



CalWSSOAP - SOAP Web service protocol for calendaring

Version 1.0

3 January 2012

Specification URIs:

This Version:

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

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

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

Previous Version:

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

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

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

Latest Version:

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

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

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

Technical Committee:

CalConnect TC-XML

Chair(s):

[Chair name]

Editor(s):

Michael A Douglass

Related Work:

This specification is related to:

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

Declared XML Namespace(s):

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

Declared Properties and Relations Namespaces

Properties and extended relation types are prefixed with the URL"

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

33 **Abstract:**

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

35 **Status:**

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

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

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

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

Notices

49

50 Copyright © OASIS® 2008. All Rights Reserved.

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

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

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

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

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

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

78 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
79 might be claimed to pertain to the implementation or use of the technology described in this document or
80 the extent to which any license under such rights might or might not be available; neither does it
81 represent that it has made any effort to identify any such rights. Information on OASIS' procedures with
82 respect to rights in any document or deliverable produced by an OASIS Technical Committee can be
83 found on the OASIS website. Copies of claims of rights made available for publication and any
84 assurances of licenses to be made available, or the result of an attempt made to obtain a general license
85 or permission for the use of such proprietary rights by implementers or users of this OASIS Committee
86 Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no
87 representation that any information or list of intellectual property rights will at any time be complete, or
88 that any claims in such list are, in fact, Essential Claims.

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

94

95 Table of Contents

96	1 Introduction.....	6
97	1.1 Terminology.....	6
98	1.2 Normative References.....	6
99	1.3 Non-normative References.....	7
100	2 Issues not addressed by this specification.....	8
101	2.1 Access Control.....	8
102	2.2 Provisioning.....	8
103	2.3 Copy/Move.....	8
104	2.4 Creating Collections.....	8
105	2.5 Retrieving collections.....	8
106	2.6 Setting service and resource properties.....	8
107	3 CalWS Glossary.....	9
108	3.1 Calendar Object Resource.....	9
109	3.2 Uid.....	9
110	3.3 Collections.....	9
111	3.4 Calendar Collection.....	9
112	3.5 Scheduling Calendar Collection.....	9
113	3.6 Principal Home.....	9
114	3.7 Change token.....	9
115	4 Overview of the CalWS protocol.....	10
116	4.1 Discovery.....	10
117	4.2 Properties.....	10
118	4.3 Operations.....	10
119	4.4 Calendar Object Resources.....	11
120	4.5 Timezone information.....	11
121	4.6 Error conditions.....	11
122	5 CalWs-SOAP Messages.....	12
123	5.1 Common Elements and types.....	12
124	6 Properties.....	16
125	6.1 childCollection.....	16
126	6.2 creationDateTime.....	16
127	6.3 displayName.....	16
128	6.4 lastModifiedDateTime.....	16
129	6.5 maxAttendeesPerInstance.....	17
130	6.6 maxDateTime.....	17
131	6.7 maxInstances.....	17
132	6.8 maxResourceSize.....	17
133	6.9 minDateTime.....	17
134	6.10 principalHome.....	18

135	6.11 resourceDescription.....	18
136	6.12 resourceOwner.....	18
137	6.13 resourceTimezoneId.....	18
138	6.14 resourceType.....	18
139	6.15 supportedCalendarComponentSet.....	19
140	6.16 supportedFeatures.....	19
141	6.17 timezoneServer.....	19
142	6.18 CalWS:privilege-set XML element.....	20
143	7 Retrieving Collection and Service Properties.....	21
144	7.1 Example - retrieving server properties:.....	21
145	8 Creating Calendar Object Resources.....	23
146	8.1 Preconditions for Calendar Object Creation.....	23
147	8.2 Example - successful addItem:.....	24
148	8.3 Example - unsuccessful addItem:.....	24
149	9 Retrieving resources.....	25
150	9.1 Example - successful fetchItem:.....	25
151	9.2 Example - unsuccessful fetchItem:.....	26
152	10 Updating resources.....	27
153	10.1 Change tokens and concurrent updates.....	30
154	10.2 Example - successful update:.....	30
155	10.3 Other updates:.....	32
156	10.4 Creating an update message.....	33
157	11 Deletion of resources.....	35
158	11.1 Example - successful deleteItem:.....	35
159	11.2 Example - unsuccessful deleteItem:.....	35
160	12 Querying calendar resources.....	37
161	12.1 Calendar Query common types.....	37
162	12.2 CompFilterType.....	37
163	12.3 PropFilterType.....	38
164	12.4 ParamFilterType.....	38
165	12.5 CalendarQueryType elements.....	39
166	12.6 Specifying data to be returned.....	40
167	12.7 Pre/postconditions for calendar queries.....	40
168	12.8 Time range limited queries.....	40
169	12.9 Example: time range limited retrieval.....	40
170	13 Free-busy queries.....	44
171	13.1 Element values	44
172	13.2 Examples.....	45
173	14 Multiple operations.....	47
174	# Conformance.....	48
175		

1 Introduction

176

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

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

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

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

1.1 Terminology

188

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

1.2 Normative References

192

- 193 [RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC
194 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 195 [RFC 2616] Fielding, et al, *Hypertext Transfer Protocol -- HTTP/1.1*
196 <http://tools.ietf.org/html/rfc2616>
- 197 [RFC 3339] Klyne g., Newman C., *Date and Time on the Internet: Timestamps*
198 <http://tools.ietf.org/html/rfc3339>
- 199 [RFC 4790] Newman, et al. *Internet Application Protocol Collation Registry*.
200 <http://www.ietf.org/rfc/rfc4790.txt>.
- 201 [RFC 4791] Daboo, et al. *Calendar Extensions to WebDAV (CalDAV)*.
202 <http://www.ietf.org/rfc/rfc4791.txt>.
- 203 [draft caldav-sched] Desruisseaux, et al. *CalDAV Scheduling extensions to WebDAV*
204 <http://tools.ietf.org/html/draft-desruisseaux-caldav-sched-08>
- 205 [RFC 4918] L. Dusseault, *HTTP Extensions for Web Distributed Authoring and Versioning*
206 *(WebDAV)*
207 <http://tools.ietf.org/html/rfc4918>
- 208 [RFC 5545] B. Desruisseaux, *Internet Calendaring and Scheduling Core Object Specification*
209 *(iCalendar)*
210 <http://tools.ietf.org/html/rfc5545>
- 211 [RFC 5546] C. Daboo. *iCalendar Transport-Independent Interoperability Protocol (iTIP)*
212 <http://tools.ietf.org/html/rfc5546>
- 213 [draft-xcal] C. Daboo, M. Douglass, S. Lees *xCal: The XML format for iCalendar*
214 <https://datatracker.ietf.org/idtracker/draft-daboo-et-al-icalendar-in-xml>
- 215 [draft-timezones] C. Daboo, M. Douglass: *Timzone Service Protocol*
216 <http://tools.ietf.org/html/draft-douglass-timezone-service>

