

---

# 1 WS-Calendar SOAP-based Services Version 1.0

## 2 Working Draft 11

3 6 November 2012

### 4 Technical Committee:

5 OASIS Web Services Calendar (WS-Calendar) TC

### 6 Chair:

7 Toby Considine ([toby.considine@unc.edu](mailto:toby.considine@unc.edu)), University of North Carolina at Chapel Hill

### 8 Editor:

9 Michael Douglass ([douglm@rpi.edu](mailto:douglm@rpi.edu)) Rensselaer Polytechnic Institute

### 10 Related work:

11 This specification is related to:

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

### 16 Abstract:

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

### 21 Status:

22 This [Working Draft](#) (WD) has been produced by one or more TC Members; it has not yet been  
23 voted on by the TC or [approved](#) as a Committee Draft (Committee Specification Draft or a  
24 Committee Note Draft). The OASIS document [Approval Process](#) begins officially with a TC vote  
25 to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-  
26 approve it any number of times as a Committee Draft.

27  
28  
29 Copyright © OASIS Open 2011. All Rights Reserved.

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

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

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

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

---

# 48 Table of Contents

49	1 Introduction.....	5
50	1.1 Terminology.....	5
51	1.2 Normative References.....	5
52	1.3 Non-normative References.....	6
53	1.4 Namespace.....	6
54	2 Issues not addressed by this specification.....	7
55	2.1 Access Control.....	7
56	2.2 Provisioning.....	7
57	2.3 Copy/Move.....	7
58	2.4 Creating Collections.....	7
59	2.5 Retrieving collections.....	7
60	2.6 Setting service and resource properties.....	7
61	3 CalWS Glossary.....	8
62	3.1.1 Calendar Object Resource.....	8
63	3.1.2 Uid.....	8
64	3.1.3 Collections.....	8
65	3.1.4 Calendar Collection.....	8
66	3.1.5 Scheduling Calendar Collection.....	8
67	3.1.6 Principal Home.....	8
68	3.1.7 Change token.....	8
69	4 Basic Calendar Access.....	9
70	4.1 Overview of the CalWS protocol.....	9
71	4.1.1 Discovery.....	9
72	4.1.2 Properties.....	9
73	4.1.3 Operations.....	9
74	4.1.4 Calendar Object Resources.....	10
75	4.1.5 Timezone information.....	10
76	4.1.6 Error conditions.....	10
77	4.1.6.1 Example: error with error condition.....	10
78	4.2 CalWs-SOAP Messages.....	10
79	4.2.1 Common Elements and types.....	11
80	4.2.1.1 ErrorCodeType.....	12
81	4.2.1.2 BaseResponseType.....	14
82	4.3 Properties.....	14
83	4.3.1 childCollection.....	15
84	4.3.2 creationDateTime.....	15
85	4.3.3 displayName.....	15
86	4.3.4 lastModifiedDateTime.....	15
87	4.3.5 maxAttendeesPerInstance.....	15
88	4.3.6 maxDateTime.....	16
89	4.3.7 maxInstances.....	16
90	4.3.8 maxResourceSize.....	16
91	4.3.9 minDateTime.....	16

92	4.3.10 principalHome.....	17
93	4.3.11 resourceDescription.....	17
94	4.3.12 resourceOwner.....	17
95	4.3.13 resourceTimezoneId.....	17
96	4.3.14 resourceType.....	17
97	4.3.15 supportedCalendarComponentSet.....	18
98	4.3.16 supportedFeatures.....	18
99	4.3.17 timezoneServer.....	19
100	4.3.18 CalWS:privilege-set XML element.....	19
101	4.4 Retrieving Collection and Service Properties.....	19
102	4.4.1 Example - retrieving server properties:.....	20
103	4.5 Creating Calendar Object Resources.....	21
104	4.5.1 Preconditions for Calendar Object Creation.....	21
105	4.5.2 Example - successful addItem:.....	22
106	4.6 Retrieving resources.....	23
107	4.6.1 Example - successful fetchItem:.....	23
108	4.6.2 Example - unsuccessful fetchItem:.....	24
109	4.7 Updating resources.....	24
110	4.7.1 Change tokens and concurrent updates.....	28
111	4.7.2 Example - successful update:.....	29
112	4.7.3 Other updates:.....	30
113	4.7.4 Creating an update message.....	31
114	4.8 Deletion of resources.....	32
115	4.8.1 Example - successful deleteItem:.....	32
116	4.8.2 Example - unsuccessful deleteItem:.....	33
117	4.9 Querying calendar resources.....	33
118	4.9.1 Calendar Query common types.....	33
119	4.9.2 CompFilterType.....	34
120	4.9.3 PropFilterType.....	34
121	4.9.4 ParamFilterType.....	35
122	4.9.5 CalendarQueryType elements.....	36
123	4.9.6 Specifying data to be returned.....	36
124	4.9.7 Pre/postconditions for calendar queries.....	37
125	4.9.8 Time range limited queries.....	37
126	4.9.9 Example: time range limited retrieval.....	37
127	4.10 Free-busy queries.....	39
128	4.10.1 Element values .....	40
129	4.10.1.1 start.....	40
130	4.10.1.2 end.....	40
131	4.10.2 Examples.....	40
132	4.11 Multiple operations.....	42
133	5 Conformance.....	44
134	5.1 Start, end and duration in calendar components.....	44
135	5.1.1 Updating, transporting and maintaining start, and and duration.....	44
136	5.1.2 VEVENT:.....	44
137	5.1.3 VTODO:.....	44

138	5.1.4 VJOURNAL:.....	44
139	5.1.5 VAVAILABILITY.....	45
140	5.1.6 AVAILABILITY.....	45
141	5.2 Recurrences.....	45
142	5.3 Alarms:.....	45
143	5.4 Unrecognized or unsupported elements.....	45
144	Appendix A. Acknowledgments.....	46
145	Participants:.....	46
146	Appendix B. Revision History.....	47
147		

## 148 1 Introduction

149 The CalWS SOAP protocol is built upon and makes the same assumptions about structure as the  
150 CalDAV protocol defined in [RFC 4791] and related specifications. It does NOT require nor assume the  
151 WebDAV nor CalDAV protocol.

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

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

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

### 160 1.1 Terminology

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

### 164 1.2 Normative References

- 165     **[RFC 2119]**                 S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*.  
166     IETF RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 167     **[RFC 2616]**                 Fielding, et al, *Hypertext Transfer Protocol -- HTTP/1.1*  
168     <http://tools.ietf.org/html/rfc2616>
- 169     **[RFC 4791]**                 Daboo, et al. *Calendar Extensions to WebDAV (CalDAV)*.  
170     <http://www.ietf.org/rfc/rfc4791.txt>.
- 171     **[RFC 6638]**                 Desruisseaux, et al. *CalDAV Scheduling extensions to CalDAV*  
172     <http://tools.ietf.org/html/rfc6638>
- 173     **[RFC 5545]**                 B. Desruisseaux, *Internet Calendaring and Scheduling Core Object*  
174     *Specification (iCalendar)*  
175     <http://tools.ietf.org/html/rfc5546>
- 176     **[RFC 5546]**                 C. Daboo, *iCalendar Transport-Independent Interoperability Protocol*  
177     *(iTIP)*  
178     <http://tools.ietf.org/html/rfc5545>
- 179     **[RFC 6321]**                 C. Daboo, M. Douglass, S. Lees *xCal: The XML format for iCalendar*  
180     <http://www.ietf.org/rfc/rfc6321.txt>
- 181     **[draft-timezones]**         C. Daboo, M. Douglass: *Timezone Service Protocol*  
182     <http://tools.ietf.org/html/draft-douglass-timezone-service>
- 183     **[FreeBusy Read URL]**       E York. *Freebusy read URL*  
184     [http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)  
185     [%20V1.0.pdf](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)
- 186     **[SOAP11]**                 Simple Object Access Protocol (SOAP) 1.1, 8 May 2000  
187     <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 188
- 189     **[WSDL11]**                 Web Services Description Language (WSDL) 1.1, 15 March 2001  
190     <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

12  
191 [WS-Calendar] *WS-Calendar Version 1.0*. 19 January 2011. OASIS Committee Specification  
192 <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar->  
193 [spec-v1.0-cs01.pdf](http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf).

