
1 WS-Calendar SOAP-based Services Version 1.0

2 Working Draft 11

3 6 November 2012

4 Technical Committee:

5 OASIS Web Services Calendar (WS-Calendar) TC

6 Chair:

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

8 Editor:

9 Michael Douglass (douglm@rpi.edu) Rensselaer Polytechnic Institute

10 Related work:

11 This specification is related to:

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

16 Abstract:

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

21 Status:

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

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

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

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

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

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

48 Table of Contents

49	1 Introduction.....	5
50	1.1 Terminology.....	5
51	1.2 Normative References.....	5
52	1.3 Non-normative References.....	6
53	1.4 Namespace.....	6
54	2 Issues not addressed by this specification.....	7
55	2.1 Access Control.....	7
56	2.2 Provisioning.....	7
57	2.3 Copy/Move.....	7
58	2.4 Creating Collections.....	7
59	2.5 Retrieving collections.....	7
60	2.6 Setting service and resource properties.....	7
61	3 Basic Calendar Access.....	8
62	3.1 CalWS Glossary.....	8
63	3.1.1 Calendar Object Resource.....	8
64	3.1.2 Uid.....	8
65	3.1.3 Collections.....	8
66	3.1.4 Calendar Collection.....	8
67	3.1.5 Scheduling Calendar Collection.....	8
68	3.1.6 Principal Home.....	8
69	3.1.7 Change token.....	8
70	3.2 Overview of the CalWS protocol.....	9
71	3.2.1 Discovery.....	9
72	3.2.2 Properties.....	9
73	3.2.3 Operations.....	9
74	3.2.4 Calendar Object Resources.....	9
75	3.2.5 Timezone information.....	10
76	3.2.6 Error conditions.....	10
77	3.2.6.1 Example: error with error condition.....	10
78	3.3 CalWs-SOAP Messages.....	10
79	3.3.1 Common Elements and types.....	11
80	3.3.1.1 ErrorCodeType.....	12
81	3.3.1.2 BaseResponseType.....	14
82	3.4 Properties.....	14
83	3.4.1 childCollection.....	15
84	3.4.2 creationDateTime.....	15
85	3.4.3 displayName.....	15
86	3.4.4 lastModifiedDateTime.....	15
87	3.4.5 maxAttendeesPerInstance.....	15
88	3.4.6 maxDateTime.....	16
89	3.4.7 maxInstances.....	16
90	3.4.8 maxResourceSize.....	16
91	3.4.9 minDateTime.....	16

92	3.4.10 principalHome.....	17
93	3.4.11 resourceDescription.....	17
94	3.4.12 resourceOwner.....	17
95	3.4.13 resourceTimezoneId.....	17
96	3.4.14 resourceType.....	17
97	3.4.15 supportedCalendarComponentSet.....	18
98	3.4.16 supportedFeatures.....	18
99	3.4.17 timezoneServer.....	19
100	3.4.18 CalWS:privilege-set XML element.....	19
101	3.5 Retrieving Collection and Service Properties.....	19
102	3.5.1 Example - retrieving server properties:.....	20
103	3.6 Creating Calendar Object Resources.....	21
104	3.6.1 Preconditions for Calendar Object Creation.....	21
105	3.6.2 Example - successful addItem:.....	22
106	3.7 Retrieving resources.....	23
107	3.7.1 Example - successful fetchItem:.....	23
108	3.7.2 Example - unsuccessful fetchItem:.....	24
109	3.8 Updating resources.....	24
110	3.8.1 Change tokens and concurrent updates.....	28
111	3.8.2 Example - successful update:.....	29
112	3.8.3 Other updates:.....	30
113	3.8.4 Creating an update message.....	31
114	3.9 Deletion of resources.....	32
115	3.9.1 Example - successful deleteItem:.....	32
116	3.9.2 Example - unsuccessful deleteItem:.....	33
117	3.10 Querying calendar resources.....	33
118	3.10.1 Calendar Query common types.....	33
119	3.10.2 CompFilterType.....	34
120	3.10.3 PropFilterType.....	34
121	3.10.4 ParamFilterType.....	35
122	3.10.5 CalendarQueryType elements.....	36
123	3.10.6 Specifying data to be returned.....	36
124	3.10.7 Pre/postconditions for calendar queries.....	37
125	3.10.8 Time range limited queries.....	37
126	3.10.9 Example: time range limited retrieval.....	37
127	3.11 Free-busy queries.....	39
128	3.11.1 Element values	40
129	3.11.1.1 start.....	40
130	3.11.1.2 end.....	40
131	3.11.2 Examples.....	40
132	3.12 Multiple operations.....	42
133	4 Conformance.....	44
134	4.1 Start, end and duration in calendar components.....	44
135	4.1.1 Updating, transporting and maintaining start, and and duration.....	44
136	4.1.2 VEVENT:.....	44
137	4.1.3 VTODO:.....	44

138	4.1.4 VJOURNAL:.....	44
139	4.1.5 VAVAILABILITY.....	45
140	4.1.6 AVAILABILITY.....	45
141	4.2 Recurrences.....	45
142	4.3 Alarms:.....	45
143	4.4 Unrecognized or unsupported elements.....	45
144	Appendix A. Acknowledgments.....	46
145	Participants:.....	46
146	Appendix B. Revision History.....	47
147		

