

CALCONNECT DOCUMENT CD1301

Type: Proposal
Title: CalWS-SOAP SOAP Web Services Protocol for Calendaring
Version: 1.0
Date: 2013-01-10
Status: Published
Source: TC XML

This document was subject to public review at the same time as the OASIS WS-Calendar publication of CalWS-SOAP and contains the same content..

This document is copyright ©2013 by The Calendaring and Scheduling Consortium and is licensed under the Creative Commons 3.0 Unported License: <http://creativecommons.org/licenses/by/3.0/>.

CalWS-SOAP Version 1.0

09 November 2012

Technical Committee:

CalConnect TC-XML

Chair:

Michael Douglass (dougln@rpi.edu), [Rensselaer Polytechnic Institute](#)

Editor:

Michael Douglass (dougln@rpi.edu), [Rensselaer Polytechnic Institute](#)

Related work:

This specification is related to:

- RFC 6321 – xCal: iCalendar in XML. <http://www.ietf.org/rfc/rfc6321.txt>
- *WS-Calendar Version 1.0*. Latest version.
<http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.html>

Abstract:

This document describes standard messages and interactions for service interactions with a system that hosts calendar-based information using SOAP. Hosted information can be either traditional personal and enterprise calendar information or services that support XML payloads developed in conformance with the WS-Calendar specification.

Table of Contents

1	Introduction.....	5
1.1	Terminology.....	5
1.2	Normative References.....	5
1.3	Non-normative References.....	6
1.4	Namespace.....	6
2	Issues not addressed by this specification.....	7
2.1	Access Control.....	7
2.2	Provisioning.....	7
2.3	Copy/Move.....	7
2.4	Creating Collections.....	7
2.5	Retrieving collections.....	7
2.6	Setting service and resource properties.....	7
3	CalWS Glossary.....	8
3.1.1	Calendar Object Resource.....	8
3.1.2	Uid.....	8
3.1.3	Collections.....	8
3.1.4	Calendar Collection.....	8
3.1.5	Scheduling Calendar Collection.....	8
3.1.6	Principal Home.....	8
3.1.7	Change token.....	8
4	Basic Calendar Access.....	9
4.1	Overview of the CalWS protocol.....	9
4.1.1	Discovery.....	9
4.1.2	Properties.....	9
4.1.3	Operations.....	9
4.1.4	Calendar Object Resources.....	10
4.1.5	Timezone information.....	10
4.1.6	Error conditions.....	10
4.1.6.1	Example: error with error condition.....	10
4.2	CalWs-SOAP Messages.....	10
4.2.1	Common Elements and types.....	11
4.2.1.1	ErrorCodeType.....	11
4.2.1.2	BaseResponseType.....	13
4.3	Properties.....	13
4.3.1	childCollection.....	13
4.3.2	creationDateTime.....	13

4.3.3	displayName	14
4.3.4	lastModifiedDateTime	14
4.3.5	maxAttendeesPerInstance	14
4.3.6	maxDateTime	14
4.3.7	maxInstances	14
4.3.8	maxResourceSize	14
4.3.9	minDateTime	15
4.3.10	principalHome	15
4.3.11	resourceDescription	15
4.3.12	resourceOwner	15
4.3.13	resourceTimezoneId	15
4.3.14	resourceType	15
4.3.15	supportedCalendarComponentSet	16
4.3.16	supportedFeatures	16
4.3.17	timezoneServer	16
4.3.18	CalWS:privilege-set XML element	17
4.4	Retrieving Collection and Service Properties	17
4.4.1	Example - retrieving server properties:	17
4.5	Creating Calendar Object Resources	18
4.5.1	Preconditions for Calendar Object Creation	18
4.5.2	Example - successful addItem:	19
4.6	Retrieving resources	20
4.6.1	Example - successful fetchItem:	20
4.6.2	Example - unsuccessful fetchItem:	21
4.7	Updating resources	21
4.7.1	Change tokens and concurrent updates	25
4.7.2	Example - successful update:	25
4.7.3	Other updates:	27
4.7.4	Creating an update message	28
4.8	Deletion of resources	28
4.8.1	Example - successful deleteItem:	28
4.8.2	Example - unsuccessful deleteItem:	29
4.9	Querying calendar resources	29
4.9.1	Calendar Query common types	29
4.9.2	CompFilterType	30
4.9.3	PropFilterType	30
4.9.4	ParamFilterType	31
4.9.5	CalendarQueryType elements	32

4.9.6	Specifying data to be returned.....	32
4.9.7	Pre/postconditions for calendar queries.....	32
4.9.8	Time range limited queries.....	33
4.9.9	Example: time range limited retrieval.....	33
4.10	Free-busy queries.....	35
4.10.1	Element values	36
4.10.1.1	start.....	36
4.10.1.2	end.....	36
4.10.2	Examples.....	36
4.11	Multiple operations.....	38
5	Conformance.....	39
5.1	Start, end and duration in calendar components.....	39
5.1.1	Updating, transporting and maintaining start, and and duration.....	39
5.1.2	VEVENT:.....	39
5.1.3	VTODO:.....	39
5.1.4	VJOURNAL:.....	39
5.1.5	VAVAILABILITY.....	40
5.1.6	AVAILABILITY.....	40
5.2	Recurrences.....	40
5.3	Alarms:.....	40
5.4	Unrecognized or unsupported elements.....	40
Appendix A	Acknowledgments.....	41
Appendix B	Revision History.....	42

1 Introduction

The CalWS SOAP 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 4791] Daboo, et al. *Calendar Extensions to WebDAV (CalDAV)*. <http://www.ietf.org/rfc/rfc4791>
- [RFC 6638] Desruisseaux, et al. *CalDAV Scheduling extensions to CalDAV* <http://tools.ietf.org/html/rfc6638>
- [RFC 5545] B. Desruisseaux, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* <http://tools.ietf.org/html/rfc5546>
- [RFC 5546] C. Daboo, *iCalendar Transport-Independent Interoperability Protocol (iTIP)* <http://tools.ietf.org/html/rfc5545>
- [RFC 6321] C. Daboo, M. Douglass, S. Lees *xCal: The XML format for iCalendar* <http://www.ietf.org/rfc/rfc6321>
- [draft-timezones] C. Daboo, M. Douglass: *Timezone 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/>
- [WSDL11] Web Services Description Language (WSDL) 1.1, 15 March 2001 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [WS-Calendar] *WS-Calendar Version 1.0*. 19 January 2011. OASIS Committee Specification <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf>

18 **1.3 Non-normative References**

- [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>

19 **1.4 Namespace**

20 XML namespaces and prefixes used in this standard:

21 Table 1-1: XML Namespaces in this standard

<i>Prefix</i>	<i>Namespace</i>
xcal	urn:ietf:params:xml:ns:icalendar-2.0
CalWS	http://docs.oasis-open.org/ws-calendar/ns/soap

22

23 **2 Issues not addressed by this specification.**

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

26 **2.1 Access Control**

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

29 **2.2 Provisioning**

30 The protocol will not provide any explicit provisioning operations. If it is possible to authenticate or ad-
31 dress a principals calendar resources then they **MUST** be automatically created if necessary or appropri-
32 ate

33 **2.3 Copy/Move**

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

38 **2.4 Creating Collections**

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

41 **2.5 Retrieving collections**

42 This operation is currently undefined.

43 **2.6 Setting service and resource properties.**

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

46 3 CalWS Glossary

47 3.1.1 Calendar Object Resource

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

50 3.1.2 Uid

51 The UID of an event is defined in [RFC 5545] as a "persistent, globally unique identifier for the calendar
52 component". It is in fact, slightly more complicated in that all overrides to a recurring event have the same
53 UID as the master event. Copies of a meeting invitation sent to attendees must also have the same UID.
54 In this protocol the UID is the key by which we locate calendar object resources (see above) and any as-
55 sociated overrides within a calendar collection (see below).

56 3.1.3 Collections

57 A collection is a set of resources which may be entities or other collections. In file systems a collection is
58 commonly referred to as a folder. Collections are referred to by a collection id which is specific to a ser-
59 vice and may take any form. For many systems they will be path-like.

60 3.1.4 Calendar Collection

61 A collection only allowed to contain calendar object resources. The UIDs for components within a calen-
62 dar collection must be unique. The combination of a calendar collection id and the UID MUST be a unique
63 key within a set of resources made available through this service.

64 3.1.5 Scheduling Calendar Collection

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

67 3.1.6 Principal Home

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

70 3.1.7 Change token

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

76 4 Basic Calendar Access

77 This section defines properties, messages and operations sufficient to provide basic access and opera-
78 tions on a calendar store. These are sufficient to store, retrieve and update calendaring entities and to ob-
79 tain various reports on the current state of the store.