### 194 1.3 Non-normative References

195 [WS-Addr] W3C Recommendation, Web Services Addressing 1.0 - Core, and Web  
196 Services Addressing 1.0 - SOAP Binding, 9 May 2006  
197 <http://www.w3.org/2002/ws/addr/>  
198 [WT-I-Basic] Basic Profile Version 1.1, 10 April 2006  
199 <http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html>  
200 [WS-I-Bind] Web Services-Interoperability Organization (WS-I) Simple SOAP Binding Profile  
201 Version 1.0, 24 August 2004  
202 <http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html>

### 203 1.4 Namespace

204 XML namespaces and prefixes used in this standard:

205 Table 1-1: XML Namespaces in this standard

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

206

---

## 207 **2 Issues not addressed by this specification.**

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

### 210 **2.1 Access Control**

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

### 213 **2.2 Provisioning**

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

### 217 **2.3 Copy/Move**

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

### 222 **2.4 Creating Collections**

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

### 225 **2.5 Retrieving collections**

226 This operation is currently undefined.

### 227 **2.6 Setting service and resource properties.**

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

---

## 230 **3 CalWS Glossary**

### 231 **3.1.1 Calendar Object Resource**

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

### 234 **3.1.2 Uid**

235 The UID of an event is defined in [RFC 5545] as a "persistent, globally unique identifier for the calendar  
236 component". It is in fact, slightly more complicated in that all overrides to a recurring event have the same  
237 UID as the master event. Copies of a meeting invitation sent to attendees must also have the same UID.

238 In this protocol the UID is the key by which we locate calendar object resources (see above) and any  
239 associated overrides within a calendar collection (see below).

### 240 **3.1.3 Collections**

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

### 244 **3.1.4 Calendar Collection**

245 A collection only allowed to contain calendar object resources. The UIDs for components within a  
246 calendar collection must be unique. The combination of a calendar collection id and the UID MUST be a  
247 unique key within a set of resources made available through this service.

### 248 **3.1.5 Scheduling Calendar Collection**

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

### 251 **3.1.6 Principal Home**

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

### 254 **3.1.7 Change token**

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



---

## 260 4 Basic Calendar Access

261 This section defines properties, messages and operations sufficient to provide basic access and  
262 operations on a calendar store. These are sufficient to store, retrieve and update calendaring entities and  
263 to obtain various reports on the current state of the store.

264 Any service supporting this protocol MUST return a calendarAccessFeature element in the  
265 supportedFeatures property in the getPropertiesResponse message as specified in supportedFeatures

### 266 4.1 Overview of the CalWS protocol

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

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

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

274 A system entity acting as a CalWS requester transmits a CalWS request element within the body of a  
275 SOAP message to a system entity acting as a CalWS responder. The CalWS requester MUST NOT  
276 include more than one CalWS request per SOAP message or include any additional XML elements in the  
277 SOAP body (though see Section 4.1.1 for multiple messages packaged in one request).

278 The CalWS responder MUST return either a CalWS response element within the body of another SOAP  
279 message or generate a SOAP fault. The CalWS responder MUST NOT include more than one CalWS  
280 response per SOAP message or include any additional XML elements in the SOAP body. If a CalWS  
281 responder cannot, for some reason, process a CalWS request, it MUST generate a SOAP fault. (SOAP  
282 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)

#### 283 4.1.1 Discovery

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

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

#### 289 4.1.2 Properties

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

#### 293 4.1.3 Operations

294 The following operations are defined by this specification:

- 295 • Retrieval and update of service and resource properties
- 296 • Creation of a calendar object
- 297 • Retrieval of a single calendar object
- 298 • Multiget of one or more calendar objects
- 299 • Update of a calendar object
- 300 • Deletion of a calendar object

- 24  
301 • Query  
302 • Free-busy query  
303 • Multiple operations

#### 304 **4.1.4 Calendar Object Resources**

305 The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC 4791] section  
306 4.2. An additional constraint for CalWS is that no timezone specifications are transferred with the data.

#### 307 **4.1.5 Timezone information**

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

#### 313 **4.1.6 Error conditions**

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

317 A "precondition" for a method describes the state of the server that must be true for that method to be  
318 performed. A "postcondition" of a method describes the state of the server that must be true after that  
319 method has been completed. Any violation of these conditions will result in an error response in the  
320 message.

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

##### 325 **4.1.6.1 Example: error with error condition**

```
326 <?xml version="1.0" encoding="utf-8"  
327     xmlns:CW="http://docs.oasis-open.org/ws-calendar/ns/soap" ?>  
328 <CW:error>  
329   <CW:uidConflict>  
330     <CW:href>/user/mike/calendar/abcd-0123456789.ics</CW:href>  
331   </CW:uidConflict>  
332   <CW:description>Unknown property </CW:description>  
333 </CW:error>
```

#### 334 **4.2 CalWs-SOAP Messages.**

335 This section describes the common elements and structure of CalWs-SOAP messages. The conventions  
336 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		

337 *Table 1: Field column descriptions*338 **4.2.1 Common Elements and types**

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

341 All requests must include an href which specifies the target for the request. There is also an id attribute  
342 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.

343 *Table 2: BaseRequestType elements*

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

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

347 *Table 3: ErrorResponse elements*

30

#### 348 **4.2.1.1 ErrorCodeType**

349 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"> <li>1. The server does not allow the creation of calendar collections at the given location in its namespace, or</li> <li>2. The parent collection of the Request-URI exists but cannot accept members</li> </ol>
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.

Field	Type	Description
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 responsibility. 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.

350 *Table 4: ErrorCodeType definitions*351 **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.

352 *Table 5: BaseResponseType elements*353 **4.3 Properties**

354 The getPropertiesResponse message contains 0 or more properties defined below. Some properties  
 355 apply to the service as a whole while others apply only to the targeted resource. The targeted resource  
 356 may have property values which override those for the service. For example, the timezone identifier for a  
 357 particular collection may differ from the default timezone identifier for the system.

358 Each property is an XML complex type based on the GetPropertiesBasePropertyType.

39  
359

### 4.3.1 childCollection

360 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

361 *Table 6: ChildCollectionType fields*

362 See resourceType for descriptions of CollectionType and Calendar CollectionType.

### 4.3.2 creationDateTime

364 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

365 *Table 7: CreationDateTimeType fields*

### 4.3.3 displayName

367 This property SHOULD be returned for any targeted resource.

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

368 *Table 8: DisplayNameType fields*

### 4.3.4 lastModifiedDateTime

370 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

371 *Table 9: LastModifiedDateTimeType fields*

### 4.3.5 maxAttendeesPerInstance

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

42

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

375 *Table 10: MaxAttendeesPerInstanceType fields*376 **4.3.6 maxDateTime**

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

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

379 *Table 11: MaxDateTimeType fields*380 **4.3.7 maxInstances**

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

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

383 *Table 12: MaxInstancesType fields*384 **4.3.8 maxResourceSize**

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

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.

387 *Table 13: MaxResourceSizeType fields*388 **4.3.9 minDateTime**

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

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

391 *Table 14: MinDateTimeType fields*



45  
392

### 4.3.10 principalHome

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

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

395 *Table 15: PrincipalHomeType fields*

### 4.3.11 resourceDescription

397 Provides some descriptive text for the targeted collection.

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

398 *Table 16: ResourceDescriptionType fields*

### 4.3.12 resourceOwner

400 This property SHOULD be returned for any targeted resource.

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

401 *Table 17: ResourceownerType fields*

### 4.3.13 resourceTimezoneId

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

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

405 *Table 18: ResourceTimezoneIdType fields*

### 4.3.14 resourceType

407 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.