217 **[FreeBusy Read URL]** E York. *Freebusy read URL*
 218 <http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL>
 219 [%20V1.0.pdf](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)

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

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

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

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

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

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

234 1.3 Non-normative References

235 **[Reference]** [reference citation]

236 **[Reference]** [reference citation]

237

238

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

OASIS

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

Title (italicized or in quotation marks)

Approval Date (Month YYYY)

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

For example:

[EDXL-HAVE] OASIS Standard, "Emergency Data Exchange Language (EDXL) Hospital AVailability Exchange (HAVE) Version 1.0", November 2008.

http://docs.oasis-open.org/emergency/edxl-have/os/emergency_edxl_have-1.0-spec-os.doc

240 **2 Issues not addressed by this specification.**

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

243 **2.1 Access Control**

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

246 **2.2 Provisioning**

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

250 **2.3 Copy/Move**

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

255 **2.4 Creating Collections**

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

258 **2.5 Retrieving collections**

259 This operation is currently undefined.

260 **2.6 Setting service and resource properties.**

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

263 **3 CalWS Glossary**

264 **3.1 Calendar Object Resource**

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

267 **3.2 Uid**

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

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

273 **3.3 Collections**

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

277 **3.4 Calendar Collection**

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

281 **3.5 Scheduling Calendar Collection**

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

284 **3.6 Principal Home**

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

287 **3.7 Change token**

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

293 4 Overview of the CalWS protocol

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

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

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

301 A system entity acting as a CalWS requester transmits a CalWS request element within the body of a
302 SOAP message to a system entity acting as a CalWS responder. The CalWS requester MUST NOT
303 include more than one CalWS request per SOAP message or include any additional XML elements in the
304 SOAP body (though see Section 14 for multiple messages packaged in one request).

305 The CalWS responder MUST return either a CalWS response element within the body of another SOAP
306 message or generate a SOAP fault. The CalWS responder MUST NOT include more than one CalWS
307 response per SOAP message or include any additional XML elements in the SOAP body. If a CalWS
308 responder cannot, for some reason, process a CalWS request, it MUST generate a SOAP fault. (SOAP
309 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)

310 4.1 Discovery

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

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

316 4.2 Properties

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

320 4.3 Operations

321 The following operations are defined by this specification:

- 322 • Retrieval and update of service and resource properties
- 323 • Creation of a calendar object
- 324 • Retrieval of a single calendar object
- 325 • Multiget of one or more calendar objects
- 326 • Update of a calendar object
- 327 • Deletion of a calendar object
- 328 • Query
- 329 • Free-busy query
- 330 • Multiple operations

331 4.4 Calendar Object Resources

332 The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC 4791] section
333 4.2. An additional constraint for CalWS is that no timezone specifications are transferred with the data.

334 4.5 Timezone information

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

340 4.6 Error conditions

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

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

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

352 Example: error with error condition

```
353 <?xml version="1.0" encoding="utf-8"  
354     xmlns:CW="http://docs.oasis-open.org/ns/wscal/calws-soap"  
355     xmlns:C="urn:ietf:params:xml:ns:caldav" ?>  
356 <CW:error>  
357   <CW:uidConflict>  
358     <CW:href>/user/mike/calendar/abcd-0123456789.ics</CW:href>  
359   </CW:uidConflict>  
360   <CW:description>Unknown property </CW:description>  
361 </CW:error>
```

362

5 CalWs-SOAP Messages.

363
364

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

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

365 *Table 1: Field column descriptions*

366

5.1 Common Elements and types

367
368

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

369
370

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

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

371 *Table 2: BaseRequestType elements*372
373
374

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

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

375 *Table 3: ErrorResponse elements*

376 **ErrorCodeType**

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

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

Field	Type	Description
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.

378 Table 4: ErrorCodeType definitions

379 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.

380 Table 5: BaseResponseType elements

6 Properties

381

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

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

6.1 childCollection

387

388 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

389 *Table 6: ChildCollectionType fields*

390 See resourceType for descriptions of CollectionType and Calendar CollectionType.

6.2 creationDateTime

391

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

Field	Type	#	?	Description
dateTime	dateTime	1	Y	A date-time as defined in [RFC 3339] Section 5.6.

393 *Table 7: CreationDateTimeType fields*

6.3 displayName

394

395 This property SHOULD be returned for any targeted resource.

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

396 *Table 8: DisplayNameType fields*

6.4 lastModifiedDateTime

397

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

Field	Type	#	?	Description
dateTime	dateTime	1	Y	A date-time as defined in [RFC 3339] Section 5.6.

399 *Table 9: LastModifiedDateTimeType fields*

400 **6.5 maxAttendeesPerInstance**

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

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

403 *Table 10: MaxAttendeesPerInstanceType fields*

404 **6.6 maxDateTime**

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

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

407 *Table 11: MaxDateTimeType fields*

408 **6.7 maxInstances**

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

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

411 *Table 12: MaxInstancesType fields*

412 **6.8 maxResourceSize**

413 This property SHOULD be returned for the service and MAY be returned for any targeted collection
414 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.

415 *Table 13: MaxResourceSizeType fields*

416 **6.9 minDateTime**

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

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

419 *Table 14: MinDateTimeType fields*

420 **6.10 principalHome**

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

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

423 *Table 15: PrincipalHomeType fields*

424 **6.11 resourceDescription**

425 Provides some descriptive text for the targeted collection.

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

426 *Table 16: ResourceDescriptionType fields*

427 **6.12 resourceOwner**

428 This property SHOULD be returned for any targeted resource.

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

429 *Table 17: ResourceownerType fields*

430 **6.13 resourceTimezoneId**

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

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

433 *Table 18: ResourceTimezoneIdType fields*

434 **6.14 resourceType**

435 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.

436 *Table 19: ResourceType fields*

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

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

438 *Table 20: XresourceType fields*

439 6.15 supportedCalendarComponentSet

440 This property identifies which component types the service is prepared to store. The allowable
441 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.

442 *Table 21: SupportedCalendarComponentSetType fields*

443 6.16 supportedFeatures

444 This property SHOULD be returned for the service and MAY be returned for any targeted collection
445 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.

446 *Table 22: SupportedFeaturesType fields*

447 6.17 timezoneServer

448 This property SHOULD be returned for the service and MAY be returned for any targeted collection
449 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.