80 Any service supporting this protocol MUST return a calendarAccessFeature element in the supportedFea-
81 tures property in the getPropertiesResponse message as specified in supportedFeatures

82 4.1 Overview of the CalWS protocol

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

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

89 The basic process for using SOAP for CalWS operations is:

90 A system entity acting as a CalWS requester transmits a CalWS request element within the body of a
91 SOAP message to a system entity acting as a CalWS responder. The CalWS requester MUST NOT in-
92 clude more than one CalWS request per SOAP message or include any additional XML elements in the
93 SOAP body (though see Section 4.11 for multiple messages packaged in one request).

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

99 4.1.1 Discovery

100 CalWS implementers (service providers) MUST provide a WSDL WSDL11 to describe their implementa-
101 tions. This WSDL MAY or may not be made public via a standard discovery mechanism (such as UDDI)
102 or other method.

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

105 4.1.2 Properties

106 A service or resource will have a number of properties which describe the current state of that service or
107 resource. These properties are accessed through the execution of a properties operation specifying the
108 target resource. See Retrieving Collection and Service Properties below

109 4.1.3 Operations

110 The following operations are defined by this specification:

- 111 • Retrieval and update of service and resource properties
- 112 • Creation of a calendar object
- 113 • Retrieval of a single calendar object
- 114 • Multiget of one or more calendar objects
- 115 • Update of a calendar object
- 116 • Deletion of a calendar object
- 117 • Query
- 118 • Free-busy query
- 119 • Multiple operations

120 **4.1.4 Calendar Object Resources**

121 The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC 4791] section
122 4.2. An additional constraint for CalWS is that no timezone specifications are transferred with the data.

123 **4.1.5 Timezone information**

124 It is assumed that the client and server each have access to a full set of up to date timezone information.
125 Timezones will be referenced by a timezone identifier from the full set of Olson data together with a set of
126 well-known aliases. CalWS services may advertise a timezone service (which may be the same service
127 acting as a timezone server) through the server properties object. The timezone service operations are
128 defined in [draft-timezones]. The service can provide a list of timezone identifiers and aliases.

129 **4.1.6 Error conditions**

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

133 A "precondition" for a method describes the state of the server that must be true for that method to be
134 performed. A "postcondition" of a method describes the state of the server that must be true after that
135 method has been completed. Any violation of these conditions will result in an error response in the mes-
136 sage.

137 Each method specification defines the preconditions that must be satisfied before the method can suc-
138 ceed. A number of postconditions are generally specified which define the state that must exist after the
139 execution of the operation. Preconditions and postconditions are defined as error elements in the CalWS-
140 SOAP XML namespace, "http://docs.oasis-open.org/ws-calendar/ns/soap".

141 **4.1.6.1 Example: error with error condition**

```
142 <?xml version="1.0" encoding="utf-8"  
143     xmlns:CW="http://docs.oasis-open.org/ws-calendar/ns/soap" ?>  
144 <CW:error>  
145   <CW:uidConflict>  
146     <CW:href>/user/mike/calendar/abcd-0123456789.ics</CW:href>  
147   </CW:uidConflict>  
148   <CW:description>Unknown property </CW:description>  
149 </CW:error>
```

150 **4.2 CalWs-SOAP Messages.**

151 This section describes the common elements and structure of CalWs-SOAP messages. The conventions
152 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		

153 *Table 1: Field column descriptions*

154 **4.2.1 Common Elements and types**

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

157 All requests must include an href which specifies the target for the request. There is also an id attribute
158 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.

159 *Table 2: BaseRequestType elements*

160 A response may include an error response element of type ErrorResponse. This element will be re-
161 turned in response messages when some form of processing error occurs and provides further informa-
162 tion 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

163 *Table 3: ErrorResponse elements*

164 **4.2.1.1 ErrorCodeType**

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

Field	Type	Description
forbidden	ForbiddenType	Attempted to carry out a forbidden operation.
targetExists	TargetExistsType	
targetDoesNotExist	TargetDoesNotExistType	The supplied href does not reference an existing resource.
targetNotEntity	TargetNotEntityType	The supplied href does not target an entity. For example a fetch item was attempted against a collection.
notCalendarData	NotCalendarDataType	The supplied entity is not calendar data.
invalidCalendarData	InvalidCalendarDataType	The supplied entity does not represent valid calendar data.
invalidCalendarObjectResource	InvalidCalendarObjectResourceType	The supplied entity does not represent valid calendar data.
unsupportedCalendarComponent	UnsupportedCalendarComponentType	Indicates that the calendar collection does not accept components of the type the client is attempting to store. The accepted component types can be determined by examining the calendar collection properties.
invalidCalendarCollectionLocation	InvalidCalendarCollectionLocationType	Error indicating at least one of two conditions: <ol style="list-style-type: none"> 1. The server does not allow the creation of calendar collections at the given location in its namespace, or 2. The parent collection of the Request-URI exists but cannot accept members
exceedsMaxResourceSize	ExceedsMaxResourceSizeType	Error indicating that the total size of the event or task is too large. The maximum size is set by the target system and can be determined from the properties.
beforeMinDateTime	BeforeMinDateTimeType	Error indicating that the start or end of an event or task is too far into the past. The minimum date is set by the target system and can be determined from the properties.
afterMaxDateTime	AfterMaxDateTimeType	Error indicating that the start or end of an event or task is too far into the future. The maximum date is set by the target system and can be determined from the properties.
tooManyInstances	TooManyInstancesType	Error indicating that a recurring event has too many instances. The maximum number is set by the target system and can be determined from the properties.
tooManyAttendeesPerInstance	TooManyAttendeesPerInstanceType	Error indicating that a scheduling message has too many attendees. The maximum number is set by the target system and can be determined from the properties.
partialSuccess	PartialSuccessType	Indicates that a MultiOpType operation was partially successful. Returned when the operation is marked as non-atomic and one or more sub-operations failed. The entire response needs to be examined to determine failing operations.
missingChangeToken	MissingChangeTokenType	An operation was attempted which required a change token but none was supplied. Note that it appears that the marshalling or demarshalling should handle this as the token is required. It doesn't.
mismatchedChangeToken	MismatchedChangeTokenType	An update operation was attempted with a change token value which does not match that held by the service. The client must refetch the entity to refresh its cached value and token. Note that matching of tokens is a server responsi-

Field	Type	Description
		bility. The token is opaque to the client but probably structured to the server. Certain non-conflicting updates may be allowed even if the token has changed.
invalidFilter	InvalidFilterType	
uidConflict	UidConflictType	An attempt was made to store an entity which would result in more than one entity having equal uids. The entity uid must be unique within a collection. Recurring event or task overrides have the same uid and are considered part of a single entity.

166 Table 4: ErrorCodeType definitions

167 4.2.1.2 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.

168 Table 5: BaseResponseType elements

169 4.3 Properties

170 The getPropertiesResponse message contains 0 or more properties defined below. Some properties apply to the service as a whole while others apply only to the targeted resource. The targeted resource may have property values which override those for the service. For example, the timezone identifier for a particular collection may differ from the default timezone identifier for the system.
 173 Each property is an XML complex type based on the GetPropertiesBasePropertyType.

175 4.3.1 childCollection

176 Provides information about a child collections for the target.

Field	Type	#	?	Description
href	string	1	Y	The URI of the collection.
collection	CollectionType	1	Y	This is a collection
calendarCollection	CalendarCollectionType	0..1	C	If present this is a calendar collection

177 Table 6: ChildCollectionType fields

178 See resourceType for descriptions of CollectionType and Calendar CollectionType.

179 4.3.2 creationDateTime

180 This property MAY be returned for the service and SHOULD be returned for any targeted resource.

Field	Type	#	?	Description
dateTime	dateTime	1	Y	Creation dat/time of the resource

181 Table 7: CreationDateTimeType fields

182 4.3.3 displayName

183 This property SHOULD be returned for any targeted resource.

Field	Type	#	?	Description
string	string	1	Y	The displayable name.

184 Table 8: DisplayNameType fields

185 4.3.4 lastModifiedDateTime

186 This property MAY be returned for the service and SHOULD be returned for any targeted resource.

Field	Type	#	?	Description
dateTime	dateTime	1	Y	Last modified date/time of the resource

187 Table 9: LastModifiedDateTimeType fields

188 4.3.5 maxAttendeesPerInstance

189 This property SHOULD be returned for the service and MAY be returned for any targeted collection re-
190 source.

Field	Type	#	?	Description
integer	integer	1	Y	The maximum number of attendees allowed per event or task instance.

