an unsuccessful request targeting an invalid resource.

```
>> Request <<

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:freebusyReport
           xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
           xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
           xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/user/douglm/calendar</ns2:href>
      <ns2:time-range>
        <ns2:start>2011-04-01T04:00:00Z</ns2:start>
        <ns2:end>2011-04-30T04:00:00Z</ns2:end>
      </ns2:time-range>
    </ns2:freebusyReport>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>> Response <<

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:freebusyReportResponse
            xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
            xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
            xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>Error</ns2:status>
      <ns2:message>Only principal href supported</ns2:message>
    </ns2:freebusyReportResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The following is an example of a request to retrieve Freebusy data for a user:

```
>> Request <<

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:freebusyReport
           xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
           xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
           xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:href>/principals/users/douglm</ns2:href>
      <ns2:time-range>
        <ns2:start>2011-04-01T04:00:00Z</ns2:start>
        <ns2:end>2011-04-30T04:00:00Z</ns2:end>
      </ns2:time-range>
    </ns2:freebusyReport>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

>> Response <<

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:freebusyReportResponse
            xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
            xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
            xmlns:ns4="http://docs.oasis-open.org/ns/xri/xrd-1.0">
      <ns2:status>OK</ns2:status>
```

```
      <ns3:icalendar>
        <ns3:vcalendar>
          <ns3:properties>
            <ns3:prodid>
              <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
            </ns3:prodid>
            <ns3:version>
              <ns3:text>2.0</ns3:text>
            </ns3:version>
          </ns3:properties>
          <ns3:components>
            <ns3:vfreebusy>
              <ns3:properties>
                <ns3:attendee>
                  <ns3:parameters>
                    <ns3:partstat>
                      <ns3:text>NEEDS-ACTION</ns3:text>
                    </ns3:partstat>
                  </ns3:parameters>
                  <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
                </ns3:attendee>
                <ns3:created>
                  <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
                </ns3:created>
                <ns3:dtend>
                  <ns3:date-time>2011-04-30T00:00:00Z</ns3:date-time>
                </ns3:dtend>
                <ns3:dtstamp>
                  <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
                </ns3:dtstamp>
                <ns3:dtstart>
                  <ns3:date-time>2011-04-01T00:00:00Z</ns3:date-time>
                </ns3:dtstart>
                <ns3:freebusy>
                  <ns3:parameters>
                    <ns3:fbtype>
                      <ns3:text>BUSY</ns3:text>
                    </ns3:fbtype>
                  </ns3:parameters>
                  <ns3:period>
                    <ns3:start>2011-04-06T15:00:00Z</ns3:start>
                    <ns3:end>2011-04-06T16:00:00Z</ns3:end>
                  </ns3:period>
                </ns3:freebusy>
                <ns3:last-modified>
                  <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
                </ns3:last-modified>
                <ns3:organizer>
                  <ns3:parameters/>
                  <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
                </ns3:organizer>
                <ns3:uid>
                  <ns3:text>2UTDVPZ9H0EQL9QISI44SP5IFPC4N75</ns3:text>
                </ns3:uid>
              </ns3:properties>
            </ns3:vfreebusy>
          </ns3:components>
        </ns3:vcalendar>
      </ns3:icalendar>
    </ns2:freebusyReportResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

14 Multiple operations

Each of the previously described operations acts upon a single entity or resource only. Frequently we have the need to update an interconnected set of entities so that we maintain the consistency of the structure. This requires an atomic operation which can successfully update all the entities or roll back the operation on failure.

The MultiOpType operation provides such a feature. It is essentially a wrapper around any of the other operations which guarantees the success of the entire set or a roll back. Using the id attribute for requests, each individual response can be located in the result.

The MultiOpType request takes the following elements

Field	Type	#	?	Description
operations	Sequence of BaseOperationType	1	Y	Contains one or more operations

Table 30: MultiOpType elements

The response type is also simple containing a single element containing all the responses.

Field	Type	#	?	Description
responses	Sequence of BaseResponseType	1	Y	Contains zero or more responses

Table 31: MultiOpResponseType elements

15 CalWS XML Elements

15.1 description XML Element

Name: description

Purpose: May be used in error responses to provide some useful information about the error.