408 *Table 19: ResourceType fields*

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

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

410 *Table 20: XresourceType fields*411 **4.3.15 supportedCalendarComponentSet**412 This property identifies which component types the service is prepared to store. The allowable  
413 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.

414 *Table 21: SupportedCalendarComponentSetType fields*415 **4.3.16 supportedFeatures**416 This property SHOULD be returned for the service and MAY be returned for any targeted collection  
417 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.

418 *Table 22: SupportedFeaturesType fields*

51  
419 **4.3.17 timezoneServer**

420 This property SHOULD be returned for the service and MAY be returned for any targeted collection  
421 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.

422 *Table 23: TimezoneServerType fields*

423 **4.3.18 CalWS:privilege-set XML element**

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

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

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

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

- 430 • CalWS: Read - current principal has read access
- 431 • CalWS: Write - current principal has write access

```
432 <calws:privilege-set>
433 <calws:privilege><calws:read></calws:privilege>
434 <calws:privilege><calws:write></calws:privilege>
435 </calws:privilege-set>
```

436 **4.4 Retrieving Collection and Service Properties**

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

439 The service properties define the global limits and defaults. Any properties defined on collections within  
440 the service hierarchy override those service defaults. The service may choose to prevent such overriding  
441 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.

442 *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

443 *Table 25: GetPropertiesResponseType fields*444 **4.4.1 Example - retrieving server properties:**

```

445 >>Request
446
447 <?xml version="1.0" encoding="UTF-8"?>
448 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
449   <SOAP-ENV:Header/>
450   <SOAP-ENV:Body>
451     <ns2:getProperties xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
452       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
453       <ns2:href/></ns2:href>
454     </ns2:getProperties>
455   </SOAP-ENV:Body>
456 </SOAP-ENV:Envelope>
457
458 >>Response
459
460 <?xml version="1.0" encoding="UTF-8"?>
461 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
462   <SOAP-ENV:Header />
463   <SOAP-ENV:Body>
464     <ns2:getPropertiesResponse
465       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
466       xmlns:ns4="urn:ietf:params:xml:ns:icalendar-2.0"
467       id="0" >
468     <ns2:href/></ns2:href>
469     <ns2:lastModifiedDate>
470       <ns2:dateTime>2012-01-04T18:21:14Z</ns2:dateTime>
471     </ns2:lastModifiedDate>
472     <ns2:supportedCalendarComponentSet>
473       <ns4:vevent />
474       <ns4:vtodo />
475       <ns4:vavailability />
476     </ns2:supportedCalendarComponentSet>
477     <ns2:resourceType>
478       <ns2:collection />
479     </ns2:resourceType>
480     <ns2:supportedFeatures>
481       <ns2:calendarAccessFeature />
482     </ns2:supportedFeatures>
483     <ns2:maxInstances>
484       <ns2:integer>1000</ns2:integer>
485     </ns2:maxInstances>
486     <ns2:maxResourceSize>
487       <ns2:integer>100000</ns2:integer>
488     </ns2:maxResourceSize>
489     </ns2:getPropertiesResponse>
490   </SOAP-ENV:Body>
491 </SOAP-ENV:Envelope>
492
493

```

57

## 494 4.5 Creating Calendar Object Resources

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

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

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

500 Table 26: AddItemType fields

501 The service will respond with an AddItemResponseType giving either the href and change token of the  
502 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

503 Table 27: AddItemResponseType additional fields

### 504 4.5.1 Preconditions for Calendar Object Creation

- 505 • **CalWS:target-exists:** The entity already exists.
- 506 • **CalWS:not-calendar-data:** The resource submitted MUST be a supported media type (i.e., iCalendar)  
507 for calendar object resources;
- 508 • **CalWS:invalid-calendar-data:** The resource submitted MUST be valid data for the media type being  
509 specified (i.e., MUST contain valid iCalendar data);
- 510 • **CalWS:invalid-calendar-object-resource:** The resource submitted in the request MUST obey all  
511 restrictions specified in Calendar Object Resources (e.g., calendar object resources MUST NOT  
512 contain more than one type of calendar component, calendar object resources MUST NOT specify  
513 the iCalendar METHOD property, etc.);
- 514 • **CalWS:unsupported-calendar-component:** The resource submitted in the request MUST contain a  
515 type of calendar component that is supported in the targeted calendar collection;
- 516 • **CalWS:uid-conflict:** The resource submitted in the request MUST NOT specify an iCalendar UID  
517 property value already in use in the targeted calendar collection or overwrite an existing calendar  
518 object resource with one that has a different UID property value. Servers SHOULD report the URL  
519 of the resource that is already making use of the same UID property value in the CalWS:href  
520 element  
521 <!ELEMENT uid-conflict (CalWS:href)>
- 522 • **CalWS:exceeds-max-resource-size:** The resource submitted in the request MUST have an octet size  
523 less than or equal to the value of the CalDAV:max-resource-size property value on the calendar  
524 collection where the resource will be stored;
- 525 • **CalWS:before-min-date-time:** The resource submitted in the request MUST have all of its iCalendar  
526 DATE or DATE-TIME property values (for each recurring instance) greater than or equal to the  
527 value of the CalDAV:min-date-time property value on the calendar collection where the resource  
528 will be stored;

- 60  
529 • **CalWS:after-max-date-time:** The resource submitted in the request MUST have all of its iCalendar  
530 DATE or DATE-TIME property values (for each recurring instance) less than the value of the  
531 CalDAV:max-date-time property value on the calendar collection where the resource will be stored;  
532 • **CalWS:too-many-instances:** The resource submitted in the request MUST generate a number of  
533 recurring instances less than or equal to the value of the CalDAV: max-instances property value on  
534 the calendar collection where the resource will be stored;  
535 • **CalWS:too-many-attendees-per-instance:** The resource submitted in the request MUST have a  
536 number of ATTENDEE properties on any one instance less than or equal to the value of the  
537 CalDAV:max-attendees-per-instance property value on the calendar collection where the resource  
538 will be stored;

## 539 4.5.2 Example - successful addItem:

```
540 >>Request
541 <?xml version="1.0" encoding="UTF-8"?>
542 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
543   <SOAP-ENV:Header/>
544   <SOAP-ENV:Body>
545     <ns2:addItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
546       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
547       <ns2:href>/user/douglm/calendar</ns2:href>
548       <ns3:icalendar>
549         <ns3:vcalendar>
550           <ns3:components>
551             <ns3:vevent>
552               <ns3:properties>
553                 <ns3:uid>
554                   <ns3:text>1302064354993</ns3:text>
555                 </ns3:uid>
556                 <ns3:summary>
557                   <ns3:text>try this</ns3:text>
558                 </ns3:summary>
559                 <ns3:dtstart>
560                   <ns3:date-time>20110406T150000Z</ns3:date-time>
561                 </ns3:dtstart>
562                 <ns3:dtend>
563                   <ns3:date-time>20110406T160000Z</ns3:date-time>
564                 </ns3:dtend>
565               </ns3:properties>
566             </ns3:vevent>
567           </ns3:components>
568         </ns3:vcalendar>
569       </ns3:icalendar>
570     </ns2:addItem>
571   </SOAP-ENV:Body>
572 </SOAP-ENV:Envelope>
573
574 >>Response
575
576 <?xml version="1.0" encoding="UTF-8"?>
577 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
578   <SOAP-ENV:Header/>
579   <SOAP-ENV:Body>
580     <ns2:addItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
581       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
582       <ns2:status>OK</ns2:status>
583       <ns2:href>/user/douglm/calendar/1302064354993.ics</ns2:href>
584       <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
585     </ns2:addItemResponse>
586   </SOAP-ENV:Body>
587 </SOAP-ENV:Envelope>
```