148 1 Introduction

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

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

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

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

160 1.1 Terminology

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

164 1.2 Normative References

- 165 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*.
166 IETF RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 167 **[RFC 2616]** Fielding, et al, *Hypertext Transfer Protocol -- HTTP/1.1*
168 <http://tools.ietf.org/html/rfc2616>
- 169 **[RFC 4791]** Daboo, et al. *Calendar Extensions to WebDAV (CalDAV)*.
170 <http://www.ietf.org/rfc/rfc4791.txt>.
- 171 **[draft caldav-sched]** Desruisseaux, et al. *CalDAV Scheduling extensions to WebDAV*
172 <http://tools.ietf.org/html/draft-desruisseaux-caldav-sched-08>
- 173 **[RFC 5545]** B. Desruisseaux, *Internet Calendaring and Scheduling Core Object*
174 *Specification (iCalendar)*
175 <http://tools.ietf.org/html/rfc5546>
- 176 **[RFC 5546]** C. Daboo, *iCalendar Transport-Independent Interoperability Protocol*
177 *(iTIP)*
178 <http://tools.ietf.org/html/rfc5545>
- 179 **[RFC 6321]** C. Daboo, M. Douglass, S. Lees *xCal: The XML format for iCalendar*
180 <http://www.ietf.org/rfc/rfc6321.txt>
- 181 **[draft-timezones]** C. Daboo, M. Douglass: *Timezone Service Protocol*
182 <http://tools.ietf.org/html/draft-douglass-timezone-service>
- 183 **[FreeBusy Read URL]** E York. *Freebusy read URL*
184 <http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL>
185 [%20V1.0.pdf](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)
- 186 **[SOAP11]** Simple Object Access Protocol (SOAP) 1.1, 8 May 2000
187 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 188
- 189 **[WSDL11]** Web Services Description Language (WSDL) 1.1, 15 March 2001
190 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

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

194 1.3 Non-normative References

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

205 1.4 Namespace

206 XML namespaces and prefixes used in this standard:

207 Table 1-1: XML Namespaces in this standard

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

208

209 **2 Issues not addressed by this specification.**

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

212 **2.1 Access Control**

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

215 **2.2 Provisioning**

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

219 **2.3 Copy/Move**

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

224 **2.4 Creating Collections**

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

227 **2.5 Retrieving collections**

228 This operation is currently undefined.

229 **2.6 Setting service and resource properties.**

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

232 **3 Basic Calendar Access**

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

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

238 **3.1 CalWS Glossary**

239 **3.1.1 Calendar Object Resource**

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

242 **3.1.2 Uid**

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

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

248 **3.1.3 Collections**

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

252 **3.1.4 Calendar Collection**

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

256 **3.1.5 Scheduling Calendar Collection**

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

259 **3.1.6 Principal Home**

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

262 **3.1.7 Change token**

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

21

268 **3.2 Overview of the CalWS protocol**

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

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

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

276 A system entity acting as a CalWS requester transmits a CalWS request element within the body of a
277 SOAP message to a system entity acting as a CalWS responder. The CalWS requester MUST NOT
278 include more than one CalWS request per SOAP message or include any additional XML elements in the
279 SOAP body (though see Section 3.12 for multiple messages packaged in one request).

280 The CalWS responder MUST return either a CalWS response element within the body of another SOAP
281 message or generate a SOAP fault. The CalWS responder MUST NOT include more than one CalWS
282 response per SOAP message or include any additional XML elements in the SOAP body. If a CalWS
283 responder cannot, for some reason, process a CalWS request, it MUST generate a SOAP fault. (SOAP
284 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)

285 **3.2.1 Discovery**

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

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

291 **3.2.2 Properties**

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

295 **3.2.3 Operations**

296 The following operations are defined by this specification:

- 297 • Retrieval and update of service and resource properties
- 298 • Creation of a calendar object
- 299 • Retrieval of a single calendar object
- 300 • Multiget of one or more calendar objects
- 301 • Update of a calendar object
- 302 • Deletion of a calendar object
- 303 • Query
- 304 • Free-busy query
- 305 • Multiple operations

306 **3.2.4 Calendar Object Resources**

307 The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC 4791] section
308 4.2. An additional constraint for CalWS is that no timezone specifications are transferred with the data.

309 3.2.5 Timezone information

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

315 3.2.6 Error conditions

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

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

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

327 3.2.6.1 Example: error with error condition

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

336 3.3 CalWs-SOAP Messages.

337 This section describes the common elements and structure of CalWs-SOAP messages. The conventions
 338 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		

339 *Table 1: Field column descriptions*340 **3.3.1 Common Elements and types**

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

343 All requests must include an href which specifies the target for the request. There is also an id attribute
344 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.

345 *Table 2: BaseRequestType elements*

346 A response may include an error response element of type ErrorResponse. This element will be
347 returned in response messages when some form of processing error occurs and provides further
348 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

349 *Table 3: ErrorResponse elements*

30
350

3.3.1.1 ErrorCodeType

351 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.

Field	Type	Description
partialSuccess	PartialSuccessType	Indicates that a MultiOpType operation was partially successful. Returned when the operation is marked as non-atomic and one or more sub-operations failed. The entire response needs to be examined to determine failing operations.
missingChangeToken	MissingChangeTokenType	An operation was attempted which required a change token but none was supplied. Note that it appears that the marshalling or demarshalling should handle this as the token is required. It doesn't.
mismatchedChangeToken	MismatchedChangeTokenType	An update operation was attempted with a change token value which does not match that held by the service. The client must refetch the entity to refresh its cached value and token. Note that matching of tokens is a server responsibility. The token is opaque to the client but probably structured to the server. Certain non-conflicting updates may be allowed even if the token has changed.
invalidFilter	InvalidFilterType	
uidConflict	UidConflictType	An attempt was made to store an entity which would result in more than one entity having equal uids. The entity uid must be unique within a collection. Recurring event or task overrides have the same uid and are considered part of a single entity.

352 *Table 4: ErrorCodeType definitions*353 **3.3.1.2 BaseResponseType**

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

354 *Table 5: BaseResponseType elements*355 **3.4 Properties**

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

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

39

361 3.4.1 childCollection

362 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

363 Table 6: ChildCollectionType fields

364 See resourceType for descriptions of CollectionType and Calendar CollectionType.

365 3.4.2 creationDateTime

366 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 Error: Reference source not found Section 5.6.

367 Table 7: CreationDateTimeType fields

368 3.4.3 displayName

369 This property SHOULD be returned for any targeted resource.

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

370 Table 8: DisplayNameType fields

371 3.4.4 lastModifiedDateTime

372 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 [WS-Calendar].