191 Table 10: MaxAttendeesPerInstanceType fields

192 4.3.6 maxDateTime

193 This property SHOULD be returned for the service and MAY be returned for any targeted collection re-
194 source.

Field	Type	#	?	Description
dateTime	dateTime	1	Y	The maximum date and time for an event.

195 Table 11: MaxDateTimeType fields

196 4.3.7 maxInstances

197 This property SHOULD be returned for the service and MAY be returned for any targeted collection re-
198 source.

Field	Type	#	?	Description
integer	integer	1	Y	The maximum number of instances for a recurring event.

199 Table 12: MaxInstancesType fields

200 4.3.8 maxResourceSize

201 This property SHOULD be returned for the service and MAY be returned for any targeted collection re-
202 source.

Field	Type	#	?	Description
integer	integer	1	Y	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.

203 Table 13: MaxResourceSizeType fields

204 4.3.9 minDateTime

205 This property SHOULD be returned for the service and MAY be returned for any targeted collection re-
206 source.

Field	Type	#	?	Description
dateTime	dateTime	1	Y	The minimum date and time for an event.

207 Table 14: MinDateTimeType fields

208 4.3.10 principalHome

209 This property SHOULD be returned for the service and MAY be returned for any targeted collection re-
210 source.

Field	Type	#	?	Description
string	string	1	Y	The home path of the currently authenticated user.

211 Table 15: PrincipalHomeType fields

212 4.3.11 resourceDescription

213 Provides some descriptive text for the targeted collection.

Field	Type	#	?	Description
string	string	1	Y	The descriptive text.

214 Table 16: ResourceDescriptionType fields

215 4.3.12 resourceOwner

216 This property SHOULD be returned for any targeted resource.

Field	Type	#	?	Description
string	string	1	Y	The principal URL of the resource owner.

217 Table 17: ResourceownerType fields

218 4.3.13 resourceTimezoneId

219 This property SHOULD be returned for the service and MAY be returned for any targeted collection re-
220 source.

Field	Type	#	?	Description
string	string	1	Y	The timezone identifier.

221 Table 18: ResourceTimezoneIdType fields

222 4.3.14 resourceType

223 Provides information about a targeted resource.

Field	Type	#	?	Description
href	string	1	Y	The URI of the collection.
collection	CollectionType	0..1	C	If present this is a collection
calendarCollection	CalendarCollectionType	0..1	C	If present this is a calendar collection
inbox	InboxType	0..1	C	If present this is a scheduling inbox
outbox	OutboxType	0..1	C	If present this is a scheduling outbox
inbox	InboxType	0..1	C	If present this is a scheduling inbox
xresource	XresourceType	0..1	C	If present provides further type information.

224 *Table 19: ResourceType fields*

225 All the child types are empty elements with the exception of XresourceType.

Field	Type	#	?	Description
string	string	1	Y	Extra information.

226 *Table 20: XresourceType fields*

227 **4.3.15 supportedCalendarComponentSet**

228 This property identifies which component types the service is prepared to store. The allowable components may be different for different targets on the same service.

Field	Type	#	?	Description
Any valid iCalendar component name	xcal:BaseComponentType	0..n	C	One or more empty iCalendar components.

230 *Table 21: SupportedCalendarComponentSetType fields*

231 **4.3.16 supportedFeatures**

232 This property SHOULD be returned for the service and MAY be returned for any targeted collection resource. The property shows what protocol features are supported by the server.

Field	Type	#	?	Description
calendarAccessFeature	CalendarAccessFeatureType	1	Y	Indicates the service supports this protocol.

234 *Table 22: SupportedFeaturesType fields*

235 **4.3.17 timezoneServer**

236 This property SHOULD be returned for the service and MAY be returned for any targeted collection resource.

Field	Type	#	?	Description
string	string	1	Y	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.

238 *Table 23: TimezoneServerType fields*

239 4.3.18 CalWS:privilege-set XML element

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

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

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

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

- 246 • CalWS: Read - current principal has read access
- 247 • CalWS: Write - current principal has write access

```
248 <calws:privilege-set>  
249 <calws:privilege><calws:read></calws:privilege>  
250 <calws:privilege><calws:write></calws:privilege>  
251 </calws:privilege-set>
```

252 4.4 Retrieving Collection and Service Properties

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

255 The service properties define the global limits and defaults. Any properties defined on collections within
256 the service hierarchy override those service defaults. The service may choose to prevent such overriding
257 of defaults and limits when appropriate. The tables below show the fields for request and response.

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

258 Table 24: GetPropertiesType fields

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request. "/" for the service.
?	GetPropertiesBasePropertyType	0..n	C	0 or more properties of the targeted resource

259 Table 25: GetPropertiesResponseType fields

260 4.4.1 Example - retrieving server properties:

```
261 >>Request  
262  
263 <?xml version="1.0" encoding="UTF-8"?>  
264 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">  
265 <SOAP-ENV:Header/>  
266 <SOAP-ENV:Body>  
267 <ns2:getProperties xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"  
268 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">  
269 <ns2:href></ns2:href>  
270 </ns2:getProperties>  
271 </SOAP-ENV:Body>  
272 </SOAP-ENV:Envelope>  
273  
274 >>Response  
275  
276 <?xml version="1.0" encoding="UTF-8"?>  
277 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">  
278 <SOAP-ENV:Header />  
279 <SOAP-ENV:Body>  
280 <ns2:getPropertiesResponse  
281 xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"  
282 xmlns:ns4="urn:ietf:params:xml:ns:icalendar-2.0"  
283 id="0" >  
284 <ns2:href></ns2:href>  
285 <ns2:lastModifiedDate>
```

```

286     <ns2:dateTime>2012-01-04T18:21:14Z</ns2:dateTime>
287   </ns2:lastModifiedDateTime>
288   <ns2:supportedCalendarComponentSet>
289     <ns4:vevent />
290     <ns4:vtodo />
291     <ns4:vavailability />
292   </ns2:supportedCalendarComponentSet>
293   <ns2:resourceType>
294     <ns2:collection />
295   </ns2:resourceType>
296   <ns2:supportedFeatures>
297     <ns2:calendarAccessFeature />
298   </ns2:supportedFeatures>
299   <ns2:maxInstances>
300     <ns2:integer>1000</ns2:integer>
301   </ns2:maxInstances>
302   <ns2:maxResourceSize>
303     <ns2:integer>100000</ns2:integer>
304   </ns2:maxResourceSize>
305   </ns2:getPropertiesResponse>
306 </SOAP-ENV:Body>
307 </SOAP-ENV:Envelope>
308
309

```

310 4.5 Creating Calendar Object Resources

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

314 The icalendar entity in the request MUST contain only a single calendaring entity with any related over-
315 rides.

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

316 *Table 26: AddItemType fields*

317 The service will respond with an AddItemResponseType giving either the href and change token of the
318 new entity or an error response.

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

319 *Table 27: AddItemResponseType additional fields*

320 4.5.1 Preconditions for Calendar Object Creation

- 321 • **CalWS:target-exists:** The entity already exists.
- 322 • **CalWS:not-calendar-data:** The resource submitted MUST be a supported media type (i.e., iCalendar)
323 for calendar object resources;
- 324 • **CalWS:invalid-calendar-data:** The resource submitted MUST be valid data for the media type being
325 specified (i.e., MUST contain valid iCalendar data);
- 326 • **CalWS:invalid-calendar-object-resource:** The resource submitted in the request MUST obey all re-
327 strictions specified in Calendar Object Resources (e.g., calendar object resources MUST NOT con-
328 tain more than one type of calendar component, calendar object resources MUST NOT specify the
329 iCalendar METHOD property, etc.);
- 330 • **CalWS:unsupported-calendar-component:** The resource submitted in the request MUST contain a
331 type of calendar component that is supported in the targeted calendar collection;

- 332 • **CalWS:uid-conflict:** The resource submitted in the request MUST NOT specify an iCalendar UID
333 property value already in use in the targeted calendar collection or overwrite an existing calendar
334 object resource with one that has a different UID property value. Servers SHOULD report the URL
335 of the resource that is already making use of the same UID property value in the CalWS:href ele-
336 ment
337 <!ELEMENT uid-conflict (CalWS:href)>
- 338 • **CalWS:exceeds-max-resource-size:** The resource submitted in the request MUST have an octet size
339 less than or equal to the value of the CalDAV:max-resource-size property value on the calendar col-
340 lection where the resource will be stored;
- 341 • **CalWS:before-min-date-time:** The resource submitted in the request MUST have all of its iCalendar
342 DATE or DATE-TIME property values (for each recurring instance) greater than or equal to the
343 value of the CalDAV:min-date-time property value on the calendar collection where the resource
344 will be stored;
- 345 • **CalWS:after-max-date-time:** The resource submitted in the request MUST have all of its iCalendar
346 DATE or DATE-TIME property values (for each recurring instance) less than the value of the Cal-
347 DAV:max-date-time property value on the calendar collection where the resource will be stored;
- 348 • **CalWS:too-many-instances:** The resource submitted in the request MUST generate a number of re-
349 curring instances less than or equal to the value of the CalDAV: max-instances property value on
350 the calendar collection where the resource will be stored;
- 351 • **CalWS:too-many-attendees-per-instance:** The resource submitted in the request MUST have a
352 number of ATTENDEE properties on any one instance less than or equal to the value of the Cal-
353 DAV:max-attendees-per-instance property value on the calendar collection where the resource will
354 be stored;