63

## 589 4.6 Retrieving resources

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

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

593 *Table 28: FetchItemType fields*

594 The service will respond with a FetchItemResponseType containing either the change token, its href and  
595 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

596 *Table 29: FetchItemResponseType additional fields*

### 597 4.6.1 Example - successful fetchItem:

```

598 >>Request
599
600 <?xml version="1.0" encoding="UTF-8"?>
601 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
602   <SOAP-ENV:Header/>
603   <SOAP-ENV:Body>
604     <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
605                 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
606       <ns2:href>/user/doug1m/calendar/1302105461170.ics</ns2:href>
607     </ns2:fetchItem>
608   </SOAP-ENV:Body>
609 </SOAP-ENV:Envelope>
610
611 >>Response
612
613 <?xml version="1.0" encoding="UTF-8"?>
614 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
615   <SOAP-ENV:Header/>
616   <SOAP-ENV:Body>
617     <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
618                          xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
619       <ns2:status>OK</ns2:status>
620       <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
621       <ns2:href>/user/doug1m/calendar/1302105461170.ics</ns2:href>
622       <ns3:icalendar>
623         <ns3:vcalendar>
624           <ns3:properties>
625             <ns3:prodid>
626               <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
627             </ns3:prodid>
628             <ns3:version>
629               <ns3:text>2.0</ns3:text>
630             </ns3:version>
631           </ns3:properties>
632           <ns3:components>
633             <ns3:vevent>
634               <ns3:properties>

```

```

66
635         <ns3:created>
636             <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
637         </ns3:created>
638         <ns3:dtend>
639             <ns3:date-time>20110406T160000Z</ns3:date-time>
640         </ns3:dtend>
641         <ns3:dtstamp>
642             <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
643         </ns3:dtstamp>
644         <ns3:dtstart>
645             <ns3:date-time>20110406T150000Z</ns3:date-time>
646         </ns3:dtstart>
647         <ns3:last-modified>
648             <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
649         </ns3:last-modified>
650         <ns3:summary>
651             <ns3:text>try this</ns3:text>
652         </ns3:summary>
653         <ns3:uid>
654             <ns3:text>1302105461170</ns3:text>
655         </ns3:uid>
656         </ns3:properties>
657     </ns3:vevent>
658 </ns3:components>
659 </ns3:vcalendar>
660 </ns3:icalendar>
661 </ns2:fetchItemResponse>
662 </SOAP-ENV:Body>
663 </SOAP-ENV:Envelope>

```

## 664 4.6.2 Example - unsuccessful fetchItem:

```

665 >>Request
666
667 <?xml version="1.0" encoding="UTF-8"?>
668 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
669     <SOAP-ENV:Header/>
670     <SOAP-ENV:Body>
671         <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
672             xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
673             <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
674         </ns2:fetchItem>
675     </SOAP-ENV:Body>
676 </SOAP-ENV:Envelope>
677
678 >>Response
679
680 <?xml version="1.0" encoding="UTF-8"?>
681 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
682     <SOAP-ENV:Header/>
683     <SOAP-ENV:Body>
684         <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
685             xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
686             <ns2:status>Error</ns2:status>
687             <ns2:errorResponse>
688                 <ns2:targetDoesNotExist/>
689             </ns2:errorResponse>
690         </ns2:fetchItemResponse>
691     </SOAP-ENV:Body>
692 </SOAP-ENV:Envelope>

```

## 693 4.7 Updating resources

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

- 695 • An iCalendar element contains...
- 696 • a single vCalendar element which contains...
- 697 • one or more calendaring components, event, task etc each of which contain...



- 69  
 698 • zero or more components, alarms etc or one or more properties each of which contains...  
 699 • zero or more parameters and one or more values.

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

```

701     <icalendar>
702       <vcalendar>
703         <components>
704           <vevent>
705             <properties>
706               <uid>
707                 <text>1302064354993-a</text>
708               </uid>
709               <summary>
710                 <text>try this</text>
711               </summary>
712               <dtstart>
713                 <date-time>2011-07-18T15:00:00Z</date-time>
714               </dtstart>
715               <dtend>
716                 <date-time>2011-07-18T16:00:00Z</date-time>
717               </dtend>
718             </properties>
719           </vevent>
720         </components>
721       </vcalendar>
722     </icalendar>
  
```

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

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

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

- 731 • Remove - components, properties or parameters  
 732 • Add - components, properties or parameters  
 733 • Change - property or parameter values

734 Removals MUST be processed ahead of additions

735 Preconditions as specified in Preconditions for Calendar Object Creation are applicable. The response  
 736 will indicate success or failure of the update. If the change token value does not match that held by the  
 737 service a mismatchedChangeToken error status will be returned. The client should re-fetch the entity to  
 738 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	ComponentSelectionType	1..*	Y	Must select vcalendar

739 *Table 30: UpdateItem Type fields*

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

72

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

743 *Table 31: ComponentsSelectionType fields*

744 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

745 *Table 32: PropertiesSelectionType fields*746 To complete that pattern there is also a ParametersSelectionType used to select property parameters for  
747 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	ParameterReferenceType	0..1	N	Supplies parameters to remove
add	ParameterReferenceType	0..1	N	Species parameters to add

748 *Table 33: ParametersSelectionType fields*749 Each of these refers to a reference type. These either provide a complete entity for addition or identify the  
750 entity for removal. The three reference types are:

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

751 *Table 34: ComponentReferenceType fields*

75

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.

752 *Table 35: PropertyReferenceType fields*

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

753 *Table 36: ParameterReferenceType fields*

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

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

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

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

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

766 *Table 37: ComponentSelectionType fields*

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

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

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

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

773 *Table 38: PropertySelectionType fields*

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

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

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

778 *Table 39: ParameterSelectionType fields*

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

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

781 *Table 40: UpdateItemResponseType additional fields*

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

### 783 4.7.1 Change tokens and concurrent updates

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

791 If, on the other hand, the token has a section indicating that only participation status changes have been  
792 made, then other changes can take place. For a reference on implementing such a token see "Avoiding  
793 Conflicts when Updating Scheduling Object Resources" in [RFC 6638]. This describes the use of a  
794 schedule-tag.

81  
795

## 4.7.2 Example - successful update:

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

```
797 <ns3:icalendar>
798   <ns3:vcalendar>
799     <ns3:components>
800       <ns3:vevent>
801         <ns3:properties>
802           <ns3:uid>
803             <ns3:text>1302064354993-a</ns3:text>
804           </ns3:uid>
805           <ns3:summary>
806             <ns3:text>try this</ns3:text>
807           </ns3:summary>
808           <ns3:dtstart>
809             <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
810           </ns3:dtstart>
811           <ns3:dtend>
812             <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
813           </ns3:dtend>
814         </ns3:properties>
815       </ns3:vevent>
816     </ns3:components>
817   </ns3:vcalendar>
818 </ns3:icalendar>
```