Description: A textual description of the error, which SHOULD be localized if possible. Mosylt of use to developers and debuggers.

```
<!ELEMENT description (#PCDATA) >
```

15.2 error XML Element

Name: error

Purpose: Error responses, particularly 403 Forbidden and 409 Conflict, sometimes need more information to indicate what went wrong. In these cases, servers MAY return an XML response body with a document element of 'error', containing child elements identifying particular condition codes.

Description: Contains at least one XML element, and MUST NOT contain text or mixed content. Any element that is a child of the 'error' element and is not the **description** element is considered to be a precondition or postcondition code. Unrecognized elements MUST be ignored.

```
<!ELEMENT error ANY >
```

15.3 CalWS:href XML Element

Name: href

Purpose: MUST contain a URI or a relative reference.

Description: There may be limits on the value of 'href' depending on the context of its use. Refer to the specification text where 'href' is used to see what limitations apply in each case.

Value: Simple-ref.

```
<!ELEMENT href (#PCDATA)>
```

15.4 CalWS:target-exists XML Element

Name: target-exists

Purpose: MUST contain a URI or a relative reference.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT target-exists EMPTY >
```

15.5 CalWS:not-calendar-data XML Element

Name: not-calendar-data

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT not-calendar-data EMPTY >
```

15.6 CalWS:invalid-calendar-data XML Element

Name: invalid-calendar-data

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

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

15.7 CalWS:invalid-calendar-object-resource XML Element

Name: invalid-calendar-object-resource

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT invalid-calendar-object-resource EMPTY >
```

15.8 CalWS:unsupported-calendar-component XML Element

Name: unsupported-calendar-component

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT unsupported-calendar-component EMPTY >
```

15.9 CalWS:uid-conflict XML Element

Name: uid-conflict

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT uid-conflict (CalWS:href)>
```

15.10 CalWS:invalid-calendar-collection-location XML Element

Name: invalid-calendar-collection-location

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT invalid-calendar-collection-location EMPTY >
```

15.11 CalWS:exceeds-max-resource-size XML Element

Name: exceeds-max-resource-size

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

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

15.12 CalWS:before-min-date-time XML Element

Name: before-min-date-time

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT before-min-date-time EMPTY >
```

15.13 CalWS:after-max-date-time XML Element

Name: after-max-date-time

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT after-max-date-time EMPTY >
```

15.14 CalWS:too-many-instances XML Element

Name: too-many-instances

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT too-many-instances EMPTY >
```

15.15 CalWS:too-many-attendees-per-instance XML Element

Name: too-many-attendees-per-instance

Purpose: CalWS precondition.

Description: See section Preconditions for Calendar Object Creation

```
<!ELEMENT too-many-attendees-per-instance EMPTY >
```

15.16 CalWS:privilege-set

Name: privilege-set

Purpose: Specify access rights to a collection or entity

Description: Appears within a link relation describing collections or entities and specifies the set of privileges allowed to the current authenticated principal for that collection or entity.

```
<!ELEMENT privilege-set (privilege*)>
```

15.17 CalWS:privilege

Name: privilege

Purpose: Specifies a single access right

Description: Each privilege element defines a privilege or access right. The following set is currently defined

```
<!ELEMENT privilege ANY>
```

15.18 CalWS:read

Name: read

Purpose: Specifies read access

```
<!ELEMENT read NONE>
```

15.19 CalWS:write

Name: read

Purpose: Specifies read access

```
<!ELEMENT write NONE>
```

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:

- Cyrus Daboo, Apple

The authors would also like to thank the Calendaring and Scheduling Consortium and the TC-XML committee for help with this specification.

Appendix B. Non-Normative Text

Appendix C. Revision History

Revision	Date	Editor	Changes
01	July 15 2011	M. Douglass	Added etoken to ensure consistent updates. Added a multi op which allows the atomic processing of multiple operations in one request Added an id attribute to requests and responses.
Initial	Mar 15 2011	M. Douglass	Initial publication - a first pass at a rewrite from CalWS-REST