355 4.5.2 Example - successful addItem:

```

356 >>Request
357
358 <?xml version="1.0" encoding="UTF-8"?>
359 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
360 <SOAP-ENV:Header/>
361 <SOAP-ENV:Body>
362 <ns2:addItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
363 <ns3:icalendar xmlns:ns3="urn:iETF:params:xml:ns:icalendar-2.0">
364 <ns2:href>/user/douglm/calendar</ns2:href>
365 <ns3:icalendar>
366 <ns3:vcalendar>
367 <ns3:components>
368 <ns3:vevent>
369 <ns3:properties>
370 <ns3:uid>
371 <ns3:text>1302064354993</ns3:text>
372 </ns3:uid>
373 <ns3:summary>
374 <ns3:text>try this</ns3:text>
375 </ns3:summary>
376 <ns3:dtstart>
377 <ns3:date-time>20110406T150000Z</ns3:date-time>
378 </ns3:dtstart>
379 <ns3:dtend>
380 <ns3:date-time>20110406T160000Z</ns3:date-time>
381 </ns3:dtend>
382 </ns3:properties>
383 </ns3:vevent>
384 </ns3:components>
385 </ns3:vcalendar>
386 </ns3:icalendar>
387 </ns2:addItem>
388 </SOAP-ENV:Body>
389 </SOAP-ENV:Envelope>
390
391 >>Response
392
393 <?xml version="1.0" encoding="UTF-8"?>
394 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

```

```

395 <SOAP-ENV:Header/>
396 <SOAP-ENV:Body>
397   <ns2:addItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
398     xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
399     <ns2:status>OK</ns2:status>
400     <ns2:href>/user/douglm/calendar/1302064354993.ics</ns2:href>
401     <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
402   </ns2:addItemResponse>
403 </SOAP-ENV:Body>
404 </SOAP-ENV:Envelope>

```

405 4.6 Retrieving resources

406 Fetching calendar object resources is carried out by using a CalWS-SOAP fetchItem request with an href
 407 specifying the entity to be fetched. The response will contain the calendaring entity with any related over-
 408 rides.

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

409 *Table 28: FetchItemType fields*

410 The service will respond with a FetchItemResponseType containing either the change token, its href and
 411 the entity or an error response.

Field	Type	#	?	Description
changeToken	string	0..1	N	The change token for the fetched entity
href	string	1	Y	Identify the entity.
icalendar	xcal:IcalendarType	0..1	N	The fetched entity

412 *Table 29: FetchItemResponseType additional fields*

413 4.6.1 Example - successful fetchItem:

```

414 >>Request
415
416 <?xml version="1.0" encoding="UTF-8"?>
417 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
418   <SOAP-ENV:Header/>
419   <SOAP-ENV:Body>
420     <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
421       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
422       <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
423     </ns2:fetchItem>
424   </SOAP-ENV:Body>
425 </SOAP-ENV:Envelope>
426
427 >>Response
428
429 <?xml version="1.0" encoding="UTF-8"?>
430 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
431   <SOAP-ENV:Header/>
432   <SOAP-ENV:Body>
433     <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
434       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
435       <ns2:status>OK</ns2:status>
436       <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
437       <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
438       <ns3:icalendar>
439         <ns3:vcalendar>
440           <ns3:properties>
441             <ns3:prodid>
442               <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
443             </ns3:prodid>
444             <ns3:version>
445               <ns3:text>2.0</ns3:text>
446             </ns3:version>

```

```

447     </ns3:properties>
448     <ns3:components>
449       <ns3:vevent>
450         <ns3:properties>
451           <ns3:created>
452             <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
453           </ns3:created>
454           <ns3:dtend>
455             <ns3:date-time>20110406T160000Z</ns3:date-time>
456           </ns3:dtend>
457           <ns3:dtstamp>
458             <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
459           </ns3:dtstamp>
460           <ns3:dtstart>
461             <ns3:date-time>20110406T150000Z</ns3:date-time>
462           </ns3:dtstart>
463           <ns3:last-modified>
464             <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
465           </ns3:last-modified>
466           <ns3:summary>
467             <ns3:text>try this</ns3:text>
468           </ns3:summary>
469           <ns3:uid>
470             <ns3:text>1302105461170</ns3:text>
471           </ns3:uid>
472         </ns3:properties>
473       </ns3:vevent>
474     </ns3:components>
475   </ns3:vcalendar>
476 </ns3:icalendar>
477 </ns2:fetchItemResponse>
478 </SOAP-ENV:Body>
479 </SOAP-ENV:Envelope>

```

480 4.6.2 Example - unsuccessful fetchItem:

```

481 >>Request
482
483 <?xml version="1.0" encoding="UTF-8"?>
484 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
485   <SOAP-ENV:Header/>
486   <SOAP-ENV:Body>
487     <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
488       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
489       <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
490     </ns2:fetchItem>
491   </SOAP-ENV:Body>
492 </SOAP-ENV:Envelope>
493
494 >>Response
495
496 <?xml version="1.0" encoding="UTF-8"?>
497 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
498   <SOAP-ENV:Header/>
499   <SOAP-ENV:Body>
500     <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
501       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
502       <ns2:status>Error</ns2:status>
503       <ns2:errorResponse>
504         <ns2:targetDoesNotExist/>
505       </ns2:errorResponse>
506     </ns2:fetchItemResponse>
507   </SOAP-ENV:Body>
508 </SOAP-ENV:Envelope>

```

509 4.7 Updating resources

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

- 511 • An iCalendar element contains...
- 512 • a single vCalendar element which contains...

- 513 • one or more calendaring components, event, task etc each of which contain...
- 514 • zero or more components, alarms etc or one or more properties each of which contains...
- 515 • zero or more parameters and one or more values.

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

```

517     <icalendar>
518         <vcalendar>
519             <components>
520                 <vevent>
521                     <properties>
522                         <uid>
523                             <text>1302064354993-a</text>
524                         </uid>
525                         <summary>
526                             <text>try this</text>
527                         </summary>
528                         <dtstart>
529                             <date-time>2011-07-18T15:00:00Z</date-time>
530                         </dtstart>
531                         <dtend>
532                             <date-time>2011-07-18T16:00:00Z</date-time>
533                         </dtend>
534                     </properties>
535                 </vevent>
536             </components>
537         </vcalendar>
538     </icalendar>

```

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

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

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

- 547 • Remove - components, properties or parameters
- 548 • Add - components, properties or parameters
- 549 • Change - property or parameter values

550 Removals MUST be processed ahead of additions

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

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request.
changeToken	string	1	Y	The change token held by the client for that entity
select	ComponentSelection-Type	1..*	Y	Must select vcalendar

555 *Table 30: UpdateItem Type fields*

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

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

559 *Table 31: ComponentsSelectionType fields*

560 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
remove	PropertyReferenceType	0..1	N	Supplies properties to remove
add	PropertyReferenceType	0..1	N	Species properties to add

561 *Table 32: PropertiesSelectionType fields*

562 To complete that pattern there is also a ParametersSelectionType used to select property parameters for
563 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
remove	ParameterReference-Type	0..1	N	Supplies parameters to remove
add	ParameterReference-Type	0..1	N	Species parameters to add

564 *Table 33: ParametersSelectionType fields*

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

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

567 *Table 34: 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.

568 *Table 35: PropertyReferenceType fields*

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

569 *Table 36: ParameterReferenceType fields*

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

573 ComponentSelectionType contains three child elements. The first is any valid icalendar component ele-
574 ment which is to be matched at the current level.

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

578 The optional components selection allows selection and possible updates to the nested icalendar compo-
579 nents element of the component. An iCalendar components element cannot take a value so the only up-
580 dates possible are addition and removal of components. Nested components may be selected for up-
581 dates.