373 Table 9: LastModifiedDateTimeType fields

374 3.4.5 maxAttendeesPerInstance

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

42

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

377 *Table 10: MaxAttendeesPerInstanceType fields*378 **3.4.6 maxDateTime**

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

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

381 *Table 11: MaxDateTimeType fields*382 **3.4.7 maxInstances**

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

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

385 *Table 12: MaxInstancesType fields*386 **3.4.8 maxResourceSize**

387 This property SHOULD be returned for the service and MAY be returned for any targeted collection
388 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.

389 *Table 13: MaxResourceSizeType fields*390 **3.4.9 minDateTime**

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

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

393 *Table 14: MinDateTimeType fields*

45
394 **3.4.10 principalHome**

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

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

397 *Table 15: PrincipalHomeType fields*

398 **3.4.11 resourceDescription**

399 Provides some descriptive text for the targeted collection.

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

400 *Table 16: ResourceDescriptionType fields*

401 **3.4.12 resourceOwner**

402 This property SHOULD be returned for any targeted resource.

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

403 *Table 17: ResourceownerType fields*

404 **3.4.13 resourceTimezoneId**

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

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

407 *Table 18: ResourceTimezoneIdType fields*

408 **3.4.14 resourceType**

409 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.

410 *Table 19: ResourceType fields*

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

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

412 *Table 20: XresourceType fields*413 **3.4.15 supportedCalendarComponentSet**

414 This property identifies which component types the service is prepared to store. The allowable
415 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.

416 *Table 21: SupportedCalendarComponentSetType fields*417 **3.4.16 supportedFeatures**

418 This property SHOULD be returned for the service and MAY be returned for any targeted collection
419 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.

420 *Table 22: SupportedFeaturesType fields*

51
421 **3.4.17 timezoneServer**

422 This property SHOULD be returned for the service and MAY be returned for any targeted collection
423 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.

424 *Table 23: TimezoneServerType fields*

425 **3.4.18 CalWS:privilege-set XML element**

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

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

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

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

- 432 • CalWS: Read - current principal has read access
- 433 • CalWS: Write - current principal has write access

434 `<calws:privilege-set>`
435 `<calws:privilege><calws:read></calws:privilege>`
436 `<calws:privilege><calws:write></calws:privilege>`
437 `</calws:privilege-set>`

438 **3.5 Retrieving Collection and Service Properties**

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

441 The service properties define the global limits and defaults. Any properties defined on collections within
442 the service hierarchy override those service defaults. The service may choose to prevent such overriding
443 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.

444 *Table 24: GetPropertiesType fields*

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

445 *Table 25: GetPropertiesResponseType fields*446 **3.5.1 Example - retrieving server properties:**

```

447 >>Request
448
449 <?xml version="1.0" encoding="UTF-8"?>
450 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
451   <SOAP-ENV:Header/>
452   <SOAP-ENV:Body>
453     <ns2:getProperties xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
454       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
455       <ns2:href/></ns2:href>
456     </ns2:getProperties>
457   </SOAP-ENV:Body>
458 </SOAP-ENV:Envelope>
459
460 >>Response
461
462 <?xml version="1.0" encoding="UTF-8"?>
463 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
464   <SOAP-ENV:Header />
465   <SOAP-ENV:Body>
466     <ns2:getPropertiesResponse
467       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
468       xmlns:ns4="urn:ietf:params:xml:ns:icalendar-2.0"
469       id="0" >
470     <ns2:href/></ns2:href>
471     <ns2:lastModifiedDate>
472       <ns2:dateTime>2012-01-04T18:21:14Z</ns2:dateTime>
473     </ns2:lastModifiedDate>
474     <ns2:supportedCalendarComponentSet>
475       <ns4:vevent />
476       <ns4:vtodo />
477       <ns4:vavailability />
478     </ns2:supportedCalendarComponentSet>
479     <ns2:resourceType>
480       <ns2:collection />
481     </ns2:resourceType>
482     <ns2:supportedFeatures>
483       <ns2:calendarAccessFeature />
484     </ns2:supportedFeatures>
485     <ns2:maxInstances>
486       <ns2:integer>1000</ns2:integer>
487     </ns2:maxInstances>
488     <ns2:maxResourceSize>
489       <ns2:integer>100000</ns2:integer>
490     </ns2:maxResourceSize>
491     </ns2:getPropertiesResponse>
492   </SOAP-ENV:Body>
493 </SOAP-ENV:Envelope>
494
495

```

57

496 3.6 Creating Calendar Object Resources

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

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

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

502 Table 26: AddItemType fields

503 The service will respond with an AddItemResponseType giving either the href and change token of the
504 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

505 Table 27: AddItemResponseType additional fields

506 3.6.1 Preconditions for Calendar Object Creation

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

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

541 3.6.2 Example - successful addItem:

```

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

```

63

591 3.7 Retrieving resources

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

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

595 Table 28: FetchItemType fields

596 The service will respond with a FetchItemResponseType containing either the change token, its href and
597 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

598 Table 29: FetchItemResponseType additional fields

599 3.7.1 Example - successful fetchItem:

```

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

```

```

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

```

666 3.7.2 Example - unsuccessful fetchItem:

```

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

```

695 3.8 Updating resources

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

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

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

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

```

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

```

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

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

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

- 733 • Remove - components, properties or parameters
734 • Add - components, properties or parameters
735 • Change - property or parameter values

736 Removals MUST be processed ahead of additions

737 Preconditions as specified in Preconditions for Calendar Object Creation are applicable. The response
738 will indicate success or failure of the update. If the change token value does not match that held by the
739 service a mismatchedChangeToken error status will be returned. The client should re-fetch the entity to
740 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

741 *Table 30: UpdateItem Type fields*

742 The ComponentSelectionType contains three repeating child elements. The first allows for selection of
743 nested components which can then be updated. The next allows addition of entire components and the
744 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

745 *Table 31: ComponentsSelectionType fields*

746 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

747 *Table 32: PropertiesSelectionType fields*748 To complete that pattern there is also a ParametersSelectionType used to select property parameters for
749 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

750 *Table 33: ParametersSelectionType fields*751 Each of these refers to a reference type. These either provide a complete entity for addition or identify the
752 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.