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

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

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

```
825 >>Request
826
827 <?xml version="1.0" encoding="UTF-8"?>
828 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
829   <SOAP-ENV:Header/>
830   <SOAP-ENV:Body>
831     <ns2:updateItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
832       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
833       <ns2:href>/user/douglm/calendar/1302064354993-a.ics</ns2:href>
834       <ns2:changeToken>"20110802T032608Z-0" null</ns2:changeToken>
835       <ns2:select>
836         <ns3:vcalendar/>
837         <ns2:components>
838           <ns2:component>
839             <ns3:vevent>
840               <ns3:properties>
841                 <ns3:uid>
842                   <ns3:text>1302064354993-a</ns3:text>
843                 </ns3:uid>
844               </ns3:properties>
845             </ns3:vevent>
846           <ns2:properties>
847             <ns2:property>
848               <ns3:dtstart>
849                 <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
850               </ns3:dtstart>
851             <ns2:parameters>
852               <ns2:add>
853                 <ns3:tzid>
854                   <ns3:text>America/New_York</ns3:text>
855                 </ns3:tzid>
856               </ns2:add>
857             </ns2:parameters>
858             <ns2:change>
859               <ns3:dtstart>
```

```

84
860         <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
861         </ns3:dtstart>
862     </ns2:change>
863 </ns2:property>
864 <ns2:property>
865     <ns3:summary>
866     <ns3:text>try this</ns3:text>
867 </ns3:summary>
868 <ns2:change>
869     <ns3:summary>
870     <ns3:text>A changed summary - again and again and again</ns3:text>
871 </ns3:summary>
872 </ns2:change>
873 </ns2:property>
874 <ns2:add>
875     <ns3:categories>
876     <ns3:text>newcategory-2</ns3:text>
877     <ns3:text>resources</ns3:text>
878     <ns3:text>paper</ns3:text>
879 </ns3:categories>
880 </ns2:add>
881 </ns2:properties>
882 </ns2:component>
883 </ns2:components>
884 </ns2:select>
885 </ns2:updateItem>
886 </SOAP-ENV:Body>
887 </SOAP-ENV:Envelope>
888
889 >>Response
890
891 <?xml version="1.0" encoding="UTF-8"?>
892 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
893   <SOAP-ENV:Header/>
894   <SOAP-ENV:Body>
895     <ns2:updateItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
896                           xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
897                           id="0">
898       <ns2:status>OK</ns2:status>
899     </ns2:updateItemResponse>
900   </SOAP-ENV:Body>
901 </SOAP-ENV:Envelope>

```

### 902 4.7.3 Other updates:

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

- 905 • Addition of properties
- 906 • Removal of properties
- 907 • Addition of parameters to properties
- 908 • Removal of parameters from properties
- 909 • Changing parameter values.

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

```

913     <ns2:properties>
914     <ns2:property>
915     <ns3:dtstart>
916     <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
917     </ns3:dtstart>
918     <ns2:parameters>
919     <ns2:add>
920     <ns3:tzid>
921     <ns3:text>America/New_York</ns3:text>
922     </ns3:tzid>
923     </ns2:add>

```

```

87
924         </ns2:parameters>
925         <ns2:change>
926           <ns3:dtstart>
927             <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
928           </ns3:dtstart>
929         </ns2:change>
930       </ns2:property>
931     </ns2:properties>

```

932 In this example we add two categories to the event.

```

933     <ns2:properties>
934       <ns2:add>
935         <ns3:categories>
936           <ns3:text>paper</ns3:text>
937         </ns3:categories>
938       </ns2:add>
939       <ns2:add>
940         <ns3:categories>
941           <ns3:text>resources</ns3:text>
942         </ns3:categories>
943       </ns2:add>
944     </ns2:properties>

```

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

```

946     <ns2:properties>
947       <ns2:remove>
948         <ns3:dtend>
949           <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
950         </ns3:dtend>
951       </ns2:remove>
952       <ns2:add>
953         <ns3:duration>
954           <ns3:duration>PT1H</ns3:duration>
955         </ns3:duration>
956       </ns2:add>
957     </ns2:properties>

```

958 In this example we change the dtstart timezone identifier.

```

959     <ns2:properties>
960       <ns2:property>
961         <ns3:dtstart>
962           <ns3:parameters>
963             <ns3:tzid>
964               <ns3:text>America/New_York</ns3:text>
965             </ns3:tzid>
966           </ns3:parameters>
967           <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
968         </ns3:dtstart>
969         <ns2:parameters>
970           <ns2:parameter>
971             <ns3:tzid>
972               <ns3:text>America/New_York</ns3:text>
973             </ns3:tzid>
974           <ns2:change>
975             <ns3:tzid>
976               <ns3:text>America/Montreal</ns3:text>
977             </ns3:tzid>
978           </ns2:change>
979         </ns2:parameter>
980       </ns2:parameters>
981     </ns2:property>
982   </ns2:properties>

```

983

## 984 4.7.4 Creating an update message.

985 The update can be created in many ways but the most common approach is to build the update while  
986 modifications take place or to create one as the result of comparing old and new versions. It appears that

90  
 987 comparing XML for differences is difficult. However, we can take advantage of the structure of  
 988 calendaring entities to simplify the process. There are implementations available which take the diff  
 989 approach to producing an update stream.

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

994 These properties are

- 995 • categories
- 996 • exdate
- 997 • freebusy
- 998 • rdate

999 This normalization can take place before comparison.

1000 Some properties are multi-valued and may only appear once. At the moment the only standard property is  
 1001 resource which may take a comma separated list. This should be treated as a single multi-valued property  
 1002 when comparing. The order is unimportant. Sorting the values may help.

1003 Some properties may appear multiple times, for example comment. Comparison should take account of  
 1004 parameters. Ordering all properties appropriately allows for relatively simple comparison.

## 1005 4.8 Deletion of resources

1006 Deletion of calendar object resources is carried out by using a CalWS-SOAP deleteItem request with an  
 1007 href specifying the entity to be deleted. The deleteItem request is not valid when the href specifies a  
 1008 collection.

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

1009 *Table 41: DeleteItem fields*

1010 The service will respond with a DeleteItemResponseType containing the status and a possible error  
 1011 response. There are no additional elements.

### 1012 4.8.1 Example - successful deleteItem:

```

1013 >>Request
1014 <?xml version="1.0" encoding="UTF-8"?>
1015 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1016   <SOAP-ENV:Header/>
1017   <SOAP-ENV:Body>
1018     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1019       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1020       <ns2:href>/user/douglm/calendar/1302620814655.ics</ns2:href>
1021     </ns2:deleteItem>
1022   </SOAP-ENV:Body>
1023 </SOAP-ENV:Envelope>
1024
1025 >>Response
1026
1027 <?xml version="1.0" encoding="UTF-8"?>
1028 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1029   <SOAP-ENV:Header/>
1030   <SOAP-ENV:Body>
1031     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1032       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1033       <ns2:status>OK</ns2:status>
1034     </ns2:deleteItemResponse>
1035
  
```



93  
1036  
1037

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

## 1038 4.8.2 Example - unsuccessful deleteItem:

1039

>>Request

1040

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

1051

>>Response

1052

1053

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

1066

## 1067 4.9 Querying calendar resources

1068 Querying provides a mechanism by which information can be obtained from the service through possibly  
1069 complex queries. A skeleton icalendar entity can be provided to limit the amount of information returned to  
1070 the client. A query takes the parts

- 1071 • Limitations on the data returned
- 1072 • Selection of the data
- 1073 • Optional timezone id for floating time calculations.

### 1074 4.9.1 Calendar Query common types

1075 The UTCTimeRangeType is used in a number of places to define a time range within which components  
1076 must appear or property values must lie. The values are UTC time-date, the start is inclusive and the end  
1077 is exclusive.

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

1078 *Table 42: UTCTimeRangeType elements*

1079 The TextMatchType is used to match text values in properties and parameters. The collation attribute  
1080 species a collation as defined in [RFC4790].

1081 Servers are REQUIRED to support the "i;ascii-casemap" and "i;octet" collations which provide a basic  
1082 case insensitive and case sensitive match respectively.

96  
1083

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

1084 Table 43: TextMatchType attributes

## 1085 4.9.2 CompFilterType

1086 This type defines a search query for the calendar query operation. It specifies the component types to  
1087 return, absence tests or basic matching operations on properties and time ranges.

1088 The top level comp-filter element (which must match a vcalendar component may contain zero or more  
1089 comp-filter elements to match events, tasks or other contained components. These in turn may contain  
1090 further nested comp-filter elements to match further levels of nested components.