450 Table 23: TimezoneServerType fields

451 6.18 CalWS:privilege-set XML element

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

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

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

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

- 458 • CalWS: Read - current principal has read access
 - 459 • CalWS: Write - current principal has write access
- ```
460 <calws:privilege-set>
461 <calws:privilege><calws:read></calws:privilege>
462 <calws:privilege><calws:write></calws:privilege>
463 </calws:privilege-set>
```

## 7 Retrieving Collection and Service Properties

464

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

467 The service properties define the global limits and defaults. Any properties defined on collections within  
468 the service hierarchy override those service defaults. The service may choose to prevent such overriding  
469 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. |

470 Table 24: `GetPropertiesType` fields

| Field | Type                                           | #    | ? | Description                                              |
|-------|------------------------------------------------|------|---|----------------------------------------------------------|
| href  | string                                         | 1    | Y | Identify the target of the request. "/" for the service. |
| ?     | <code>GetPropertiesBaseProperty</code><br>Type | 0..n | C | 0 or more properties of the targeted resource            |

471 Table 25: `GetPropertiesResponseType` fields

### 7.1 Example - retrieving server properties:

472

```
473 >>Request
474
475 <?xml version="1.0" encoding="UTF-8"?>
476 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
477 <SOAP-ENV:Header/>
478 <SOAP-ENV:Body>
479 <ns2:getProperties xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
480 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
481 <ns2:href/></ns2:href>
482 </ns2:getProperties>
483 </SOAP-ENV:Body>
484 </SOAP-ENV:Envelope>
485
486 >>Response
487
488 <?xml version="1.0" encoding="UTF-8"?>
489 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
490 <SOAP-ENV:Header />
491 <SOAP-ENV:Body>
492 <ns2:getPropertiesResponse
493 xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
494 xmlns:ns4="urn:ietf:params:xml:ns:icalendar-2.0"
495 id="0">
496 <ns2:href/></ns2:href>
497 <ns2:lastModifiedDate>
498 <ns2:dateTime>2012-01-04T18:21:14Z</ns2:dateTime>
499 </ns2:lastModifiedDate>
500 <ns2:supportedCalendarComponentSet>
501 <ns4:vevent />
502 <ns4:vtodo />
503 <ns4:vavailability />
504 </ns2:supportedCalendarComponentSet>
505 <ns2:resourceType>
506 <ns2:collection />
507 </ns2:resourceType>
508 <ns2:supportedFeatures>
509 <ns2:calendarAccessFeature />
510 </ns2:supportedFeatures>
```

```
511 <ns2:maxInstances>
512 <ns2:integer>1000</ns2:integer>
513 </ns2:maxInstances>
514 <ns2:maxResourceSize>
515 <ns2:integer>1000000</ns2:integer>
516 </ns2:maxResourceSize>
517 </ns2:getPropertiesResponse>
518 </SOAP-ENV:Body>
519 </SOAP-ENV:Envelope>
520
521
```

## 8 Creating Calendar Object Resources

522

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

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

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

528 *Table 26: AddItem fields*

529 The service will respond with an AddItemResponseType giving either the href and change token of the  
530 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                  |

531 *Table 27: AddItemResponseType additional fields*

### 8.1 Preconditions for Calendar Object Creation

532

- 533 • **CalWS:target-exists:** The entity already exists.
- 534 • **CalWS:not-calendar-data:** The resource submitted MUST be a supported media type (i.e.,  
535 iCalendar) for calendar object resources;
- 536 • **CalWS:invalid-calendar-data:** The resource submitted MUST be valid data for the media type  
537 being specified (i.e., MUST contain valid iCalendar data);
- 538 • **CalWS:invalid-calendar-object-resource:** The resource submitted in the request MUST obey all  
539 restrictions specified in Calendar Object Resources (e.g., calendar object resources MUST NOT  
540 contain more than one type of calendar component, calendar object resources MUST NOT specify  
541 the iCalendar METHOD property, etc.);
- 542 • **CalWS:unsupported-calendar-component:** The resource submitted in the request MUST contain  
543 a type of calendar component that is supported in the targeted calendar collection;
- 544 • **CalWS:uid-conflict:** The resource submitted in the request MUST NOT specify an iCalendar UID  
545 property value already in use in the targeted calendar collection or overwrite an existing calendar  
546 object resource with one that has a different UID property value. Servers SHOULD report the URL  
547 of the resource that is already making use of the same UID property value in the CalWS:href  
548 element  
549 <!ELEMENT uid-conflict (CalWS:href)>
- 550 • **CalWS:exceeds-max-resource-size:** The resource submitted in the request MUST have an octet  
551 size less than or equal to the value of the CalDAV:max-resource-size property value on the  
552 calendar collection where the resource will be stored;
- 553 • **CalWS:before-min-date-time:** The resource submitted in the request MUST have all of its  
554 iCalendar DATE or DATE-TIME property values (for each recurring instance) greater than or equal  
555 to the value of the CalDAV:min-date-time property value on the calendar collection where the  
556 resource will be stored;
- 557 • **CalWS:after-max-date-time:** The resource submitted in the request MUST have all of its iCalendar  
558 DATE or DATE-TIME property values (for each recurring instance) less than the value of the  
559 CalDAV:max-date-time property value on the calendar collection where the resource will be stored;

- 560 • **CalWS:too-many-instances:** The resource submitted in the request MUST generate a number of  
561 recurring instances less than or equal to the value of the CalDAV: max-instances property value on  
562 the calendar collection where the resource will be stored;
- 563 • **CalWS:too-many-attendees-per-instance:** The resource submitted in the request MUST have a  
564 number of ATTENDEE properties on any one instance less than or equal to the value of the  
565 CalDAV:max-attendees-per-instance property value on the calendar collection where the resource  
566 will be stored;

## 567 8.2 Example - successful addItem:

```

568 >>Request
569
570 <?xml version="1.0" encoding="UTF-8"?>
571 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
572 <SOAP-ENV:Header/>
573 <SOAP-ENV:Body>
574 <ns2:addItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
575 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
576 <ns2:href>/user/douglm/calendar</ns2:href>
577 <ns3:icalendar>
578 <ns3:vcalendar>
579 <ns3:components>
580 <ns3:vevent>
581 <ns3:properties>
582 <ns3:uid>
583 <ns3:text>1302064354993</ns3:text>
584 </ns3:uid>
585 <ns3:summary>
586 <ns3:text>try this</ns3:text>
587 </ns3:summary>
588 <ns3:dtstart>
589 <ns3:date-time>20110406T150000Z</ns3:date-time>
590 </ns3:dtstart>
591 <ns3:dtend>
592 <ns3:date-time>20110406T160000Z</ns3:date-time>
593 </ns3:dtend>
594 </ns3:properties>
595 </ns3:vevent>
596 </ns3:components>
597 </ns3:vcalendar>
598 </ns3:icalendar>
599 </ns2:addItem>
600 </SOAP-ENV:Body>
601 </SOAP-ENV:Envelope>
602
603 >>Response
604
605 <?xml version="1.0" encoding="UTF-8"?>
606 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
607 <SOAP-ENV:Header/>
608 <SOAP-ENV:Body>
609 <ns2:addItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
610 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
611 <ns2:status>OK</ns2:status>
612 <ns2:href>/user/douglm/calendar/1302064354993.ics</ns2:href>
613 <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
614 </ns2:addItemResponse>
615 </SOAP-ENV:Body>
616 </SOAP-ENV:Envelope>

```

## 617 8.3 Example - unsuccessful addItem:

618 TBD



619

## 9 Retrieving resources

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

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

623 *Table 28: FetchItemType fields*

624 The service will respond with a FetchItemResponseType containing either the change token, its href and  
625 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                      |