753 *Table 34: ComponentReferenceType fields*

75

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

754 *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.

755 *Table 36: ParameterReferenceType fields*

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

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

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

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

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

768 *Table 37: ComponentSelectionType fields*

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

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

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

78

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

775 *Table 38: PropertySelectionType fields*

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

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

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

780 *Table 39: ParameterSelectionType fields*

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

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

783 *Table 40: UpdateItemResponseType additional fields*

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

785 **3.8.1 Change tokens and concurrent updates**

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

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

81
797

3.8.2 Example - successful update:

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

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

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

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

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

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

```

84
862         <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
863         </ns3:dtstart>
864     </ns2:change>
865 </ns2:property>
866 <ns2:property>
867     <ns3:summary>
868     <ns3:text>try this</ns3:text>
869 </ns3:summary>
870 <ns2:change>
871     <ns3:summary>
872     <ns3:text>A changed summary - again and again and again</ns3:text>
873 </ns3:summary>
874 </ns2:change>
875 </ns2:property>
876 <ns2:add>
877     <ns3:categories>
878     <ns3:text>newcategory-2</ns3:text>
879     <ns3:text>resources</ns3:text>
880     <ns3:text>paper</ns3:text>
881 </ns3:categories>
882 </ns2:add>
883 </ns2:properties>
884 </ns2:component>
885 </ns2:components>
886 </ns2:select>
887 </ns2:updateItem>
888 </SOAP-ENV:Body>
889 </SOAP-ENV:Envelope>
890 >>Response
891
892 <?xml version="1.0" encoding="UTF-8"?>
893 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
894 <SOAP-ENV:Header/>
895 <SOAP-ENV:Body>
896     <ns2:updateItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
897         xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
898         id="0">
899         <ns2:status>OK</ns2:status>
900     </ns2:updateItemResponse>
901 </SOAP-ENV:Body>
902 </SOAP-ENV:Envelope>
903

```

904 3.8.3 Other updates:

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

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

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

```

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

```

```

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

```

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

```

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

```

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

```

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

```

960 In this example we change the dtstart timezone identifier.

```

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

```

985

986 3.8.4 Creating an update message.

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

90
 989 comparing XML for differences is difficult. However, we can take advantage of the structure of
 990 calendaring entities to simplify the process. There are implementations available which take the diff
 991 approach to producing an update stream.
 992 There are some special cases to consider when comparing. Some properties are multi-valued and may
 993 themselves appear more than once. There is no semantic information implied by any grouping though
 994 parameters may need to be taken into account. These properties need to be normalized before
 995 comparison and when updating them we produce a change which treats each value as a single property.

996 These properties are

- 997 • categories
- 998 • exdate
- 999 • freebusy
- 1000 • rdate

1001 This normalization can take place before comparison.

1002 Some properties are multi-valued and may only appear once. At the moment the only standard property is
 1003 resource which may take a comma separated list. This should be treated as a single multi-valued property
 1004 when comparing. The order is unimportant. Sorting the values may help.

1005 Some properties may appear multiple times, for example comment. Comparison should take account of
 1006 parameters. Ordering all properties appropriately allows for relatively simple comparison.

1007 3.9 Deletion of resources

1008 Deletion of calendar object resources is carried out by using a CalWS-SOAP deleteItem request with an
 1009 href specifying the entity to be deleted. The deleteItem request is not valid when the href specifies a
 1010 collection.

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

1011 *Table 41: DeleteItem fields*

1012 The service will respond with a DeleteItemResponseType containing the status and a possible error
 1013 response. There are no additional elements.

1014 3.9.1 Example - successful deleteItem:

```

1015 >>Request
1016 <?xml version="1.0" encoding="UTF-8"?>
1017 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1018   <SOAP-ENV:Header/>
1019   <SOAP-ENV:Body>
1020     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1021       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1022       <ns2:href>/user/douglm/calendar/1302620814655.ics</ns2:href>
1023     </ns2:deleteItem>
1024   </SOAP-ENV:Body>
1025 </SOAP-ENV:Envelope>
1026
1027 >>Response
1028
1029 <?xml version="1.0" encoding="UTF-8"?>
1030 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1031   <SOAP-ENV:Header/>
1032   <SOAP-ENV:Body>
1033     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1034       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1035       <ns2:status>OK</ns2:status>
1036     </ns2:deleteItemResponse>
1037
  
```


93
1038
1039

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

1040 3.9.2 Example - unsuccessful deleteItem:

```

1041 >>Request
1042
1043 <?xml version="1.0" encoding="UTF-8"?>
1044 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1045   <SOAP-ENV:Header/>
1046   <SOAP-ENV:Body>
1047     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1048       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1049       <ns2:href>/user/doug1m/calendar/nosuchevent.ics</ns2:href>
1050     </ns2:deleteItem>
1051   </SOAP-ENV:Body>
1052 </SOAP-ENV:Envelope>
1053
1054 >>Response
1055
1056 <?xml version="1.0" encoding="UTF-8"?>
1057 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1058   <SOAP-ENV:Header/>
1059   <SOAP-ENV:Body>
1060     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1061       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1062       <ns2:status>Error</ns2:status>
1063       <ns2:errorResponse>
1064         <ns2:targetDoesNotExist/>
1065       </ns2:errorResponse>
1066     </ns2:deleteItemResponse>
1067   </SOAP-ENV:Body>
1068 </SOAP-ENV:Envelope>

```

1069 3.10 Querying calendar resources

1070 Querying provides a mechanism by which information can be obtained from the service through possibly
1071 complex queries. A skeleton icalendar entity can be provided to limit the amount of information returned to
1072 the client. A query takes the parts

- 1073 • Limitations on the data returned
- 1074 • Selection of the data
- 1075 • Optional timezone id for floating time calculations.

1076 3.10.1 Calendar Query common types

1077 The UTCTimeRangeType is used in a number of places to define a time range within which components
1078 must appear or property values must lie. The values are UTC time-date, the start is inclusive and the end
1079 is exclusive.