1091 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 BaseComponentType 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 elements. "allof" is a logical AND of the child elements.
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

1092 Table 44: CompFilterType elements

## 1093 4.9.3 PropFilterType

1094 The prop-filter element may test for the absence of a property or match values or specify zero or more  
1095 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

1096 *Table 45: PropFilterType elements*1097 **4.9.4 ParamFilterType**

1098 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	

1099 *Table 46: ParamFilterType elements*

## 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

1101 *Table 47: CalendarQueryType elements*

## 1102 4.9.6 Specifying data to be returned

1103 This is achieved by specifying one of the following

- 1104 • allprop: return all properties and calendar data. (some properties are specified as not being part of the  
1105 allprop set so are not returned)

- 105  
1106 • Set the icalendar element. This is an icalendar valueless pattern entity which provides a map of the  
1107 components and properties to be returned. Neither the pattern nor the returned result need to be  
1108 valid icalendar entities in that required properties may be absent if unselected.

## 1109 4.9.7 Pre/postconditions for calendar queries

1110 The preconditions as defined in [RFC 4791] Section 7.8 apply here. CalWS errors may be reported by the  
1111 service when preconditions or postconditions are violated.

## 1112 4.9.8 Time range limited queries.

1113 Time-range limited retrieval has some special characteristics. The simplest case is a single event or task  
1114 which overlaps the requested time-period. Recurring items and other components such as alarms  
1115 complicate the picture.

## 1116 4.9.9 Example: time range limited retrieval

1117 This example shows the time-range limited retrieval from a calendar which results in 2 events, one a  
1118 recurring event and one a simple non-recurring event.

```
1119 >> Request <<
1120 <?xml version="1.0" encoding="UTF-8"?>
1121 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1122 <SOAP-ENV:Header/>
1123 <SOAP-ENV:Body>
1124 <ns2:calendarQuery xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1125 <ns2:href>/user/douglm/calendar</ns2:href>
1126 <ns3:icalendar xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1127 <ns3:vcalendar>
1128 <ns3:components>
1129 <ns3:vevent>
1130 <ns3:properties>
1131 <ns3:summary/>
1132 <ns3:dtstart/>
1133 <ns3:dtend/>
1134 <ns3:duration/>
1135 <ns3:uid/>
1136 <ns3:recurrence-id/>
1137 <ns3:rrule/>
1138 <ns3:rdate/>
1139 <ns3:exdate/>
1140 </ns3:properties>
1141 </ns3:vevent>
1142 </ns3:components>
1143 </ns3:vcalendar>
1144 </ns3:icalendar>
1145 <ns2:filter>
1146 <ns2:compFilter test="anyof">
1147 <ns3:vcalendar />
1148 <ns2:compFilter>
1149 <ns3:vevent />
1150 <ns2:time-range end="20110430T040000Z" start="20110401T040000Z"/>
1151 </ns2:compFilter>
1152 </ns2:filter>
1153 </ns2:calendarQuery>
1154 </SOAP-ENV:Body>
1155 </SOAP-ENV:Envelope>
1156
1157 >> Response <<
1158 <?xml version="1.0" encoding="UTF-8"?>
1159 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1160 <SOAP-ENV:Header/>
1161 <SOAP-ENV:Body>
1162 <ns2:calendarQueryResponse
```

```

108
1166         xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1167         xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1168 <ns2:status>OK</ns2:status>
1169 <ns2:response>
1170   <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
1171   <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
1172   <ns2:propstat>
1173     <ns2:prop>
1174       <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1175         <ns3:icalendar>
1176           <ns3:vcalendar>
1177             <ns3:properties>
1178               <ns3:prodid>
1179                 <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1180               </ns3:prodid>
1181               <ns3:version>
1182                 <ns3:text>2.0</ns3:text>
1183               </ns3:version>
1184             </ns3:properties>
1185             <ns3:components>
1186               <ns3:vevent>
1187                 <ns3:properties>
1188                   <ns3:dtend>
1189                     <ns3:date-time>20110406T160000Z</ns3:date-time>
1190                   </ns3:dtend>
1191                   <ns3:dtstart>
1192                     <ns3:date-time>20110406T150000Z</ns3:date-time>
1193                   </ns3:dtstart>
1194                   <ns3:summary>
1195                     <ns3:text>try this</ns3:text>
1196                   </ns3:summary>
1197                   <ns3:uid>
1198                     <ns3:text>1302105461170</ns3:text>
1199                   </ns3:uid>
1200                 </ns3:properties>
1201               </ns3:vevent>
1202             </ns3:components>
1203           </ns3:vcalendar>
1204         </ns3:icalendar>
1205       </ns2:calendar-data>
1206     </ns2:prop>
1207   <ns2:status>OK</ns2:status>
1208 </ns2:propstat>
1209 </ns2:response>
1210 <ns2:response>
1211   <ns2:href>/user/douglm/calendar/CAL-00f1fc61-2f021bca-012f-022947f8-
1212 00000006.ics</ns2:href>
1213   <ns2:changeToken>"20110405T140920Z-0"</ns2:changeToken>
1214   <ns2:propstat>
1215     <ns2:prop>
1216       <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1217         <ns3:icalendar>
1218           <ns3:vcalendar>
1219             <ns3:properties>
1220               <ns3:prodid>
1221                 <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1222               </ns3:prodid>
1223               <ns3:version>
1224                 <ns3:text>2.0</ns3:text>
1225               </ns3:version>
1226             </ns3:properties>
1227             <ns3:components>
1228               <ns3:vevent>
1229                 <ns3:properties>
1230                   <ns3:duration>
1231                     <ns3:duration>PT1H</ns3:duration>
1232                   </ns3:duration>
1233                   <ns3:dtstart>
1234                   <ns3:parameters>
1235                     <ns3:tzid>
1236                       <ns3:text>America/New_York</ns3:text>

```

```

111
1237         </ns3:tzid>
1238         </ns3:parameters>
1239         <ns3:date-time>20110412T110000</ns3:date-time>
1240     </ns3:dtstart>
1241     <ns3:summary>
1242         <ns3:text>Test recurring event</ns3:text>
1243     </ns3:summary>
1244     <ns3:uid>
1245         <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1246 00000006demobedework@mysite.edu</ns3:text>
1247     </ns3:uid>
1248     <ns3:rrule>
1249         <ns3:recur>
1250             <ns3:freq>WEEKLY</ns3:freq>
1251             <ns3:count>2</ns3:count>
1252             <ns3:interval>1</ns3:interval>
1253         </ns3:recur>
1254     </ns3:rrule>
1255 </ns3:properties>
1256 </ns3:vevent>
1257 <ns3:vevent>
1258     <ns3:properties>
1259         <ns3:recurrence-id>
1260             <ns3:parameters>
1261                 <ns3:tzid>
1262                     <ns3:text>America/New_York</ns3:text>
1263                 </ns3:tzid>
1264             </ns3:parameters>
1265             <ns3:date-time>20110419T150000Z</ns3:date-time>
1266         </ns3:recurrence-id>
1267         <ns3:duration>
1268             <ns3:duration>PT1H</ns3:duration>
1269         </ns3:duration>
1270         <ns3:dtstart>
1271             <ns3:parameters>
1272                 <ns3:tzid>
1273                     <ns3:text>America/New_York</ns3:text>
1274                 </ns3:tzid>
1275             </ns3:parameters>
1276             <ns3:date-time>20110419T120000</ns3:date-time>
1277         </ns3:dtstart>
1278         <ns3:summary>
1279             <ns3:text>Test recurring event</ns3:text>
1280         </ns3:summary>
1281         <ns3:uid>
1282             <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1283 00000006demobedework@mysite.edu</ns3:text>
1284         </ns3:uid>
1285     </ns3:properties>
1286 </ns3:vevent>
1287 </ns3:components>
1288 </ns3:vcalendar>
1289 </ns3:icalendar>
1290 </ns2:calendar-data>
1291 </ns2:prop>
1292     <ns2:status>OK</ns2:status>
1293 </ns2:propstat>
1294 </ns2:response>
1295 </ns2:calendarQueryResponse>
1296 </SOAP-ENV:Body>
1297 </SOAP-ENV:Envelope>
1298

```

## 1299 4.10 Free-busy queries

1300 Freebusy queries are used to obtain freebusy information for a principal. The result contains information  
1301 only for events to which the current principal has sufficient access and may be affected by components  
1302 and rules available only to the server (for instance office hours availability).