626 *Table 29: FetchItemResponseType additional fields*

### 9.1 Example - successful fetchItem:

```

628 >>Request
629
630 <?xml version="1.0" encoding="UTF-8"?>
631 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
632 <SOAP-ENV:Header/>
633 <SOAP-ENV:Body>
634 <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
635 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
636 <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
637 </ns2:fetchItem>
638 </SOAP-ENV:Body>
639 </SOAP-ENV:Envelope>
640
641 >>Response
642
643 <?xml version="1.0" encoding="UTF-8"?>
644 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
645 <SOAP-ENV:Header/>
646 <SOAP-ENV:Body>
647 <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
648 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
649 <ns2:status>OK</ns2:status>
650 <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
651 <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
652 <ns3:icalendar>
653 <ns3:vcalendar>
654 <ns3:properties>
655 <ns3:prodid>
656 <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
657 </ns3:prodid>
658 <ns3:version>
659 <ns3:text>2.0</ns3:text>
660 </ns3:version>
661 </ns3:properties>
662 <ns3:components>
663 <ns3:vevent>
664 <ns3:properties>

```

```

665 <ns3:created>
666 <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
667 </ns3:created>
668 <ns3:dtend>
669 <ns3:date-time>20110406T160000Z</ns3:date-time>
670 </ns3:dtend>
671 <ns3:dtstamp>
672 <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
673 </ns3:dtstamp>
674 <ns3:dtstart>
675 <ns3:date-time>20110406T150000Z</ns3:date-time>
676 </ns3:dtstart>
677 <ns3:last-modified>
678 <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
679 </ns3:last-modified>
680 <ns3:summary>
681 <ns3:text>try this</ns3:text>
682 </ns3:summary>
683 <ns3:uid>
684 <ns3:text>1302105461170</ns3:text>
685 </ns3:uid>
686 </ns3:properties>
687 </ns3:vevent>
688 </ns3:components>
689 </ns3:vcalendar>
690 </ns3:icalendar>
691 </ns2:fetchItemResponse>
692 </SOAP-ENV:Body>
693 </SOAP-ENV:Envelope>

```

## 9.2 Example - unsuccessful fetchItem:

```

694
695 >>Request
696
697 <?xml version="1.0" encoding="UTF-8"?>
698 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
699 <SOAP-ENV:Header/>
700 <SOAP-ENV:Body>
701 <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
702 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
703 <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
704 </ns2:fetchItem>
705 </SOAP-ENV:Body>
706 </SOAP-ENV:Envelope>
707
708 >>Response
709
710 <?xml version="1.0" encoding="UTF-8"?>
711 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
712 <SOAP-ENV:Header/>
713 <SOAP-ENV:Body>
714 <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
715 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
716 <ns2:status>Error</ns2:status>
717 <ns2:errorResponse>
718 <ns2:targetDoesNotExist/>
719 </ns2:errorResponse>
720 </ns2:fetchItemResponse>
721 </SOAP-ENV:Body>
722 </SOAP-ENV:Envelope>

```

723

## 10 Updating resources

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

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

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

```
731 <icalendar>
732 <vcalendar>
733 <components>
734 <vevent>
735 <properties>
736 <uid>
737 <text>1302064354993-a</text>
738 </uid>
739 <summary>
740 <text>try this</text>
741 </summary>
742 <dtstart>
743 <date-time>2011-07-18T15:00:00Z</date-time>
744 </dtstart>
745 <dtend>
746 <date-time>2011-07-18T16:00:00Z</date-time>
747 </dtend>
748 </properties>
749 </vevent>
750 </components>
751 </vcalendar>
752 </icalendar>
```

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

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

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

- 761 • Remove - components, properties or parameters
- 762 • Add - components, properties or parameters
- 763 • Change - property or parameter values

764 Removals MUST be processed ahead of additions

765 Preconditions as specified in Preconditions for Calendar Object Creation are applicable. The response  
766 will indicate success or failure of the update. If the change token value does not match that held by the  
767 service a mismatchedChangeToken error status will be returned. The client should re-fetch the entity to  
768 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                               |

769 *Table 30: UpdateItem fields*

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

| 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                       |

773 *Table 31: ComponentsSelectionType fields*

774 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                      |

775 *Table 32: PropertiesSelectionType fields*

776 To complete that pattern there is also a ParametersSelectionType used to select property parameters for  
777 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                       |

778 *Table 33: ParametersSelectionType fields*

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

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

781 *Table 34: ComponentReferenceType fields*

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

782 *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. |

783 *Table 36: ParameterReferenceType fields*

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

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

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

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

| Field                              | Type                                         | #    | ? | Description                                    |
|------------------------------------|----------------------------------------------|------|---|------------------------------------------------|
| Any valid iCalendar component name | xcal:VcalendarType<br>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                |

796 *Table 37: ComponentSelectionType fields*

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

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

801 The optional change element allows a change to the value of the property. The new value is specified by  
802 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                         |

803 *Table 38: PropertySelectionType fields*

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

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

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

808 *Table 39: ParameterSelectionType fields*

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

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

811 Table 40: UpdateItemResponseType additional fields

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

## 813 10.1 Change tokens and concurrent updates

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

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

## 825 10.2 Example - successful update:

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

```

827 <ns3:icalendar>
828 <ns3:vcalendar>
829 <ns3:components>
830 <ns3:vevent>
831 <ns3:properties>
832 <ns3:uid>
833 <ns3:text>1302064354993-a</ns3:text>
834 </ns3:uid>
835 <ns3:summary>
836 <ns3:text>try this</ns3:text>
837 </ns3:summary>
838 <ns3:dtstart>
839 <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
840 </ns3:dtstart>
841 <ns3:dtend>
842 <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
843 </ns3:dtend>

```

```

844 </ns3:properties>
845 </ns3:vevent>
846 </ns3:components>
847 </ns3:vcalendar>
848 </ns3:icalendar>

```

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

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

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

```

855 >>Request
856
857 <?xml version="1.0" encoding="UTF-8"?>
858 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
859 <SOAP-ENV:Header/>
860 <SOAP-ENV:Body>
861 <ns2:updateItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
862 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
863 <ns2:href>/user/douglm/calendar/1302064354993-a.ics</ns2:href>
864 <ns2:changeToken>"20110802T032608Z-0" null</ns2:changeToken>
865 <ns2:select>
866 <ns3:vcalendar/>
867 <ns2:components>
868 <ns2:component>
869 <ns3:vevent>
870 <ns3:properties>
871 <ns3:uid>
872 <ns3:text>1302064354993-a</ns3:text>
873 </ns3:uid>
874 </ns3:properties>
875 </ns3:vevent>
876 <ns2:properties>
877 <ns2:property>
878 <ns3:dtstart>
879 <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
880 </ns3:dtstart>
881 <ns2:parameters>
882 <ns2:add>
883 <ns3:tzid>
884 <ns3:text>America/New_York</ns3:text>
885 </ns3:tzid>
886 </ns2:add>
887 </ns2:parameters>
888 <ns2:change>
889 <ns3:dtstart>
890 <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
891 </ns3:dtstart>
892 </ns2:change>
893 </ns2:property>
894 <ns2:property>
895 <ns3:summary>
896 <ns3:text>try this</ns3:text>
897 </ns3:summary>
898 <ns2:change>
899 <ns3:summary>
900 <ns3:text>A changed summary - again and again and again</ns3:text>
901 </ns3:summary>
902 </ns2:change>
903 </ns2:property>
904 <ns2:add>
905 <ns3:categories>
906 <ns3:text>newcategory-2</ns3:text>
907 <ns3:text>resources</ns3:text>
908 <ns3:text>paper</ns3:text>
909 </ns3:categories>
910 </ns2:add>