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

1080 *Table 42: UTCTimeRangeType elements*

1081 The TextMatchType is used to match text values in properties and parameters. The collation attribute
1082 species a collation as defined in [RFC4790].

1083 Servers are REQUIRED to support the "i;ascii-casemap" and "i;octet" collations which provide a basic
1084 case insensitive and case sensitive match respectively.

96
1085

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

Field	Type	#	?	Description
#collation	String	0..1	N	Collation name from [RFC4790]. "
#negate-condition	boolean	0..1	N	if "true" negates the condition

1086 Table 43: TextMatchType attributes

1087 3.10.2 CompFilterType

1088 This type defines a search query for the calendar query operation. It specifies the component types to
1089 return, absence tests or basic matching operations on properties and time ranges.

1090 The top level comp-filter element (which must match a vcalendar component may contain zero or more
1091 comp-filter elements to match events, tasks or other contained components. These in turn may contain
1092 further nested comp-filter elements to match further levels of nested components.

1093 Each may also contain prop-filter elements to test for the absence of properties or to match values.

Field	Type	#	?	Description
anyComp	AnyCompType	0..1	C	One of anyComp, vcalendar or a BaseComponentType must be supplied. anyComp indicates that any component will match.
xcal:vcalendar	xcal:VcalendarType	0..1	C	Matches vcalendar at the top level. Must be provided
xcal:baseComponent	xcal:BaseComponentType	0..1	C	May be vevent or vtodo for example.
#test	String	0..1	N	"anyof" is a logical OR of the child elements. "allof" is a logical AND of the child elements.
is-not-defined	empty	0..1	N	Only this element or one or more of time-range, prop-filter or comp-filter may be present
time-range	UTCTimeRangeType	0..1	N	
comp-filter	CompFilterType	1	Y	Match against contained components
prop-filter	PropFilterType	0..n	N	Match against component properties

1094 Table 44: CompFilterType elements

1095 3.10.3 PropFilterType

1096 The prop-filter element may test for the absence of a property or match values or specify zero or more
1097 ParamFilterType elements to match against parameters.

Field	Type	#	?	Description
xcal:baseProperty	xcal:BasePropertyType	1	Y	Specifies the property to be matched.
#test	String	0..1	N	"anyof" is a logical OR of the child elements. "allof" is a logical AND of the child elements.
is-not-defined	empty	0..1	N	Only this element or optionally one of time-range or text-match followed by param-filter
time-range	UTCTimeRangeType	0..1	N	
text-match	TextMatchtype	0..1	N	
param-filter	ParamFilterType	0..n	N	Match against property parameters

1098 *Table 45: PropFilterType elements*1099 **3.10.4 ParamFilterType**

1100 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	

1101 *Table 46: ParamFilterType elements*

3.10.5 CalendarQueryType elements

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

1103 Table 47: CalendarQueryType elements

1104 3.10.6 Specifying data to be returned

1105 This is achieved by specifying one of the following

- 1106 • allprop: return all properties and calendar data. (some properties are specified as not being part of the
1107 allprop set so are not returned)

- 105
1108 • Set the icalendar element. This is an icalendar valueless pattern entity which provides a map of the
1109 components and properties to be returned. Neither the pattern nor the returned result need to be
1110 valid icalendar entities in that required properties may be absent if unselected.

1111 3.10.7 Pre/postconditions for calendar queries

1112 The preconditions as defined in [RFC 4791] Section 7.8 apply here. CalWS errors may be reported by the
1113 service when preconditions or postconditions are violated.

1114 3.10.8 Time range limited queries.

1115 Time-range limited retrieval has some special characteristics. The simplest case is a single event or task
1116 which overlaps the requested time-period. Recurring items and other components such as alarms
1117 complicate the picture.

1118 3.10.9 Example: time range limited retrieval

1119 This example shows the time-range limited retrieval from a calendar which results in 2 events, one a
1120 recurring event and one a simple non-recurring event.

```
1121 >> Request <<
1122 <?xml version="1.0" encoding="UTF-8"?>
1123 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1124 <SOAP-ENV:Header/>
1125 <SOAP-ENV:Body>
1126 <ns2:calendarQuery xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1127 <ns2:href>/user/douglm/calendar</ns2:href>
1128 <ns3:icalendar xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1129 <ns3:vcalendar>
1130 <ns3:components>
1131 <ns3:vevent>
1132 <ns3:properties>
1133 <ns3:summary/>
1134 <ns3:dtstart/>
1135 <ns3:dtend/>
1136 <ns3:duration/>
1137 <ns3:uid/>
1138 <ns3:recurrence-id/>
1139 <ns3:rrule/>
1140 <ns3:rdate/>
1141 <ns3:exdate/>
1142 </ns3:properties>
1143 </ns3:vevent>
1144 </ns3:components>
1145 </ns3:vcalendar>
1146 </ns3:icalendar>
1147 <ns2:filter>
1148 <ns2:compFilter test="anyof">
1149 <ns3:vcalendar />
1150 <ns2:compFilter>
1151 <ns3:vevent />
1152 <ns2:time-range end="20110430T040000Z" start="20110401T040000Z"/>
1153 </ns2:compFilter>
1154 </ns2:filter>
1155 </ns2:calendarQuery>
1156 </SOAP-ENV:Body>
1157 </SOAP-ENV:Envelope>
1158
1159 >> Response <<
1160 <?xml version="1.0" encoding="UTF-8"?>
1161 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1162 <SOAP-ENV:Header/>
1163 <SOAP-ENV:Body>
1164 <ns2:calendarQueryResponse
```