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

582 *Table 37: ComponentSelectionType fields*

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

585 The optional parameters selection allows selection and possible updates to the parameters of the prop-
586 erty.

587 The optional change element allows a change to the value of the property. The new value is specified by
588 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	ParametersSelection-Type	0..1	N	To match the parameters element
change	PropertyReferenceType	0..1	N	To provide a new value

589 *Table 38: PropertySelectionType fields*

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

592 The optional change element allows a change to the value of the parameter. The new value is specified
593 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	ParameterReference-Type	0..1	N	To provide a new value

594 Table 39: ParameterSelectionType fields

595 For a successful update the service will respond with a UpdateItemResponseType containing the status
596 and the new change token.

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

597 Table 40: UpdateItemResponseType additional fields

598 The change token value should be used to replace the value held by the client.

599 4.7.1 Change tokens and concurrent updates

600 The change token is used to allow a service to determine whether or not it is safe to carry out an update
601 requested by the client. The change token should be opaque to the client but will probably in fact be a
602 structured value. Calendaring transactions have some special characteristics which make it desirable to
603 allow certain non-conflicting updates to take place while other changes are taking place. For example,
604 meeting requests with a large number of attendees can be frequently updated by the server as a result of
605 attendee participation status changes. If we use an unstructured change token to represent all changes
606 this can make it very difficult to update an event while those participation status changes are being made.
607 If, on the other hand, the token has a section indicating that only participation status changes have been
608 made, then other changes can take place. For a reference on implementing such a token see "Avoiding
609 Conflicts when Updating Scheduling Object Resources" in [RFC 6638]. This describes the use of a
610 schedule-tag.

611 4.7.2 Example - successful update:

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

```
613 <ns3:icalendar>
614   <ns3:vcalendar>
615     <ns3:components>
616       <ns3:vevent>
617         <ns3:properties>
618           <ns3:uid>
619             <ns3:text>1302064354993-a</ns3:text>
620           </ns3:uid>
621           <ns3:summary>
622             <ns3:text>try this</ns3:text>
623           </ns3:summary>
624           <ns3:dtstart>
625             <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
626           </ns3:dtstart>
627           <ns3:dtend>
628             <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
629           </ns3:dtend>
630         </ns3:properties>
631       </ns3:vevent>
632     </ns3:components>
633   </ns3:vcalendar>
634 </ns3:icalendar>
```

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

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

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

```
641 >>Request
```

```

642
643 <?xml version="1.0" encoding="UTF-8"?>
644 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
645   <SOAP-ENV:Header/>
646   <SOAP-ENV:Body>
647     <ns2:updateItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
648       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
649       <ns2:href>/user/douglm/calendar/1302064354993-a.ics</ns2:href>
650       <ns2:changeToken>"20110802T032608Z-0" null</ns2:changeToken>
651       <ns2:select>
652         <ns3:vcalendar/>
653         <ns2:components>
654           <ns2:component>
655             <ns3:vevent>
656               <ns3:properties>
657                 <ns3:uid>
658                   <ns3:text>1302064354993-a</ns3:text>
659                 </ns3:uid>
660               </ns3:properties>
661             </ns3:vevent>
662           <ns2:properties>
663             <ns2:property>
664               <ns3:dtstart>
665                 <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
666               </ns3:dtstart>
667             <ns2:parameters>
668               <ns2:add>
669                 <ns3:tzid>
670                   <ns3:text>America/New_York</ns3:text>
671                 </ns3:tzid>
672               </ns2:add>
673             </ns2:parameters>
674             <ns2:change>
675               <ns3:dtstart>
676                 <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
677               </ns3:dtstart>
678             </ns2:change>
679           </ns2:property>
680           <ns2:property>
681             <ns3:summary>
682               <ns3:text>try this</ns3:text>
683             </ns3:summary>
684             <ns2:change>
685               <ns3:summary>
686                 <ns3:text>A changed summary - again and again and again</ns3:text>
687               </ns3:summary>
688             </ns2:change>
689           </ns2:property>
690           <ns2:add>
691             <ns3:categories>
692               <ns3:text>newcategory-2</ns3:text>
693               <ns3:text>resources</ns3:text>
694               <ns3:text>paper</ns3:text>
695             </ns3:categories>
696           </ns2:add>
697         </ns2:properties>
698       </ns2:component>
699     </ns2:components>
700   </ns2:select>
701 </ns2:updateItem>
702 </SOAP-ENV:Body>
703 </SOAP-ENV:Envelope>
704
705 >>Response
706
707 <?xml version="1.0" encoding="UTF-8"?>
708 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
709   <SOAP-ENV:Header/>
710   <SOAP-ENV:Body>
711     <ns2:updateItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
712       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"

```

```

713         id="0">
714     <ns2:status>OK</ns2:status>
715 </ns2:updateItemResponse>
716 </SOAP-ENV:Body>
717 </SOAP-ENV:Envelope>

```

4.7.3 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

```

729     <ns2:properties>
730     <ns2:property>
731         <ns3:dtstart>
732             <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
733         </ns3:dtstart>
734         <ns2:parameters>
735             <ns2:add>
736                 <ns3:tzid>
737                     <ns3:text>America/New_York</ns3:text>
738                 </ns3:tzid>
739             </ns2:add>
740         </ns2:parameters>
741     </ns2:change>
742     <ns3:dtstart>
743         <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
744     </ns3:dtstart>
745 </ns2:change>
746 </ns2:property>
747 </ns2:properties>

```

In this example we add two categories to the event.

```

749     <ns2:properties>
750     <ns2:add>
751         <ns3:categories>
752             <ns3:text>paper</ns3:text>
753         </ns3:categories>
754     </ns2:add>
755     <ns2:add>
756         <ns3:categories>
757             <ns3:text>resources</ns3:text>
758         </ns3:categories>
759     </ns2:add>
760 </ns2:properties>

```

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

```

762     <ns2:properties>
763     <ns2:remove>
764         <ns3:dtend>
765             <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
766         </ns3:dtend>
767     </ns2:remove>
768     <ns2:add>
769         <ns3:duration>
770             <ns3:duration>PT1H</ns3:duration>
771         </ns3:duration>
772     </ns2:add>
773 </ns2:properties>

```

In this example we change the dtstart timezone identifier.

```

775     <ns2:properties>
776     <ns2:property>
777         <ns3:dtstart>

```

```

778     <ns3:parameters>
779         <ns3:tzid>
780             <ns3:text>America/New_York</ns3:text>
781         </ns3:tzid>
782     </ns3:parameters>
783     <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
784 </ns3:dtstart>
785 <ns2:parameters>
786     <ns2:parameter>
787         <ns3:tzid>
788             <ns3:text>America/New_York</ns3:text>
789         </ns3:tzid>
790         <ns2:change>
791             <ns3:tzid>
792                 <ns3:text>America/Montreal</ns3:text>
793             </ns3:tzid>
794         </ns2:change>
795     </ns2:parameter>
796 </ns2:parameters>
797 </ns2:property>
798 </ns2:properties>
799

```

800 4.7.4 Creating an update message.

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

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

810 These properties are

- 811 • categories
- 812 • exdate
- 813 • freebusy
- 814 • rdate

815 This normalization can take place before comparison.

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

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

821 4.8 Deletion of resources

822 Deletion of calendar object resources is carried out by using a CalWS-SOAP deleteItem request with an
823 href specifying the entity to be deleted. The deleteItem request is not valid when the href specifies a col-
824 lection.

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

825 *Table 41: DeleteItem fields*

826 The service will respond with a DeleteItemResponseType containing the status and a possible error re-
827 sponse. There are no additional elements.

828 4.8.1 Example - successful deleteItem:

829 >>Request

830

```

831 <?xml version="1.0" encoding="UTF-8"?>
832 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
833   <SOAP-ENV:Header/>
834   <SOAP-ENV:Body>
835     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
836                   xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
837       <ns2:href>/user/douglm/calendar/1302620814655.ics</ns2:href>
838     </ns2:deleteItem>
839   </SOAP-ENV:Body>
840 </SOAP-ENV:Envelope>
841
842 >>Response
843
844 <?xml version="1.0" encoding="UTF-8"?>
845 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
846   <SOAP-ENV:Header/>
847   <SOAP-ENV:Body>
848     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
849                            xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
850       <ns2:status>OK</ns2:status>
851     </ns2:deleteItemResponse>
852   </SOAP-ENV:Body>
853 </SOAP-ENV:Envelope>

```

854 4.8.2 Example - unsuccessful deleteItem:

```

855 >>Request
856
857 <?xml version="1.0" encoding="UTF-8"?>
858 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
859   <SOAP-ENV:Header/>
860   <SOAP-ENV:Body>
861     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
862                   xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
863       <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
864     </ns2:deleteItem>
865   </SOAP-ENV:Body>
866 </SOAP-ENV:Envelope>
867
868 >>Response
869
870 <?xml version="1.0" encoding="UTF-8"?>
871 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
872   <SOAP-ENV:Header/>
873   <SOAP-ENV:Body>
874     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
875                            xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
876       <ns2:status>Error</ns2:status>
877       <ns2:errorResponse>
878         <ns2:targetDoesNotExist/>
879       </ns2:errorResponse>
880     </ns2:deleteItemResponse>
881   </SOAP-ENV:Body>
882 </SOAP-ENV:Envelope>

```

883 4.9 Querying calendar resources

884 Querying provides a mechanism by which information can be obtained from the service through possibly
885 complex queries. A skeleton icalendar entity can be provided to limit the amount of information returned to
886 the client. A query takes the parts

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

890 4.9.1 Calendar Query common types

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

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

894 *Table 42: UTCTimeRangeType elements*

895 The TextMatchType is used to match text values in properties and parameters. The collation attribute
896 species a collation as defined in [RFC4790].