```

```

911 </ns2:properties>
912 </ns2:component>
913 </ns2:components>
914 </ns2:select>
915 </ns2:updateItem>
916 </SOAP-ENV:Body>
917 </SOAP-ENV:Envelope>
918
919 >>Response
920
921 <?xml version="1.0" encoding="UTF-8"?>
922 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
923 <SOAP-ENV:Header/>
924 <SOAP-ENV:Body>
925 <ns2:updateItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
926 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
927 id="0">
928 <ns2:status>OK</ns2:status>
929 </ns2:updateItemResponse>
930 </SOAP-ENV:Body>
931 </SOAP-ENV:Envelope>

```

### 932 10.3 Other updates:

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

- 935 • Addition of properties
- 936 • Removal of properties
- 937 • Addition of parameters to properties
- 938 • Removal of parameters from properties
- 939 • Changing parameter values.

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

```

943 <ns2:properties>
944 <ns2:property>
945 <ns3:dtstart>
946 <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
947 </ns3:dtstart>
948 <ns2:parameters>
949 <ns2:add>
950 <ns3:tzid>
951 <ns3:text>America/New_York</ns3:text>
952 </ns3:tzid>
953 </ns2:add>
954 </ns2:parameters>
955 <ns2:change>
956 <ns3:dtstart>
957 <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
958 </ns3:dtstart>
959 </ns2:change>
960 </ns2:property>
961 </ns2:properties>

```

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

```

963 <ns2:properties>
964 <ns2:add>
965 <ns3:categories>
966 <ns3:text>paper</ns3:text>
967 </ns3:categories>
968 </ns2:add>
969 <ns2:add>
970 <ns3:categories>
971 <ns3:text>resources</ns3:text>
972 </ns3:categories>

```



```
973 </ns2:add>
974 </ns2:properties>
```

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

```
976 <ns2:properties>
977 <ns2:remove>
978 <ns3:dtend>
979 <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
980 </ns3:dtend>
981 </ns2:remove>
982 <ns2:add>
983 <ns3:duration>
984 <ns3:duration>PT1H</ns3:duration>
985 </ns3:duration>
986 </ns2:add>
987 </ns2:properties>
```

988 In this example we change the dtstart timezone identifier.

```
989 <ns2:properties>
990 <ns2:property>
991 <ns3:dtstart>
992 <ns3:parameters>
993 <ns3:tzid>
994 <ns3:text>America/New_York</ns3:text>
995 </ns3:tzid>
996 </ns3:parameters>
997 <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
998 </ns3:dtstart>
999 <ns2:parameters>
1000 <ns2:parameter>
1001 <ns3:tzid>
1002 <ns3:text>America/New_York</ns3:text>
1003 </ns3:tzid>
1004 <ns2:change>
1005 <ns3:tzid>
1006 <ns3:text>America/Montreal</ns3:text>
1007 </ns3:tzid>
1008 </ns2:change>
1009 </ns2:parameter>
1010 </ns2:parameters>
1011 </ns2:property>
1012 </ns2:properties>
```

1013

## 1014 10.4 Creating an update message.

1015 The update can be created in many ways but the most common approach is to build the update while  
1016 modifications take place or to create one as the result of comparing old and new versions. It appears that  
1017 comparing XML for differences is difficult. However, we can take advantage of the structure of  
1018 calendaring entities to simplify the process. There are implementations available which take the diff  
1019 approach to producing an update stream.

1020 There are some special cases to consider when comparing. Some properties are multi-valued and may  
1021 themselves appear more than once. There is no semantic information implied by any grouping though  
1022 parameters may need to be taken into account. These properties need to be normalized before  
1023 comparison and when updating them we produce a change which treats each value as a single property.

1024 These properties are

- 1025 • categories
- 1026 • exdate
- 1027 • freebusy
- 1028 • rdate

1029 This normalization can take place before comparison.

1030 Some properties are multi-valued and may only appear once. At the moment the only standard property  
1031 is resource which may take a comma separated list. This should be treated as a single multi-valued  
1032 property when comparing. The order is unimportant. Sorting the values may help.

1033 Some properties may appear multiple times, for example comment. Comparison should take account of  
1034 parameters. Ordering all properties appropriately allows for relatively simple comparison.

1035

## 11 Deletion of resources

1036 Deletion of calendar object resources is carried out by using a CalWS-SOAP deleteItem request with an  
 1037 href specifying the entity to be deleted. The deleteItem request is not valid when the href specifies a  
 1038 collection.

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

1039 *Table 41: DeleteItem fields*

1040 The service will respond with a DeleteItemResponseType containing the status and a possible error  
 1041 response. There are no additional elements.

### 11.1 Example - successful deleteItem:

```

1043 >>Request
1044
1045 <?xml version="1.0" encoding="UTF-8"?>
1046 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1047 <SOAP-ENV:Header/>
1048 <SOAP-ENV:Body>
1049 <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1050 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1051 <ns2:href>/user/douglm/calendar/1302620814655.ics</ns2:href>
1052 </ns2:deleteItem>
1053 </SOAP-ENV:Body>
1054 </SOAP-ENV:Envelope>
1055
1056 >>Response
1057
1058 <?xml version="1.0" encoding="UTF-8"?>
1059 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1060 <SOAP-ENV:Header/>
1061 <SOAP-ENV:Body>
1062 <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1063 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1064 <ns2:status>OK</ns2:status>
1065 </ns2:deleteItemResponse>
1066 </SOAP-ENV:Body>
1067 </SOAP-ENV:Envelope>

```

### 11.2 Example - unsuccessful deleteItem:

```

1068 >>Request
1069
1070 <?xml version="1.0" encoding="UTF-8"?>
1071 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1072 <SOAP-ENV:Header/>
1073 <SOAP-ENV:Body>
1074 <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1075 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1076 <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
1077 </ns2:deleteItem>
1078 </SOAP-ENV:Body>
1079 </SOAP-ENV:Envelope>
1080
1081 >>Response
1082
1083 <?xml version="1.0" encoding="UTF-8"?>
1084 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1085 <SOAP-ENV:Header/>
1086 <SOAP-ENV:Body>

```

```
1088 <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1089 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1090 <ns2:status>Error</ns2:status>
1091 <ns2:errorResponse>
1092 <ns2:targetDoesNotExist/>
1093 </ns2:errorResponse>
1094 </ns2:deleteItemResponse>
1095 </SOAP-ENV:Body>
1096 </SOAP-ENV:Envelope>
```

1097

## 12 Querying calendar resources

1098 Querying provides a mechanism by which information can be obtained from the service through possibly  
1099 complex queries. A skeleton icalendar entity can be provided to limit the amount of information returned  
1100 to the client. A query takes the parts

- 1101 • Limitations on the data returned
- 1102 • Selection of the data
- 1103 • Optional timezone id for floating time calculations.