108
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238

```
xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
<ns2:status>OK</ns2:status>
<ns2:response>
  <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
  <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
  <ns2:propstat>
    <ns2:prop>
      <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
        <ns3:icalendar>
          <ns3:vcalendar>
            <ns3:properties>
              <ns3:prodid>
                <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
              </ns3:prodid>
              <ns3:version>
                <ns3:text>2.0</ns3:text>
              </ns3:version>
            </ns3:properties>
            <ns3:components>
              <ns3:vevent>
                <ns3:properties>
                  <ns3:dtend>
                    <ns3:date-time>20110406T160000Z</ns3:date-time>
                  </ns3:dtend>
                  <ns3:dtstart>
                    <ns3:date-time>20110406T150000Z</ns3:date-time>
                  </ns3:dtstart>
                  <ns3:summary>
                    <ns3:text>try this</ns3:text>
                  </ns3:summary>
                  <ns3:uid>
                    <ns3:text>1302105461170</ns3:text>
                  </ns3:uid>
                </ns3:properties>
              </ns3:vevent>
            </ns3:components>
          </ns3:vcalendar>
        </ns3:icalendar>
      </ns2:calendar-data>
    </ns2:prop>
  </ns2:propstat>
  <ns2:status>OK</ns2:status>
</ns2:response>
<ns2:response>
  <ns2:href>/user/douglm/calendar/CAL-00f1fc61-2f021bca-012f-022947f8-
00000006.ics</ns2:href>
  <ns2:changeToken>"20110405T140920Z-0"</ns2:changeToken>
  <ns2:propstat>
    <ns2:prop>
      <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
        <ns3:icalendar>
          <ns3:vcalendar>
            <ns3:properties>
              <ns3:prodid>
                <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
              </ns3:prodid>
              <ns3:version>
                <ns3:text>2.0</ns3:text>
              </ns3:version>
            </ns3:properties>
            <ns3:components>
              <ns3:vevent>
                <ns3:properties>
                  <ns3:duration>
                    <ns3:duration>PT1H</ns3:duration>
                  </ns3:duration>
                  <ns3:dtstart>
                    <ns3:parameters>
                      <ns3:tzid>
                        <ns3:text>America/New_York</ns3:text>

```

111
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300

```
        </ns3:tzid>
        </ns3:parameters>
        <ns3:date-time>20110412T110000</ns3:date-time>
    </ns3:dtstart>
    <ns3:summary>
        <ns3:text>Test recurring event</ns3:text>
    </ns3:summary>
    <ns3:uid>
        <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
00000006demobedework@mysite.edu</ns3:text>
    </ns3:uid>
    <ns3:rrule>
        <ns3:recur>
            <ns3:freq>WEEKLY</ns3:freq>
            <ns3:count>2</ns3:count>
            <ns3:interval>1</ns3:interval>
        </ns3:recur>
    </ns3:rrule>
    </ns3:properties>
</ns3:vevent>
<ns3:vevent>
    <ns3:properties>
        <ns3:recurrence-id>
            <ns3:parameters>
                <ns3:tzid>
                    <ns3:text>America/New_York</ns3:text>
                </ns3:tzid>
            </ns3:parameters>
            <ns3:date-time>20110419T150000Z</ns3:date-time>
        </ns3:recurrence-id>
        <ns3:duration>
            <ns3:duration>PT1H</ns3:duration>
        </ns3:duration>
        <ns3:dtstart>
            <ns3:parameters>
                <ns3:tzid>
                    <ns3:text>America/New_York</ns3:text>
                </ns3:tzid>
            </ns3:parameters>
            <ns3:date-time>20110419T120000</ns3:date-time>
        </ns3:dtstart>
        <ns3:summary>
            <ns3:text>Test recurring event</ns3:text>
        </ns3:summary>
        <ns3:uid>
            <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
00000006demobedework@mysite.edu</ns3:text>
        </ns3:uid>
    </ns3:properties>
</ns3:vevent>
</ns3:components>
</ns3:vcalendar>
</ns3:icalendar>
</ns2:calendar-data>
</ns2:prop>
<ns2:status>OK</ns2:status>
</ns2:propstat>
</ns2:response>
</ns2:calendarQueryResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

1301 3.11 Free-busy queries

1302 Freebusy queries are used to obtain freebusy information for a principal. The result contains information
1303 only for events to which the current principal has sufficient access and may be affected by components
1304 and rules available only to the server (for instance office hours availability).

114
1305 These queries are carried out by using a CalWS-SOAP freebusyReport request with an href specifying a
1306 principal. The freebusyReport request is not valid when the href specifies any entity other than a principal.
1307 The query follows the specification defined in [FreeBusy Read URL] with certain limitations. As an
1308 authenticated user to the CalWS service scheduling read-freebusy privileges must have been granted. As
1309 an unauthenticated user equivalent access must have been granted to unauthenticated users.
1310 Freebusy information is returned by default as xcalendar vfreebusy components, as defined by [RFC
1311 6321]. Such a component is not meant to conform to the requirements of VFREEBUSY components in
1312 RFC 5546. The VFREEBUSY component SHOULD conform to section "4.6.4 Free/Busy Component" of
1313 [RFC 5545]. A client SHOULD ignore the ORGANIZER field.
1314 Since a Freebusy query can only refer to a single user, a client will already know how to match the result
1315 component to a user. A server MUST only return a single vfreebusy component.

1316 3.11.1 Element values

1317 Three values are provided: href; start; end. Only the href is required. The start and end are in XML UTC
1318 date/time format and are interpreted as follows:

1319 3.11.1.1 start

1320 **Default:** If omitted the default value is left up to the server. It may be the current day, start of the
1321 current month, etc.
1322 **Description:** Specifies the start date for the Freebusy data. The server is free to ignore this value
1323 and return data in any time range. The client must check the data for the returned time range.
1324 **Format:** An XML UTC date-time
1325 **Example:**
1326 2011-12-01T10:15:00Z
1327 **Notes:** Specifying only a start date/time without specifying an end-date/time or period should be
1328 interpreted as in [RFC 5545]. The effective period should cover the remainder of that day.

1329 3.11.1.2 end

1330 **Default:** Same as start
1331 **Description:** Specifies the end date for the Freebusy data. The server is free to ignore this value.
1332 **Format:** Same as start
1333 **Example:** Same as start
1334 The server is free to ignore the start, end and period parameters. It is recommended that the server return
1335 at least 6 weeks of data from the current day.
1336 A client MUST check the time range in the response as a server may return a different time range than
1337 the requested range.

1338 3.11.2 Examples

1339 The following is an unsuccessful request targeting an invalid resource.