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

899 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 [RFC4790]. "
#negate-condition	boolean	0..1	N	if "true" negates the condition

900 *Table 43: TextMatchType attributes*

901 4.9.2 CompFilterType

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

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

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

Field	Type	#	?	Description
anyComp	AnyCompType	0..1	C	One of anyComp, vcalendar or a BaseC- componentType must be supplied. anyComp indicates that any component will match.
xcal:vcalendar	xcal:VcalendarType	0..1	C	Matches vcalendar at the top level. Must be provided
xcal:baseComponent	xcal:BaseComponentType	0..1	C	May be vevent or vtodo for example.
#test	String	0..1	N	"anyof" is a logical OR of the child ele- ments. "allof" is a logical AND of the child ele- ments.
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

908 *Table 44: CompFilterType elements*

909 4.9.3 PropFilterType

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

Field	Type	#	?	Description
xcal:baseProperty	xcal:BasePropertyType	1	Y	Specifies the property to be matched.
#test	String	0..1	N	"anyof" is a logical OR of the child elements. "allof" is a logical AND of the child elements.
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

912 Table 45: PropFilterType elements

913 4.9.4 ParamFilterType

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

Field	Type	#	?	Description
xcal:baseParameter	xcal:BaseParameterType	1	Y	Specifies the parameter to be matched.
is-not-defined	empty	0..1	N	Only this element or text-match
text-match	TextMatchtype	0..1	N	

915 Table 46: ParamFilterType elements

916 **4.9.5 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
xcal:icalendar	xcal: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

917 *Table 47: CalendarQueryType elements*

918 **4.9.6 Specifying data to be returned**

919 This is achieved by specifying one of the following

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

925 **4.9.7 Pre/postconditions for calendar queries**

926 The preconditions as defined in [RFC 4791] Section 7.8 apply here. CalWS errors may be reported by the

927 service when preconditions or postconditions are violated.

928 4.9.8 Time range limited queries.

929 Time-range limited retrieval has some special characteristics. The simplest case is a single event or task
930 which overlaps the requested time-period. Recurring items and other components such as alarms compli-
931 cate the picture.

932 4.9.9 Example: time range limited retrieval

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

```
935 >> Request <<
936
937 <?xml version="1.0" encoding="UTF-8"?>
938 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
939   <SOAP-ENV:Header/>
940   <SOAP-ENV:Body>
941     <ns2:calendarQuery xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
942                       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
943       <ns2:href>/user/douglm/calendar</ns2:href>
944       <ns3:icalendar>
945         <ns3:vcalendar>
946           <ns3:components>
947             <ns3:vevent>
948               <ns3:properties>
949                 <ns3:summary/>
950                 <ns3:dtstart/>
951                 <ns3:dtend/>
952                 <ns3:duration/>
953                 <ns3:uid/>
954                 <ns3:recurrence-id/>
955                 <ns3:rrule/>
956                 <ns3:rdate/>
957                 <ns3:exdate/>
958               </ns3:properties>
959             </ns3:vevent>
960           </ns3:components>
961         </ns3:vcalendar>
962       </ns3:icalendar>
963       <ns2:filter>
964         <ns2:compFilter test="anyof">
965           <ns3:vcalendar />
966           <ns2:compFilter>
967             <ns3:vevent />
968             <ns2:time-range end="20110430T040000Z" start="20110401T040000Z"/>
969           </ns2:compFilter>
970         </ns2:filter>
971       </ns2:calendarQuery>
972     </SOAP-ENV:Body>
973   </SOAP-ENV:Envelope>
974
975 >> Response <<
976
977 <?xml version="1.0" encoding="UTF-8"?>
978 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
979   <SOAP-ENV:Header/>
980   <SOAP-ENV:Body>
981     <ns2:calendarQueryResponse
982                       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
983                       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
984       <ns2:status>OK</ns2:status>
985       <ns2:response>
986         <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
987         <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
988         <ns2:propstat>
989           <ns2:prop>
990             <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
991               <ns3:icalendar>
992                 <ns3:vcalendar>
993                   <ns3:properties>
994                     <ns3:prodid>
```

```

995         <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
996     </ns3:prodid>
997     <ns3:version>
998         <ns3:text>2.0</ns3:text>
999     </ns3:version>
1000 </ns3:properties>
1001 <ns3:components>
1002     <ns3:vevent>
1003         <ns3:properties>
1004             <ns3:dtend>
1005                 <ns3:date-time>20110406T160000Z</ns3:date-time>
1006             </ns3:dtend>
1007             <ns3:dtstart>
1008                 <ns3:date-time>20110406T150000Z</ns3:date-time>
1009             </ns3:dtstart>
1010             <ns3:summary>
1011                 <ns3:text>try this</ns3:text>
1012             </ns3:summary>
1013             <ns3:uid>
1014                 <ns3:text>1302105461170</ns3:text>
1015             </ns3:uid>
1016         </ns3:properties>
1017     </ns3:vevent>
1018 </ns3:components>
1019 </ns3:vcalendar>
1020 </ns3:icalendar>
1021 </ns2:calendar-data>
1022 </ns2:prop>
1023 <ns2:status>OK</ns2:status>
1024 </ns2:propstat>
1025 </ns2:response>
1026 <ns2:response>
1027
1028 <ns2:href>/user/douglm/calendar/CAL-00f1fc61-2f021bca-012f-022947f8-00000006.ics</ns2:hr
1029 ef>
1030     <ns2:changeToken>"20110405T140920Z-0"</ns2:changeToken>
1031 <ns2:propstat>
1032     <ns2:prop>
1033         <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1034             <ns3:icalendar>
1035                 <ns3:vcalendar>
1036                     <ns3:properties>
1037                         <ns3:prodid>
1038                             <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
1039                         </ns3:prodid>
1040                         <ns3:version>
1041                             <ns3:text>2.0</ns3:text>
1042                         </ns3:version>
1043                     </ns3:properties>
1044                     <ns3:components>
1045                         <ns3:vevent>
1046                             <ns3:properties>
1047                                 <ns3:duration>
1048                                     <ns3:duration>PT1H</ns3:duration>
1049                                 </ns3:duration>
1050                                 <ns3:dtstart>
1051                                     <ns3:parameters>
1052                                         <ns3:tzid>
1053                                             <ns3:text>America/New_York</ns3:text>
1054                                         </ns3:tzid>
1055                                         </ns3:parameters>
1056                                         <ns3:date-time>20110412T110000</ns3:date-time>
1057                                     </ns3:dtstart>
1058                                 <ns3:summary>
1059                                     <ns3:text>Test recurring event</ns3:text>
1060                                 </ns3:summary>
1061                                 <ns3:uid>
1062                                     <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-00000006demobedework@mysite.edu</ns3:text>
1063                                 </ns3:uid>
1064                                 <ns3:rrule>

```

```

1066         <ns3:recur>
1067             <ns3:freq>WEEKLY</ns3:freq>
1068             <ns3:count>2</ns3:count>
1069             <ns3:interval>1</ns3:interval>
1070         </ns3:recur>
1071     </ns3:rrule>
1072 </ns3:properties>
1073 </ns3:vevent>
1074 <ns3:vevent>
1075     <ns3:properties>
1076         <ns3:recurrence-id>
1077             <ns3:parameters>
1078                 <ns3:tzid>
1079                     <ns3:text>America/New_York</ns3:text>
1080                 </ns3:tzid>
1081             </ns3:parameters>
1082             <ns3:date-time>20110419T150000Z</ns3:date-time>
1083         </ns3:recurrence-id>
1084         <ns3:duration>
1085             <ns3:duration>PT1H</ns3:duration>
1086         </ns3:duration>
1087         <ns3:dtstart>
1088             <ns3:parameters>
1089                 <ns3:tzid>
1090                     <ns3:text>America/New_York</ns3:text>
1091                 </ns3:tzid>
1092             </ns3:parameters>
1093             <ns3:date-time>20110419T120000</ns3:date-time>
1094         </ns3:dtstart>
1095         <ns3:summary>
1096             <ns3:text>Test recurring event</ns3:text>
1097         </ns3:summary>
1098         <ns3:uid>
1099     <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-00000006demobedework@mysite.edu</ns3:text>
1100     </ns3:uid>
1101 </ns3:properties>
1102 </ns3:vevent>
1103 </ns3:components>
1104 </ns3:vcalendar>
1105 </ns3:icalendar>
1106 </ns2:calendar-data>
1107 </ns2:prop>
1108 <ns2:status>OK</ns2:status>
1109 </ns2:propstat>
1110 </ns2:response>
1111 </ns2:calendarQueryResponse>
1112 </SOAP-ENV:Body>
1113 </SOAP-ENV:Envelope>
1114
1115

```

1116 4.10 Free-busy queries

1117 Freebusy queries are used to obtain freebusy information for a principal. The result contains information
1118 only for events to which the current principal has sufficient access and may be affected by components
1119 and rules available only to the server (for instance office hours availability).