### 1104 12.1 Calendar Query common types

1105 The UTCTimeRangeType is used in a number of places to define a time range within which components  
1106 must appear or property values must lie. The values are UTC time-date, the start is inclusive and the end  
1107 is exclusive.

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

1108 *Table 42: UTCTimeRangeType elements*

1109 The TextMatchType is used to match text values in properties and parameters. The collation attribute  
1110 species a collation as defined in [RFC 4790].

1111 Servers are REQUIRED to support the ";ascii-casemap" and ";octet" collations which provide a basic  
1112 case insensitive and case sensitive match respectively.

1113 Elements of this type take a string value which is matched according to the attributes.

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

1114 *Table 43: TextMatchType attributes*

### 1115 12.2 CompFilterType

1116 This type defines a search query for the calendar query operation. It specifies the component types to  
1117 return, absence tests or basic matching operations on properties and time ranges.

1118 The top level comp-filter element (which must match a vcalendar component may contain zero or more  
1119 comp-filter elements to match events, tasks or other contained components. These in turn may contain  
1120 further nested comp-filter elements to match further levels of nested components.

1121 Each may also contain prop-filter elements to test for the absence of properties or to match values.

1122 Only logical conjunctions are supported, that is, all elements of a comp-filter must match for the  
1123 expression to match.

| Field              | Type                   | #    | ? | Description                                                                                                            |
|--------------------|------------------------|------|---|------------------------------------------------------------------------------------------------------------------------|
| anyComp            | AnyCompType            | 0..1 | C | One of anyComp, vcalendar or a BaseComponentType must be supplied.<br>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.<br>"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                                                                                     |

1124 Table 44: CompFilterType elements

## 1125 12.3 PropFilterType

1126 The prop-filter element may test for the absence of a property or match values or specify zero or more  
1127 ParamFilterType elements to match against parameters.

1128 Only logical conjunctions are supported, that is, all elements must match for the full expression to match.

| 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.<br>"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                                                                 |

1129 Table 45: PropFilterType elements

## 1130 12.4 ParamFilterType

1131 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 |                                        |

1132 Table 46: ParamFilterType elements

## 1133 12.5 CalendarQueryType elements

| Field              | Type                   | #    | ? | Description                                                                                                                                                                                                                                                                                                           |
|--------------------|------------------------|------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| href               | string                 | 1    | Y | Identify the target of the request. "/" for the service.                                                                                                                                                                                                                                                              |
| allprop            | empty                  | 0..1 | N | If present specifies all properties should be returned<br>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.<br>Either expand or limitRecurrenceSet may be specified but not both.<br>If specified recurring events are expanded and limited to the supplied time-range. All events times are converted to UTC.<br>This option allows for simplified event handling for certain classes of client. |
| limitRecurrenceSet | LimitRecurrenceSetType | 0..1 | N | A subclass of UTCTimeRangeType.<br>Either expand or limitRecurrenceSet may be specified but not both.<br>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                                                                                                                                                                                                                                                                                       |

1134 Table 47: CalendarQueryType elements

## 1135 12.6 Specifying data to be returned

1136 This is achieved by specifying one of the following

- 1137 • allprop: return all properties and calendar data. (some properties are specified as not being part of
- 1138 the allprop set so are not returned)
- 1139 • Set the icalendar element. This is an icalendar valueless pattern entity which provides a map of the
- 1140 components and properties to be returned. Neither the pattern nor the returned result need to be
- 1141 valid icalendar entities in that required properties may be absent if unselected.

## 1142 12.7 Pre/postconditions for calendar queries

1143 The preconditions as defined in [RFC 4791] Section 7.8 apply here. CalDav errors may be reported by

1144 the service when preconditions or postconditions are violated.

## 1145 12.8 Time range limited queries.

1146 Time-range limited retrieval has some special characteristics. The simplest case is a single event or task

1147 which overlaps the requested time-period. Recurring items and other components such as alarms

1148 complicate the picture.

## 1149 12.9 Example: time range limited retrieval

1150 This example shows the time-range limited retrieval from a calendar which results in 2 events, one a

1151 recurring event and one a simple non-recurring event.

```
1152 >> Request <<
1153 <?xml version="1.0" encoding="UTF-8"?>
1154 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1155 <SOAP-ENV:Header/>
1156 <SOAP-ENV:Body>
1157 <ns2:calendarQuery xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1158 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1159 <ns2:href>/user/douglm/calendar</ns2:href>
1160 <ns3:icalendar>
1161 <ns3:vcalendar>
1162 <ns3:components>
1163 <ns3:vevent>
1164 <ns3:properties>
1165 <ns3:summary/>
1166 <ns3:dtstart/>
1167 <ns3:dtend/>
1168 <ns3:duration/>
1169 <ns3:uid/>
1170 <ns3:recurrence-id/>
1171 <ns3:rrule/>
1172 <ns3:rdate/>
1173 <ns3:exdate/>
1174 </ns3:properties>
1175 </ns3:vevent>
1176 </ns3:components>
1177 </ns3:vcalendar>
1178 </ns3:icalendar>
1179 <ns2:filter>
1180 <ns2:compFilter test="anyof">
1181 <ns3:vcalendar />
1182 <ns2:compFilter>
1183 <ns3:vevent />
1184 <ns2:time-range end="20110430T040000Z" start="20110401T040000Z"/>
1185 </ns2:compFilter>
1186 </ns2:filter>
1187 </ns2:calendarQuery>
```



```

1189 </SOAP-ENV:Body>
1190 </SOAP-ENV:Envelope>
1191
1192 >> Response <<
1193
1194 <?xml version="1.0" encoding="UTF-8"?>
1195 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1196 <SOAP-ENV:Header/>
1197 <SOAP-ENV:Body>
1198 <ns2:calendarQueryResponse
1199 xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1200 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1201 <ns2:status>OK</ns2:status>
1202 <ns2:response>
1203 <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
1204 <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
1205 <ns2:propstat>
1206 <ns2:prop>
1207 <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1208 <ns3:icalendar>
1209 <ns3:vcalendar>
1210 <ns3:properties>
1211 <ns3:prodid>
1212 <ns3:text>//Bedework.org//BedWork V3.7//EN</ns3:text>
1213 </ns3:prodid>
1214 <ns3:version>
1215 <ns3:text>2.0</ns3:text>
1216 </ns3:version>
1217 </ns3:properties>
1218 <ns3:components>
1219 <ns3:vevent>
1220 <ns3:properties>
1221 <ns3:dtend>
1222 <ns3:date-time>20110406T160000Z</ns3:date-time>
1223 </ns3:dtend>
1224 <ns3:dtstart>
1225 <ns3:date-time>20110406T150000Z</ns3:date-time>
1226 </ns3:dtstart>
1227 <ns3:summary>
1228 <ns3:text>try this</ns3:text>
1229 </ns3:summary>
1230 <ns3:uid>
1231 <ns3:text>1302105461170</ns3:text>
1232 </ns3:uid>
1233 </ns3:properties>
1234 </ns3:vevent>
1235 </ns3:components>
1236 </ns3:vcalendar>
1237 </ns3:icalendar>
1238 </ns2:calendar-data>
1239 </ns2:prop>
1240 <ns2:status>OK</ns2:status>
1241 </ns2:propstat>
1242 </ns2:response>
1243 </ns2:response>
1244 <ns2:href>/user/douglm/calendar/CAL-00f1fc61-2f021bca-012f-022947f8-
1245 00000006.ics</ns2:href>
1246 <ns2:changeToken>"20110405T140920Z-0"</ns2:changeToken>
1247 <ns2:propstat>
1248 <ns2:prop>
1249 <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1250 <ns3:icalendar>
1251 <ns3:vcalendar>
1252 <ns3:properties>
1253 <ns3:prodid>
1254 <ns3:text>//Bedework.org//BedWork V3.7//EN</ns3:text>
1255 </ns3:prodid>
1256 <ns3:version>
1257 <ns3:text>2.0</ns3:text>
1258 </ns3:version>