```
1340 >> Request <<
1341
1342 <?xml version="1.0" encoding="UTF-8"?>
1343 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1344   <SOAP-ENV:Header/>
1345   <SOAP-ENV:Body>
1346     <ns2:freebusyReport
1347       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1348       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1349       <ns2:href>/user/douglm/calendar</ns2:href>
1350       <ns2:time-range>
1351         <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1352         <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1353       </ns2:time-range>
1354     </ns2:freebusyReport>
```



```

117      </SOAP-ENV:Body>
1355    </SOAP-ENV:Envelope>
1356
1357  >> Response <<
1358
1359  <?xml version="1.0" encoding="UTF-8"?>
1360  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1361    <SOAP-ENV:Header/>
1362    <SOAP-ENV:Body>
1363      <ns2:freebusyReportResponse
1364        xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1365        xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1366        <ns2:status>Error</ns2:status>
1367        <ns2:message>Only principal href supported</ns2:message>
1368      </ns2:freebusyReportResponse>
1369    </SOAP-ENV:Body>
1370  </SOAP-ENV:Envelope>
1371

```

1372 The following is an example of a request to retrieve Freebusy data for a user:

```

1373  >> Request <<
1374
1375  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1376    <SOAP-ENV:Header/>
1377    <SOAP-ENV:Body>
1378      <ns2:freebusyReport
1379        xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1380        xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1381        <ns2:href>/principals/users/douglm</ns2:href>
1382        <ns2:time-range>
1383          <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1384          <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1385        </ns2:time-range>
1386      </ns2:freebusyReport>
1387    </SOAP-ENV:Body>
1388  </SOAP-ENV:Envelope>
1389
1390  >> Response <<
1391
1392  <?xml version="1.0" encoding="UTF-8"?>
1393  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1394    <SOAP-ENV:Header/>
1395    <SOAP-ENV:Body>
1396      <ns2:freebusyReportResponse
1397        xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1398        xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1399        <ns2:status>OK</ns2:status>
1400        <ns3:icalendar>
1401          <ns3:vcalendar>
1402            <ns3:properties>
1403              <ns3:prodid>
1404                <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1405              </ns3:prodid>
1406              <ns3:version>
1407                <ns3:text>2.0</ns3:text>
1408              </ns3:version>
1409            </ns3:properties>
1410            <ns3:components>
1411              <ns3:vfreebusy>
1412                <ns3:properties>
1413                  <ns3:attendee>
1414                    <ns3:parameters>
1415                      <ns3:partstat>
1416                        <ns3:text>NEEDS-ACTION</ns3:text>
1417                      </ns3:partstat>
1418                    </ns3:parameters>
1419                    <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1420                  </ns3:attendee>
1421                <ns3:created>
1422                  <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1423                </ns3:created>
1424                <ns3:dtend>

```