1120 These queries are carried out by using a CalWS-SOAP freebusyReport request with an href specifying a
1121 principal. The freebusyReport request is not valid when the href specifies any entity other than a principal.
1122 The query follows the specification defined in [FreeBusy Read URL] with certain limitations. As an authen-
1123 ticated user to the CalWS service scheduling read-freebusy privileges must have been granted. As an
1124 unauthenticated user equivalent access must have been granted to unauthenticated users.

1125 Freebusy information is returned by default as xcalendar vfreebusy components, as defined by [RFC
1126 6321]. Such a component is not meant to conform to the requirements of VFREEBUSY components in
1127 RFC 5546. The VFREEBUSY component SHOULD conform to section "4.6.4 Free/Busy Component" of
1128 [RFC 5545]. A client SHOULD ignore the ORGANIZER field.

1129 Since a Freebusy query can only refer to a single user, a client will already know how to match the result
1130 component to a user. A server MUST only return a single vfreebusy component.

1131 4.10.1 Element values

1132 Three values are provided: href; start; end. Only the href is required. The start and end are in XML UTC
1133 date/time format and are interpreted as follows:

1134 4.10.1.1 start

1135 **Default:** If omitted the default value is left up to the server. It may be the current day, start of the cur-
1136 rent month, etc.

1137 **Description:** Specifies the start date for the Freebusy data. The server is free to ignore this value
1138 and return data in any time range. The client must check the data for the returned time range.

1139 **Format:** An XML UTC date-time

1140 **Example:**

```
1141 2011-12-01T10:15:00Z
```

1142 **Notes:** Specifying only a start date/time without specifying an end-date/time or period should be inter-
1143 preted as in [RFC 5545]. The effective period should cover the remainder of that day.

1144 4.10.1.2 end

1145 **Default:** Same as start

1146 **Description:** Specifies the end date for the Freebusy data. The server is free to ignore this value.

1147 **Format:** Same as start

1148 **Example:** Same as start

1149 The server is free to ignore the start, end and period parameters. It is recommended that the server return
1150 at least 6 weeks of data from the current day.

1151 A client MUST check the time range in the response as a server may return a different time range than
1152 the requested range.

1153 4.10.2 Examples

1154 The following is an unsuccessful request targeting an invalid resource.

```
1155 >> Request <<
1156
1157 <?xml version="1.0" encoding="UTF-8"?>
1158 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1159   <SOAP-ENV:Header/>
1160   <SOAP-ENV:Body>
1161     <ns2:freebusyReport
1162       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1163       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1164       <ns2:href>/user/douglm/calendar</ns2:href>
1165       <ns2:time-range>
1166         <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1167         <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1168       </ns2:time-range>
1169     </ns2:freebusyReport>
1170   </SOAP-ENV:Body>
1171 </SOAP-ENV:Envelope>
1172
1173 >> Response <<
1174
1175 <?xml version="1.0" encoding="UTF-8"?>
1176 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1177   <SOAP-ENV:Header/>
1178   <SOAP-ENV:Body>
1179     <ns2:freebusyReportResponse
1180       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1181       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1182       <ns2:status>Error</ns2:status>
1183       <ns2:message>Only principal href supported</ns2:message>
1184     </ns2:freebusyReportResponse>
1185   </SOAP-ENV:Body>
1186 </SOAP-ENV:Envelope>
```

1187 The following is an example of a request to retrieve Freebusy data for a user:

```
1188 >> Request <<
1189
1190 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

```

1191 <SOAP-ENV:Header/>
1192 <SOAP-ENV:Body>
1193   <ns2:freebusyReport
1194     xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1195     xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1196     <ns2:href>/principals/users/douglm</ns2:href>
1197     <ns2:time-range>
1198       <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1199       <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1200     </ns2:time-range>
1201   </ns2:freebusyReport>
1202 </SOAP-ENV:Body>
1203 </SOAP-ENV:Envelope>
1204
1205 >> Response <<
1206
1207 <?xml version="1.0" encoding="UTF-8"?>
1208 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1209   <SOAP-ENV:Header/>
1210   <SOAP-ENV:Body>
1211     <ns2:freebusyReportResponse
1212       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1213       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1214     <ns2:status>OK</ns2:status>
1215     <ns3:icalendar>
1216       <ns3:vcalendar>
1217         <ns3:properties>
1218           <ns3:prodid>
1219             <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1220           </ns3:prodid>
1221           <ns3:version>
1222             <ns3:text>2.0</ns3:text>
1223           </ns3:version>
1224         </ns3:properties>
1225         <ns3:components>
1226           <ns3:vfreebusy>
1227             <ns3:properties>
1228               <ns3:attendee>
1229                 <ns3:parameters>
1230                   <ns3:partstat>
1231                     <ns3:text>NEEDS-ACTION</ns3:text>
1232                   </ns3:partstat>
1233                 </ns3:parameters>
1234                 <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1235               </ns3:attendee>
1236               <ns3:created>
1237                 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1238               </ns3:created>
1239               <ns3:dtend>
1240                 <ns3:date-time>2011-04-30T00:00:00Z</ns3:date-time>
1241               </ns3:dtend>
1242               <ns3:dtstamp>
1243                 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1244               </ns3:dtstamp>
1245               <ns3:dtstart>
1246                 <ns3:date-time>2011-04-01T00:00:00Z</ns3:date-time>
1247               </ns3:dtstart>
1248               <ns3:freebusy>
1249                 <ns3:parameters>
1250                   <ns3:fbtype>
1251                     <ns3:text>BUSY</ns3:text>
1252                   </ns3:fbtype>
1253                 </ns3:parameters>
1254                 <ns3:period>
1255                   <ns3:start>2011-04-06T15:00:00Z</ns3:start>
1256                   <ns3:end>2011-04-06T16:00:00Z</ns3:end>
1257                 </ns3:period>
1258               </ns3:freebusy>
1259               <ns3:last-modified>
1260                 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1261               </ns3:last-modified>

```