```

```

1259 </ns3:properties>
1260 </ns3:components>
1261 <ns3:vevent>
1262 <ns3:properties>
1263 <ns3:duration>
1264 <ns3:duration>PT1H</ns3:duration>
1265 </ns3:duration>
1266 <ns3:dtstart>
1267 <ns3:parameters>
1268 <ns3:tzid>
1269 <ns3:text>America/New_York</ns3:text>
1270 </ns3:tzid>
1271 </ns3:parameters>
1272 <ns3:date-time>20110412T110000</ns3:date-time>
1273 </ns3:dtstart>
1274 <ns3:summary>
1275 <ns3:text>Test recurring event</ns3:text>
1276 </ns3:summary>
1277 <ns3:uid>
1278 <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1279 00000006demobedework@mysite.edu</ns3:text>
1280 </ns3:uid>
1281 <ns3:rrule>
1282 <ns3:recur>
1283 <ns3:freq>WEEKLY</ns3:freq>
1284 <ns3:count>2</ns3:count>
1285 <ns3:interval>1</ns3:interval>
1286 </ns3:recur>
1287 </ns3:rrule>
1288 </ns3:properties>
1289 </ns3:vevent>
1290 <ns3:vevent>
1291 <ns3:properties>
1292 <ns3:recurrence-id>
1293 <ns3:parameters>
1294 <ns3:tzid>
1295 <ns3:text>America/New_York</ns3:text>
1296 </ns3:tzid>
1297 </ns3:parameters>
1298 <ns3:date-time>20110419T150000Z</ns3:date-time>
1299 </ns3:recurrence-id>
1300 <ns3:duration>
1301 <ns3:duration>PT1H</ns3:duration>
1302 </ns3:duration>
1303 <ns3:dtstart>
1304 <ns3:parameters>
1305 <ns3:tzid>
1306 <ns3:text>America/New_York</ns3:text>
1307 </ns3:tzid>
1308 </ns3:parameters>
1309 <ns3:date-time>20110419T120000</ns3:date-time>
1310 </ns3:dtstart>
1311 <ns3:summary>
1312 <ns3:text>Test recurring event</ns3:text>
1313 </ns3:summary>
1314 <ns3:uid>
1315 <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1316 00000006demobedework@mysite.edu</ns3:text>
1317 </ns3:uid>
1318 </ns3:properties>
1319 </ns3:vevent>
1320 </ns3:components>
1321 </ns3:vcalendar>
1322 </ns3:icalendar>
1323 </ns2:calendar-data>
1324 </ns2:prop>
1325 <ns2:status>OK</ns2:status>
1326 </ns2:propstat>
1327 </ns2:response>
1328 </ns2:calendarQueryResponse>

```

1329  
1330  
1331

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

---

## 1332 13 Free-busy queries

1333 Freebusy queries are used to obtain freebusy information for a principal. The result contains information  
1334 only for events to which the current principal has sufficient access and may be affected by components  
1335 and rules available only to the server (for instance office hours availability).

1336 These queries are carried out by using a CalWS-SOAP freebusyReport request with an href specifying a  
1337 principal. The freebusyReport request is not valid when the href specifies any entity other than a principal.

1338 The query follows the specification defined in [FreeBusy Read URL] with certain limitations. As an  
1339 authenticated user to the CalWS service scheduling read-freebusy privileges must have been granted. As  
1340 an unauthenticated user equivalent access must have been granted to unauthenticated users.

1341 Freebusy information is returned by default as xcalendar vfreebusy components, as defined by [draft-  
1342 xcal]. Such a component is not meant to conform to the requirements of VFREEBUSY components in  
1343 [RFC 5546]. The VFREEBUSY component SHOULD conform to section "4.6.4 Free/Busy Component" of  
1344 [RFC 5545]. A client SHOULD ignore the ORGANIZER field.

1345 Since a Freebusy query can only refer to a single user, a client will already know how to match the result  
1346 component to a user. A server MUST only return a single vfreebusy component.

### 1347 13.1 Element values

1348 Three values are provided: href; start; end. Only the href is required. The start and end are in XML UTC  
1349 date/time format and are interpreted as follows:

#### 1350 start

1351 **Default:** If omitted the default value is left up to the server. It may be the current day, start of the  
1352 current month, etc.

1353 **Description:** Specifies the start date for the Freebusy data. The server is free to ignore this value  
1354 and return data in any time range. The client must check the data for the returned time range.

1355 **Format:** An XML UTC date-time

1356 **Example:**

1357 `2011-12-01T10:15:00Z`

1358 **Notes:** Specifying only a start date/time without specifying an end-date/time or period should be  
1359 interpreted as in [RFC 5545]. The effective period should cover the remainder of that day.

#### 1360 end

1361 **Default:** Same as start

1362 **Description:** Specifies the end date for the Freebusy data. The server is free to ignore this value.

1363 **Format:** Same as start

1364 **Example:** Same as start

1365 The server is free to ignore the start, end and period parameters. It is recommended that the server  
1366 return at least 6 weeks of data from the current day.

1367 A client MUST check the time range in the response as a server may return a different time range than  
1368 the requested range.