```

120      <ns3:date-time>2011-04-30T00:00:00Z</ns3:date-time>
1425    </ns3:dtend>
1426    <ns3:dtstamp>
1427      <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1428    </ns3:dtstamp>
1429    <ns3:dtstart>
1430      <ns3:date-time>2011-04-01T00:00:00Z</ns3:date-time>
1431    </ns3:dtstart>
1432    <ns3:freebusy>
1433      <ns3:parameters>
1434        <ns3:fctype>
1435          <ns3:text>BUSY</ns3:text>
1436        </ns3:fctype>
1437      </ns3:parameters>
1438      <ns3:period>
1439        <ns3:start>2011-04-06T15:00:00Z</ns3:start>
1440        <ns3:end>2011-04-06T16:00:00Z</ns3:end>
1441      </ns3:period>
1442    </ns3:freebusy>
1443    <ns3:last-modified>
1444      <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1445    </ns3:last-modified>
1446    <ns3:organizer>
1447      <ns3:parameters/>
1448      <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1449    </ns3:organizer>
1450    <ns3:uid>
1451      <ns3:text>2UTDVPZ9H0EQL9QISI44SP5IFPC4N75</ns3:text>
1452    </ns3:uid>
1453  </ns3:properties>
1454 </ns3:vfreebusy>
1455 </ns3:components>
1456 </ns3:vcalendar>
1457 </ns3:icalendar>
1458 </ns2:freebusyReportResponse>
1459 </SOAP-ENV:Body>
1460 </SOAP-ENV:Envelope>
1461

```

1462

1463 3.12 Multiple operations

1464 Each of the previously described operations acts upon a single entity or resource only. Frequently we
1465 have the need to update an interconnected set of entities so that we maintain the consistency of the
1466 structure. This requires an atomic operation which can successfully update all the entities or roll back the
1467 operation on failure.

1468 The MultiOpType operation provides such a feature. It is essentially a wrapper around any of the other
1469 operations which guarantees the success of the entire set or a roll back. Using the id attribute for
1470 requests, each individual response can be located in the result.

1471 The MultiOpType request takes the following elements

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

1472 Table 48: MultiOpType elements

1473 The response type is also simple containing a single element containing all the responses.

123

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

1474 Table 49: MultiOpResponseType elements

1475

1476

1477

1478

1479 4 Conformance

1480 Certain calendaring properties and components are interrelated and it is necessary to have knowledge of
 1481 all these properties and their current values to allow consistent update and understanding of a target
 1482 component. The normative definition for these relationships is RFC5445, RFC5446 and related RFCs.

1483 However, those specifications assume a complete view of entities being fetched or updated. This
 1484 specification allows updates of entities when only a partial view is available. In fact it is the very nature of
 1485 SOAP based transaction to provide such a partial view. Given that, parties attempting to update entities
 1486 MUST have sufficient information to ensure the end result is consistent. Services allowing updates to
 1487 entities MUST ensure that the result after an update operation is still internally consistent.

1488 4.1 Start, end and duration in calendar components

1489 A period of time is fully specified by a start and an end or duration.

1490 4.1.1 Updating, transporting and maintaining start, and and duration.

- 1491 • For all components the calculated or specified start must be at or before the end.
- 1492 • When a system updates or stores a calendar component it MUST retain the relationship of start, end
 1493 and duration. Applications MUST NOT without good cause, change a start and end pair into a start
 1494 and duration nor the reverse. Semantically they are not equivalent when DST transitions occur
 1495 during the time of the event.
- 1496 • For interoperability, iCalendar based systems SHOULD avoid the use of weekly durations and XML
 1497 based systems SHOULD avoid the use of yearly durations.

1498 4.1.2 VEVENT:

- 1499 • The three properties are DTSTART, DTEND and DURATION.
- 1500 • DTSTART MUST appear once and only one of DTEND or DURATION MAY be present.
- 1501 • The DTSTART property for a VEVENT specifies the inclusive start of the event. For recurring events, it
 1502 also specifies the very first instance in the recurrence set.
- 1503 • The DTEND property for a VEVENT calendar component specifies the non-inclusive end of the event.
- 1504 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE value
 1505 type but no DTEND nor DURATION property, the event's duration is taken to be one day.
- 1506 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE-TIME
 1507 value type but no DTEND nor DURATION property, the event ends on the same calendar date and
 1508 time of day specified by the DTSTART property, that is, it signifies a zero length instant in time.

1509 4.1.3 VTODO:

- 1510 • The three properties are DTSTART, DUE, DURATION.
- 1511 • DTSTART MAY appear once.
- 1512 • Either DUE or DURATION MAY appear in a VTODO, but DUE and DURATION MUST NOT occur in
 1513 the same VTODO.
- 1514 • If DURATION does appear in a VTODO, then DTSTART MUST also appear in the same VTODO.
- 1515 • The three properties for a VTODO are related in the same way as for VEVENT. Additionally a VTODO
 1516 calendar component without the DTSTART and DUE (or DURATION) properties specifies a
 1517 VTODO that will be associated with each successive calendar date, until it is completed.

1518 4.1.4 VJOURNAL:

- 1519 • DTSTART only, which may be a date or date-time value.

129
1520

4.1.5 VAVAILABILITY

- 1521 • DTSTART and DTEND if specified MUST be date-time values.
- 1522 • DTSTART MAY appear once and signifies start of the busy period.
- 1523 • Only one of DTEND or DURATION MAY appear and signify the end of the busy period.
- 1524 • If DURATION does appear in a VAVAILABILITY, then DTSTART MUST also appear in the same
1525 VAVAILABILITY.

4.1.6 AVAILABILITY

- 1527 • DTSTART and DTEND if specified MUST be date-time values.
- 1528 • DTSTART MUST appear once and signifies start of the free period.
- 1529 • Only one of DTEND or DURATION MAY appear and signify the end of the free period.

4.2 Recurrences.

- 1531 • The RECURRENCE-ID is a property of each instance of a recurring event. It is calculated from the
1532 DTSTART and the recurrence rules or added to the set by the RDATE property.
- 1533 • RDATE, EXDATE and RECURRENCE-ID must take the same form as the DTSTART. That is if
1534 DTSTART is a DATE value then the RDATE and EXDATE must be DATE. If DTSTART is a date-
1535 time the RDATE and EXDATE values must take the same form, including the same timezone.
- 1536 • Overrides to an instance are specified by completely specifying the instance with the appropriate
1537 RECURRENCE-ID property.
- 1538 • An RDATE adds an instance to the recurrence set.
- 1539 • An EXDATE deletes an instance by specifying the recurrence id(s) to be deleted. Applications
1540 SHOULD NOT specify overrides for instances so deleted.
- 1541 • The recurrence set is calculated from the RRULE and RDATES and then applying any EXDATE
1542 properties. That is EXDATE takes precedence over RDATE and the RRULE.

4.3 Alarms:

- 1544 • Alarms are typically anchored to the start or end of an event or task. This is defined by the RELATED
1545 parameter to the TRIGGER property.

4.4 Unrecognized or unsupported elements

- 1547 • A system SHOULD reject any attempt to store components which it does not support. A SYSTEM
1548 MUST advertise which components are supported through the use of the
1549 supportedCalendarComponentSet property.
- 1550 • A system MUST ignore any elements it does not understand.

1551 Appendix A. Acknowledgments

1552 The following individuals have participated in the creation of this specification and are gratefully
1553 acknowledged:

1554 Participants:

1555 Bruce Bartell, Southern California Edison
1556 Brad Benson, Trane
1557 Edward Cazalet, Individual
1558 Toby Considine, University of North Carolina at Chapel Hill
1559 William Cox, Individual
1560 Sharon Dinges, Trane
1561 Mike, Douglass, Rensselaer Polytechnic Institute
1562 Craig Gemmill, Tridium, Inc.
1563 Girish Ghatikar, Lawrence Berkeley National Laboratory
1564 Gerald Gray, Southern California Edison
1565 David Hardin, ENERNOC
1566 Gale Horst, Electric Power Research Institute (EPRI)
1567 Gershon Janssen, Individual
1568 Ed Koch, Akuacom Inc.
1569 Benoit Lepeuple, LonMark International*
1570 Carl Mattocks, CheckMi*
1571 Robert Old, Siemens AG
1572 Alexander Papaspyrou, Technische Universitat Dortmund
1573 Joshua Phillips, ISO/RTO Council (IRC)
1574 Jeremy J. Roberts, LonMark International
1575 David Thewlis, CalConnect

1576 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-
1577 Calendar Technical Committee, bridging to developing IETF standards and contributing the services
1578 definitions that make up Services in Section 4. The Technical Committee gratefully acknowledges their
1579 assistance and cooperation as well. Contributors to TC XML include:

1580 Cyrus Daboo, Apple
1581 Mike Douglass, Rensselaer Polytechnic Institute
1582 Steven Lees, Microsoft
1583 Tong Li, IBM
1584

1585 **Appendix B. Revision History**

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

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