```

1262         <ns3:organizer>
1263             <ns3:parameters/>
1264             <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1265         </ns3:organizer>
1266         <ns3:uid>
1267             <ns3:text>2UTDVPZ9H0EQL9QISI44SP5IFPC4N75</ns3:text>
1268         </ns3:uid>
1269         </ns3:properties>
1270     </ns3:vfreebusy>
1271 </ns3:components>
1272 </ns3:vcalendar>
1273 </ns3:icalendar>
1274 </ns2:FreebusyReportResponse>
1275 </SOAP-ENV:Body>
1276 </SOAP-ENV:Envelope>

```

1277

1278 4.11 Multiple operations

1279 Each of the previously described operations acts upon a single entity or resource only. Frequently we
1280 have the need to update an interconnected set of entities so that we maintain the consistency of the struc-
1281 ture. This requires an atomic operation which can successfully update all the entities or roll back the oper-
1282 ation on failure.

1283 The MultiOpType operation provides such a feature. It is essentially a wrapper around any of the other
1284 operations which guarantees the success of the entire set or a roll back. Using the id attribute for re-
1285 quests, each individual response can be located in the result.

1286 The MultiOpType request takes the following elements

Field	Type	#	?	Description
operations	Sequence of BaseOpera- tionType	1	Y	Contains one or more operations

1287 Table 48: MultiOpType elements

1288 The response type is also simple containing a single element containing all the responses.

Field	Type	#	?	Description
responses	Sequence of BaseRespon- seType	1	Y	Contains zero or more responses

1289 Table 49: MultiOpResponseType elements

1290

1291

1292

1293

1294 5 Conformance

1295 Certain calendaring properties and components are interrelated and it is necessary to have knowledge of
1296 all these properties and their current values to allow consistent update and understanding of a target com-
1297 ponent. The normative definition for these relationships is RFC5445, RFC5446 and related RFCs.
1298 However, those specifications assume a complete view of entities being fetched or updated. This specifi-
1299 cation allows updates of entities when only a partial view is available. In fact it is the very nature of SOAP
1300 based transaction to provide such a partial view. Given that, parties attempting to update entities **MUST**
1301 have sufficient information to ensure the end result is consistent. Services allowing updates to entities
1302 **MUST** ensure that the result after an update operation is still internally consistent.

1303 5.1 Start, end and duration in calendar components

1304 A period of time is fully specified by a start and an end or duration.

1305 5.1.1 Updating, transporting and maintaining start, and and duration.

- 1306 • For all components the calculated or specified start must be at or before the end.
- 1307 • When a system updates or stores a calendar component it **MUST** retain the relationship of start, end
1308 and duration. Applications **MUST NOT** without good cause, change a start and end pair into a start
1309 and duration nor the reverse. Semantically they are not equivalent when DST transitions occur dur-
1310 ing the time of the event.
- 1311 • For interoperability, iCalendar based systems **SHOULD** avoid the use of weekly durations and XML
1312 based systems **SHOULD** avoid the use of yearly durations.

1313 5.1.2 VEVENT:

- 1314 • The three properties are DTSTART, DTEND and DURATION.
- 1315 • DTSTART **MUST** appear once and only one of DTEND or DURATION **MAY** be present.
- 1316 • The DTSTART property for a VEVENT specifies the inclusive start of the event. For recurring events, it
1317 also specifies the very first instance in the recurrence set.
- 1318 • The DTEND property for a VEVENT calendar component specifies the non-inclusive end of the event.
- 1319 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE value
1320 type but no DTEND nor DURATION property, the event's duration is taken to be one day.
- 1321 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE-TIME
1322 value type but no DTEND nor DURATION property, the event ends on the same calendar date and
1323 time of day specified by the DTSTART property, that is, it signifies a zero length instant in time.

1324 5.1.3 VTODO:

- 1325 • The three properties are DTSTART, DUE, DURATION.
- 1326 • DTSTART **MAY** appear once.
- 1327 • Either DUE or DURATION **MAY** appear in a VTODO, but DUE and DURATION **MUST NOT** occur in
1328 the same VTODO.
- 1329 • If DURATION does appear in a VTODO, then DTSTART **MUST** also appear in the same VTODO.
- 1330 • The three properties for a VTODO are related in the same way as for VEVENT. Additionally a VTODO
1331 calendar component without the DTSTART and DUE (or DURATION) properties specifies a
1332 VTODO that will be associated with each successive calendar date, until it is completed.

1333 5.1.4 VJOURNAL:

- 1334 • DTSTART only, which may be a date or date-time value.

1335 **5.1.5 VAVAILABILITY**

- 1336 • DTSTART and DTEND if specified MUST be date-time values.
- 1337 • DTSTART MAY appear once and signifies start of the busy period.
- 1338 • Only one of DTEND or DURATION MAY appear and signify the end of the busy period.
- 1339 • If DURATION does appear in a VAVAILABILITY, then DTSTART MUST also appear in the same
- 1340 VAVAILABILITY.

1341 **5.1.6 AVAILABILITY**

- 1342 • DTSTART and DTEND if specified MUST be date-time values.
- 1343 • DTSTART MUST appear once and signifies start of the free period.
- 1344 • Only one of DTEND or DURATION MAY appear and signify the end of the free period.

1345 **5.2 Recurrences.**

- 1346 • The RECURRENCE-ID is a property of each instance of a recurring event. It is calculated from the DT-
- 1347 START and the recurrence rules or added to the set by the RDATE property.
- 1348 • RDATE, EXDATE and RECURRENCE-ID must take the same form as the DTSTART. That is if DT-
- 1349 START is a DATE value then the RDATE and EXDATE must be DATE. If DTSTART is a date-time
- 1350 the RDATE and EXDATE values must take the same form, including the same timezone.
- 1351 • Overrides to an instance are specified by completely specifying the instance with the appropriate RE-
- 1352 CURRENCE-ID property.
- 1353 • An RDATE adds an instance to the recurrence set.
- 1354 • An EXDATE deletes an instance by specifying the recurrence id(s) to be deleted. Applications
- 1355 SHOULD NOT specify overrides for instances so deleted.
- 1356 • The recurrence set is calculated from the RRULE and RDATES and then applying any EXDATE prop-
- 1357 erties. That is EXDATE takes precedence over RDATE and the RRULE.

1358 **5.3 Alarms:**

- 1359 • Alarms are typically anchored to the start or end of an event or task. This is defined by the RELATED
- 1360 parameter to the TRIGGER property.

1361 **5.4 Unrecognized or unsupported elements**

- 1362 • A system SHOULD reject any attempt to store components which it does not support. A SYSTEM
- 1363 MUST advertise which components are supported through the use of the supportedCalendarCom-
- 1364 ponentSet property.
- 1365 • A system MUST ignore any elements it does not understand.

1366 **Appendix A Acknowledgments**

1367 The following individuals have participated in the creation of this specification and are gratefully acknowl-
1368 edged:

1369 **Participants:**

1370 Bruce Bartell, Southern California Edison
1371 Brad Benson, Trane
1372 Edward Cazalet, Individual
1373 Toby Considine, University of North Carolina at Chapel Hill
1374 William Cox, Individual
1375 Sharon Dinges, Trane
1376 Mike, Douglass, Rensselaer Polytechnic Institute
1377 Craig Gemmill, Tridium, Inc.
1378 Girish Ghatikar, Lawrence Berkeley National Laboratory
1379 Gerald Gray, Southern California Edison
1380 David Hardin, ENERNOC
1381 Gale Horst, Electric Power Research Institute (EPRI)
1382 Gershon Janssen, Individual
1383 Ed Koch, Akuacom Inc.
1384 Benoit Lepeuple, LonMark International*
1385 Carl Mattocks, CheckMi*
1386 Robert Old, Siemens AG
1387 Alexander Papaspyrou, Technische Universitat Dortmund
1388 Joshua Phillips, ISO/RTO Council (IRC)
1389 Jeremy J. Roberts, LonMark International
1390 David Thewlis, CalConnect

1391 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-
1392 Calendar Technical Committee, bridging to developing IETF standards and contributing the services defi-
1393 nitions that make up Services in Section 4. The Technical Committee gratefully acknowledges their assis-
1394 tance and cooperation as well. Contributors to TC XML include:

1395 Cyrus Daboo, Apple
1396 Mike Douglass, Rensselaer Polytechnic Institute
1397 Steven Lees, Microsoft
1398 Tong Li, IBM
1399

Appendix B Revision History

Revision	Date	Editor	Changes Made
Initial	Mar 15 2011	M. Douglass (CALCONNECT)	Initial publication - a first pass at a rewrite from CalWS-REST
WD01	July 15 2011	M. Douglass (CALCONNECT)	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.
WD02		M. Douglass (CALCONNECT)	Added href to fetch response. Change propstat to be extension of BaseResponseType
WD03	September 7 2011	M. Douglass (CALCONNECT)	Add test attribute to calendar query elements.
WD04	November 11 2011	M. Douglass (CALCONNECT)	Updated calendar query to use xcal types instead of names. Assumes a later version of the xcalendar schema to make this possible. Change references to "etoken" to "changeToken", Update the error codes with descriptions and a type per error. Added some new errors.
WD05	December 15 2011	M. Douglass (CALCONNECT)	Change example from CalDAV to CalWS
WD06	January 3 2012	M. Douglass (CALCONNECT)	Remove all references to XRD. Define CalWS properties in their place.
WD07	February 7 2012	M. Douglass (CALCONNECT)	Align more closely with the OASIS template. Correct one or two minor spelling errors.
WD08	02/13/12	M. Douglass	Initial hand-off from CalConnect to OASIS

Revision	Date	Editor	Changes Made
WD09	February 14 2012	M. Douglass T Considine	Change namespace to http://docs.oasis-open.org/ws-calendar/ns/soap Fixed example, broken references. Added namespace declaration Added Summary
Wd10	July 29, 2012	T Considine	Eliminated sentence as per Jira 463
WD11	November 6, 2012	M. Douglass	Add conformance section Added missing reference to RFC5546. Restructured into sections to allow future addition of extensions. Added short introductory text to new Section 3 - "Basic Calendar Access" Fixed small typo - getPropertiesReponse Removed out-of-date and unused reference to web-linking Removed bad and unnecessary reference in renumbered sections 4.3.2 and 4.3.4 Fixed reference to draft caldav scheduling to refer to the RFC

1401