## 13.2 Examples

1370 The following is an unsuccessful request targeting an invalid resource.

```

1371 >> Request <<
1372
1373 <?xml version="1.0" encoding="UTF-8"?>
1374 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1375 <SOAP-ENV:Header/>
1376 <SOAP-ENV:Body>
1377 <ns2:freebusyReport
1378 xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1379 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1380 <ns2:href>/user/douglm/calendar</ns2:href>
1381 <ns2:time-range>
1382 <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1383 <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1384 </ns2:time-range>
1385 </ns2:freebusyReport>
1386 </SOAP-ENV:Body>
1387 </SOAP-ENV:Envelope>
1388
1389 >> Response <<
1390
1391 <?xml version="1.0" encoding="UTF-8"?>
1392 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1393 <SOAP-ENV:Header/>
1394 <SOAP-ENV:Body>
1395 <ns2:freebusyReportResponse
1396 xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1397 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1398 <ns2:status>Error</ns2:status>
1399 <ns2:message>Only principal href supported</ns2:message>
1400 </ns2:freebusyReportResponse>
1401 </SOAP-ENV:Body>
1402 </SOAP-ENV:Envelope>

```

1403 The following is an example of a request to retrieve Freebusy data for a user:

```

1404 >> Request <<
1405
1406 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1407 <SOAP-ENV:Header/>
1408 <SOAP-ENV:Body>
1409 <ns2:freebusyReport
1410 xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1411 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1412 <ns2:href>/principals/users/douglm</ns2:href>
1413 <ns2:time-range>
1414 <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1415 <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1416 </ns2:time-range>
1417 </ns2:freebusyReport>
1418 </SOAP-ENV:Body>
1419 </SOAP-ENV:Envelope>
1420
1421 >> Response <<
1422
1423 <?xml version="1.0" encoding="UTF-8"?>
1424 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1425 <SOAP-ENV:Header/>
1426 <SOAP-ENV:Body>
1427 <ns2:freebusyReportResponse
1428 xmlns:ns2="http://docs.oasis-open.org/ns/wscal/calws-soap"
1429 xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1430 <ns2:status>OK</ns2:status>
1431 <ns3:icalendar>
1432 <ns3:vcalendar>
1433 <ns3:properties>
1434 <ns3:prodid>

```

```

1435 <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1436 </ns3:prodid>
1437 <ns3:version>
1438 <ns3:text>2.0</ns3:text>
1439 </ns3:version>
1440 </ns3:properties>
1441 <ns3:components>
1442 <ns3:vfreebusy>
1443 <ns3:properties>
1444 <ns3:attendee>
1445 <ns3:parameters>
1446 <ns3:partstat>
1447 <ns3:text>NEEDS-ACTION</ns3:text>
1448 </ns3:partstat>
1449 </ns3:parameters>
1450 <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1451 </ns3:attendee>
1452 <ns3:created>
1453 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1454 </ns3:created>
1455 <ns3:dtend>
1456 <ns3:date-time>2011-04-30T00:00:00Z</ns3:date-time>
1457 </ns3:dtend>
1458 <ns3:dtstamp>
1459 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1460 </ns3:dtstamp>
1461 <ns3:dtstart>
1462 <ns3:date-time>2011-04-01T00:00:00Z</ns3:date-time>
1463 </ns3:dtstart>
1464 <ns3:freebusy>
1465 <ns3:parameters>
1466 <ns3:fbtype>
1467 <ns3:text>BUSY</ns3:text>
1468 </ns3:fbtype>
1469 </ns3:parameters>
1470 <ns3:period>
1471 <ns3:start>2011-04-06T15:00:00Z</ns3:start>
1472 <ns3:end>2011-04-06T16:00:00Z</ns3:end>
1473 </ns3:period>
1474 </ns3:vfreebusy>
1475 <ns3:last-modified>
1476 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1477 </ns3:last-modified>
1478 <ns3:organizer>
1479 <ns3:parameters/>
1480 <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1481 </ns3:organizer>
1482 <ns3:uid>
1483 <ns3:text>2UTDVPZ9H0EQL9QISI44SP5IFPC4N75</ns3:text>
1484 </ns3:uid>
1485 </ns3:properties>
1486 </ns3:vfreebusy>
1487 </ns3:components>
1488 </ns3:vcalendar>
1489 </ns3:icalendar>
1490 </ns2:FreebusyReportResponse>
1491 </SOAP-ENV:Body>
1492 </SOAP-ENV:Envelope>

```

1493

1494

## 14 Multiple operations

1495 Each of the previously described operations acts upon a single entity or resource only. Frequently we  
1496 have the need to update an interconnected set of entities so that we maintain the consistency of the  
1497 structure. This requires an atomic operation which can successfully update all the entities or roll back the  
1498 operation on failure.

1499 The MultiOpType operation provides such a feature. It is essentially a wrapper around any of the other  
1500 operations which guarantees the success of the entire set or a roll back. Using the id attribute for  
1501 requests, each individual response can be located in the result.

1502 The MultiOpType request takes the following elements

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

1503 Table 48: MultiOpType elements

1504 The response type is also simple containing a single element containing all the responses.

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

1505 Table 49: MultiOpResponseType elements

---

1506

## # Conformance

1507 The last numbered section in the specification must be the Conformance section. Conformance  
1508 Statements/Clauses go here.

1509



---

1510

## Appendix A. Acknowledgments

1511 The following individuals have participated in the creation of this specification and are gratefully  
1512 acknowledged

1513 **Participants:**

- 1514     • Cyrus Daboo, Apple

1515 The authors would also like to thank the Calendaring and Scheduling Consortium and the TC-XML  
1516 committee for help with this specification.

1517

1518

1519

1520

1521

---

## Appendix B. Non-Normative Text

1522

## Appendix C. Revision History

| Revision | Date             | Editor      | Changes                                                                                                                                                                                                                                                                             |
|----------|------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 06       | January 3 2012   | M. Douglass | Remove all references to XRD. Define CalWS properties in their place.                                                                                                                                                                                                               |
| 05       | December 15 2011 | M. Douglass | Change example from CalDAV to CalWS                                                                                                                                                                                                                                                 |
| 04       | November 11 2011 | M. Douglass | Updated calendar query to use xcal types instead of names. Assumes a later version of the xcalendar schema to make this possible<br><br>Change references to "etoken" to "changeToken"<br><br>Update the error codes with descriptions and a type per error. Added some new errors. |
| 03       | September 7 2011 | M. Douglass | Add test attribute to calendar query elements.                                                                                                                                                                                                                                      |
| 02       |                  | M. Douglass | Added href to fetch response.<br><br>Change propstat to be extension of BaseResponseType                                                                                                                                                                                            |
| 01       | July 15 2011     | M. Douglass | Added etoken to ensure consistent updates.<br><br>Added a multi op which allows the atomic processing of multiple operations in one request<br><br>Added an id attribute to requests and responses.                                                                                 |
| Initial  | Mar 15 2011      | M. Douglass | Initial publication - a first pass at a rewrite from CalWS-REST                                                                                                                                                                                                                     |