114  
1303 These queries are carried out by using a CalWS-SOAP freebusyReport request with an href specifying a  
1304 principal. The freebusyReport request is not valid when the href specifies any entity other than a principal.  
1305 The query follows the specification defined in [FreeBusy Read URL] with certain limitations. As an  
1306 authenticated user to the CalWS service scheduling read-freebusy privileges must have been granted. As  
1307 an unauthenticated user equivalent access must have been granted to unauthenticated users.  
1308 Freebusy information is returned by default as xcalendar vfreebusy components, as defined by [RFC  
1309 6321]. Such a component is not meant to conform to the requirements of VFREEBUSY components in  
1310 RFC 5546. The VFREEBUSY component SHOULD conform to section "4.6.4 Free/Busy Component" of  
1311 [RFC 5545]. A client SHOULD ignore the ORGANIZER field.  
1312 Since a Freebusy query can only refer to a single user, a client will already know how to match the result  
1313 component to a user. A server MUST only return a single vfreebusy component.

## 1314 4.10.1 Element values

1315 Three values are provided: href; start; end. Only the href is required. The start and end are in XML UTC  
1316 date/time format and are interpreted as follows:

### 1317 4.10.1.1 start

1318 **Default:** If omitted the default value is left up to the server. It may be the current day, start of the  
1319 current month, etc.  
1320 **Description:** Specifies the start date for the Freebusy data. The server is free to ignore this value  
1321 and return data in any time range. The client must check the data for the returned time range.  
1322 **Format:** An XML UTC date-time  
1323 **Example:**  
1324 2011-12-01T10:15:00Z  
1325 **Notes:** Specifying only a start date/time without specifying an end-date/time or period should be  
1326 interpreted as in [RFC 5545]. The effective period should cover the remainder of that day.

### 1327 4.10.1.2 end

1328 **Default:** Same as start  
1329 **Description:** Specifies the end date for the Freebusy data. The server is free to ignore this value.  
1330 **Format:** Same as start  
1331 **Example:** Same as start  
1332 The server is free to ignore the start, end and period parameters. It is recommended that the server return  
1333 at least 6 weeks of data from the current day.  
1334 A client MUST check the time range in the response as a server may return a different time range than  
1335 the requested range.

## 1336 4.10.2 Examples

1337 The following is an unsuccessful request targeting an invalid resource.

```
1338 >> Request <<
1339
1340 <?xml version="1.0" encoding="UTF-8"?>
1341 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1342   <SOAP-ENV:Header/>
1343   <SOAP-ENV:Body>
1344     <ns2:freebusyReport
1345       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1346       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1347       <ns2:href>/user/douglm/calendar</ns2:href>
1348       <ns2:time-range>
1349         <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1350         <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1351       </ns2:time-range>
1352     </ns2:freebusyReport>
```



```

117      </SOAP-ENV:Body>
1353    </SOAP-ENV:Envelope>
1354
1355    >> Response <<
1356
1357    <?xml version="1.0" encoding="UTF-8"?>
1358    <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1359      <SOAP-ENV:Header/>
1360      <SOAP-ENV:Body>
1361        <ns2:freebusyReportResponse
1362          xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1363          xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1364          <ns2:status>Error</ns2:status>
1365          <ns2:message>Only principal href supported</ns2:message>
1366        </ns2:freebusyReportResponse>
1367      </SOAP-ENV:Body>
1368    </SOAP-ENV:Envelope>
1369

```

1370 The following is an example of a request to retrieve Freebusy data for a user:

```

1371    >> Request <<
1372
1373    <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1374      <SOAP-ENV:Header/>
1375      <SOAP-ENV:Body>
1376        <ns2:freebusyReport
1377          xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1378          xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1379          <ns2:href>/principals/users/douglm</ns2:href>
1380          <ns2:time-range>
1381            <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1382            <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1383          </ns2:time-range>
1384        </ns2:freebusyReport>
1385      </SOAP-ENV:Body>
1386    </SOAP-ENV:Envelope>
1387
1388    >> Response <<
1389
1390    <?xml version="1.0" encoding="UTF-8"?>
1391    <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1392      <SOAP-ENV:Header/>
1393      <SOAP-ENV:Body>
1394        <ns2:freebusyReportResponse
1395          xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1396          xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1397          <ns2:status>OK</ns2:status>
1398          <ns3:icalendar>
1399            <ns3:vcalendar>
1400              <ns3:properties>
1401                <ns3:prodid>
1402                  <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1403                </ns3:prodid>
1404                <ns3:version>
1405                  <ns3:text>2.0</ns3:text>
1406                </ns3:version>
1407              </ns3:properties>
1408              <ns3:components>
1409                <ns3:vfreebusy>
1410                  <ns3:properties>
1411                    <ns3:attendee>
1412                      <ns3:parameters>
1413                        <ns3:partstat>
1414                          <ns3:text>NEEDS-ACTION</ns3:text>
1415                        </ns3:partstat>
1416                      </ns3:parameters>
1417                      <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1418                    </ns3:attendee>
1419                    <ns3:created>
1420                      <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1421                    </ns3:created>
1422                    <ns3:dtend>

```

```

120      <ns3:date-time>2011-04-30T00:00:00Z</ns3:date-time>
1423    </ns3:dtend>
1424    <ns3:dtstamp>
1425      <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1426    </ns3:dtstamp>
1427    <ns3:dtstart>
1428      <ns3:date-time>2011-04-01T00:00:00Z</ns3:date-time>
1429    </ns3:dtstart>
1430    <ns3:freebusy>
1431      <ns3:parameters>
1432        <ns3:fctype>
1433          <ns3:text>BUSY</ns3:text>
1434        </ns3:fctype>
1435      </ns3:parameters>
1436      <ns3:period>
1437        <ns3:start>2011-04-06T15:00:00Z</ns3:start>
1438        <ns3:end>2011-04-06T16:00:00Z</ns3:end>
1439      </ns3:period>
1440    </ns3:freebusy>
1441    <ns3:last-modified>
1442      <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1443    </ns3:last-modified>
1444    <ns3:organizer>
1445      <ns3:parameters/>
1446      <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1447    </ns3:organizer>
1448    <ns3:uid>
1449      <ns3:text>2UTDVPZ9H0EQL9QISI44SP5IFPC4N75</ns3:text>
1450    </ns3:uid>
1451  </ns3:properties>
1452 </ns3:vfreebusy>
1453 </ns3:components>
1454 </ns3:vcalendar>
1455 </ns3:icalendar>
1456 </ns2:freebusyReportResponse>
1457 </SOAP-ENV:Body>
1458 </SOAP-ENV:Envelope>
1459

```

1460

## 1461 4.11 Multiple operations

1462 Each of the previously described operations acts upon a single entity or resource only. Frequently we  
1463 have the need to update an interconnected set of entities so that we maintain the consistency of the  
1464 structure. This requires an atomic operation which can successfully update all the entities or roll back the  
1465 operation on failure.

1466 The MultiOpType operation provides such a feature. It is essentially a wrapper around any of the other  
1467 operations which guarantees the success of the entire set or a roll back. Using the id attribute for  
1468 requests, each individual response can be located in the result.

1469 The MultiOpType request takes the following elements

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

1470 Table 48: MultiOpType elements

1471 The response type is also simple containing a single element containing all the responses.

123

<b>Field</b>	<b>Type</b>	<b>#</b>	<b>?</b>	<b>Description</b>
responses	Sequence of BaseResponseType	1	Y	Contains zero or more responses

1472 Table 49: MultiOpResponseType elements

1473

1474

1475

1476

## 1477 5 Conformance

1478 Certain calendaring properties and components are interrelated and it is necessary to have knowledge of  
 1479 all these properties and their current values to allow consistent update and understanding of a target  
 1480 component. The normative definition for these relationships is RFC5445, RFC5446 and related RFCs.

1481 However, those specifications assume a complete view of entities being fetched or updated. This  
 1482 specification allows updates of entities when only a partial view is available. In fact it is the very nature of  
 1483 SOAP based transaction to provide such a partial view. Given that, parties attempting to update entities  
 1484 MUST have sufficient information to ensure the end result is consistent. Services allowing updates to  
 1485 entities MUST ensure that the result after an update operation is still internally consistent.

### 1486 5.1 Start, end and duration in calendar components

1487 A period of time is fully specified by a start and an end or duration.

#### 1488 5.1.1 Updating, transporting and maintaining start, and and duration.

- 1489 • For all components the calculated or specified start must be at or before the end.
- 1490 • When a system updates or stores a calendar component it MUST retain the relationship of start, end  
 1491 and duration. Applications MUST NOT without good cause, change a start and end pair into a start  
 1492 and duration nor the reverse. Semantically they are not equivalent when DST transitions occur  
 1493 during the time of the event.
- 1494 • For interoperability, iCalendar based systems SHOULD avoid the use of weekly durations and XML  
 1495 based systems SHOULD avoid the use of yearly durations.

#### 1496 5.1.2 VEVENT:

- 1497 • The three properties are DTSTART, DTEND and DURATION.
- 1498 • DTSTART MUST appear once and only one of DTEND or DURATION MAY be present.
- 1499 • The DTSTART property for a VEVENT specifies the inclusive start of the event. For recurring events, it  
 1500 also specifies the very first instance in the recurrence set.
- 1501 • The DTEND property for a VEVENT calendar component specifies the non-inclusive end of the event.
- 1502 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE value  
 1503 type but no DTEND nor DURATION property, the event's duration is taken to be one day.
- 1504 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE-TIME  
 1505 value type but no DTEND nor DURATION property, the event ends on the same calendar date and  
 1506 time of day specified by the DTSTART property, that is, it signifies a zero length instant in time.

#### 1507 5.1.3 VTODO:

- 1508 • The three properties are DTSTART, DUE, DURATION.
- 1509 • DTSTART MAY appear once.
- 1510 • Either DUE or DURATION MAY appear in a VTODO, but DUE and DURATION MUST NOT occur in  
 1511 the same VTODO.
- 1512 • If DURATION does appear in a VTODO, then DTSTART MUST also appear in the same VTODO.
- 1513 • The three properties for a VTODO are related in the same way as for VEVENT. Additionally a VTODO  
 1514 calendar component without the DTSTART and DUE (or DURATION) properties specifies a  
 1515 VTODO that will be associated with each successive calendar date, until it is completed.

#### 1516 5.1.4 VJOURNAL:

- 1517 • DTSTART only, which may be a date or date-time value.

129  
1518

### 5.1.5 VAVAILABILITY

- 1519 • DTSTART and DTEND if specified MUST be date-time values.
- 1520 • DTSTART MAY appear once and signifies start of the busy period.
- 1521 • Only one of DTEND or DURATION MAY appear and signify the end of the busy period.
- 1522 • If DURATION does appear in a VAVAILABILITY, then DTSTART MUST also appear in the same  
1523 VAVAILABILITY.

### 1524 5.1.6 AVAILABILITY

- 1525 • DTSTART and DTEND if specified MUST be date-time values.
- 1526 • DTSTART MUST appear once and signifies start of the free period.
- 1527 • Only one of DTEND or DURATION MAY appear and signify the end of the free period.

## 1528 5.2 Recurrences.

- 1529 • The RECURRENCE-ID is a property of each instance of a recurring event. It is calculated from the  
1530 DTSTART and the recurrence rules or added to the set by the RDATE property.
- 1531 • RDATE, EXDATE and RECURRENCE-ID must take the same form as the DTSTART. That is if  
1532 DTSTART is a DATE value then the RDATE and EXDATE must be DATE. If DTSTART is a date-  
1533 time the RDATE and EXDATE values must take the same form, including the same timezone.
- 1534 • Overrides to an instance are specified by completely specifying the instance with the appropriate  
1535 RECURRENCE-ID property.
- 1536 • An RDATE adds an instance to the recurrence set.
- 1537 • An EXDATE deletes an instance by specifying the recurrence id(s) to be deleted. Applications  
1538 SHOULD NOT specify overrides for instances so deleted.
- 1539 • The recurrence set is calculated from the RRULE and RDATES and then applying any EXDATE  
1540 properties. That is EXDATE takes precedence over RDATE and the RRULE.

## 1541 5.3 Alarms:

- 1542 • Alarms are typically anchored to the start or end of an event or task. This is defined by the RELATED  
1543 parameter to the TRIGGER property.

## 1544 5.4 Unrecognized or unsupported elements

- 1545 • A system SHOULD reject any attempt to store components which it does not support. A SYSTEM  
1546 MUST advertise which components are supported through the use of the  
1547 supportedCalendarComponentSet property.
- 1548 • A system MUST ignore any elements it does not understand.

---

## 1549 Appendix A. Acknowledgments

1550 The following individuals have participated in the creation of this specification and are gratefully  
1551 acknowledged:

### 1552 Participants:

1553 Bruce Bartell, Southern California Edison  
1554 Brad Benson, Trane  
1555 Edward Cazalet, Individual  
1556 Toby Considine, University of North Carolina at Chapel Hill  
1557 William Cox, Individual  
1558 Sharon Dinges, Trane  
1559 Mike, Douglass, Rensselaer Polytechnic Institute  
1560 Craig Gemmill, Tridium, Inc.  
1561 Girish Ghatikar, Lawrence Berkeley National Laboratory  
1562 Gerald Gray, Southern California Edison  
1563 David Hardin, ENERNOC  
1564 Gale Horst, Electric Power Research Institute (EPRI)  
1565 Gershon Janssen, Individual  
1566 Ed Koch, Akuacom Inc.  
1567 Benoit Lepeuple, LonMark International\*  
1568 Carl Mattocks, CheckMi\*  
1569 Robert Old, Siemens AG  
1570 Alexander Papaspyrou, Technische Universitat Dortmund  
1571 Joshua Phillips, ISO/RTO Council (IRC)  
1572 Jeremy J. Roberts, LonMark International  
1573 David Thewlis, CalConnect

1574 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-  
1575 Calendar Technical Committee, bridging to developing IETF standards and contributing the services  
1576 definitions that make up Services in Section 4. The Technical Committee gratefully acknowledges their  
1577 assistance and cooperation as well. Contributors to TC XML include:

1578 Cyrus Daboo, Apple  
1579 Mike Douglass, Rensselaer Polytechnic Institute  
1580 Steven Lees, Microsoft  
1581 Tong Li, IBM

1582

1583 **Appendix B. Revision History**

<b>Revision</b>	<b>Date</b>	<b>Editor</b>	<b>Changes Made</b>
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

<b>Revision</b>	<b>Date</b>	<b>Editor</b>	<b>Changes Made</b>
WD09	February 14 2012	M. Douglass T Considine	Change namespace to <a href="http://docs.oasis-open.org/ws-calendar/ns/soap">http://docs.oasis-open.org/ws-calendar/ns/soap</a> 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