

## CALCONNECT DOCUMENT CD1202

**Type:** Proposal  
**Title:** CalWS-SOAP SOAP Web Services Protocol for Calendaring  
**Version:** 1.0  
**Date:** 2012-02-22  
**Status:** Public Review  
**Source:** XML Technical Committee

This document incorporates by reference the CalConnect Intellectual Property Rights, Appropriate Usage, Trademarks and Disclaimer of Warranty for External (Public Review) Documents as located at

<http://www.calconnect.org/documents/disclaimerreview.pdf>

This Proposal is an in-progress working document which has been made available for a 30-day public review and comment period. Please offer any comments or suggestions to the CalConnect Public Review and Comment Mailing list. For information about this list and to subscribe please see

<http://www.calconnect.org/publicreviewdocuments.shtml>.

## RELATED SCHEMAS

This document is dependent upon the schemas for CalWS-SOAP and for xCal (RFC 6321) documented in CalConnect Code Artifacts.

<http://www.calconnect.org/artifacts/calws-soap-art.shtml>

<http://www.calconnect.org/artifacts/ical-art.shtml>

---

1 **WS-Calendar SOAP-based Services Version 1.0**

2 **Working Draft 09**

3 **15 February 2011**

4 **Technical Committee:**

5 [CalConnect TC-XML](#)

6 **Chair:**

7 [Michael Douglass \(dougim@rpi.edu\)](#) Rensselaer Polytechnic Institute

8 **Editor:**

9 [Michael Douglass \(dougim@rpi.edu\)](#) 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

22

---

## 23 Table of Contents

24	1 Introduction.....	4
25	1.1 Terminology.....	4
26	1.2 Normative References.....	4
27	1.3 Non-normative References.....	5
28	1.4 Namespace.....	5
29	2 Issues not addressed by this specification.....	6
30	2.1 Access Control.....	6
31	2.2 Provisioning.....	6
32	2.3 Copy/Move.....	6
33	2.4 Creating Collections.....	6
34	2.5 Retrieving collections.....	6
35	2.6 Setting service and resource properties.....	6
36	3 CalWS Glossary.....	7
37	3.1 Calendar Object Resource.....	7
38	3.2 Uid.....	7
39	3.3 Collections.....	7
40	3.4 Calendar Collection.....	7
41	3.5 Scheduling Calendar Collection.....	7
42	3.6 Principal Home.....	7
43	3.7 Change token.....	7
44	4 Overview of the CalWS protocol.....	8
45	4.1 Discovery.....	8
46	4.2 Properties.....	8
47	4.3 Operations.....	8
48	4.4 Calendar Object Resources.....	8
49	4.5 Timezone information.....	9
50	4.6 Error conditions.....	9
51	4.6.1 Example: error with error condition.....	9
52	5 CalWs-SOAP Messages.....	10
53	5.1 Common Elements and types.....	10
54	5.1.1 ErrorCodeType.....	11
55	5.1.2 BaseResponseType.....	13
56	6 Properties.....	14
57	6.1 childCollection.....	14
58	6.2 creationDateTime.....	14
59	6.3 displayName.....	14
60	6.4 lastModifiedDateTime.....	14
61	6.5 maxAttendeesPerInstance.....	15
62	6.6 maxDateTime.....	15
63	6.7 maxInstances.....	15
64	6.8 maxResourceSize.....	15
65	6.9 minDateTime.....	16
66	6.10 principalHome.....	16

67	6.11 resourceDescription.....	16
68	6.12 resourceOwner.....	16
69	6.13 resourceTimezoneId.....	16
70	6.14 resourceType.....	17
71	6.15 supportedCalendarComponentSet.....	17
72	6.16 supportedFeatures.....	17
73	6.17 timezoneServer.....	18
74	6.18 CalWS:privilege-set XML element.....	18
75	7 Retrieving Collection and Service Properties.....	19
76	7.1 Example - retrieving server properties:.....	19
77	8 Creating Calendar Object Resources.....	21
78	8.1 Preconditions for Calendar Object Creation.....	21
79	8.2 Example - successful addItem:.....	22
80	9 Retrieving resources.....	23
81	9.1 Example - successful fetchItem:.....	23
82	9.2 Example - unsuccessful fetchItem:.....	24
83	10 Updating resources.....	25
84	10.1 Change tokens and concurrent updates.....	29
85	10.2 Example - successful update:.....	29
86	10.3 Other updates:.....	31
87	10.4 Creating an update message.....	32
88	11 Deletion of resources.....	33
89	11.1 Example - successful deleteItem:.....	33
90	11.2 Example - unsuccessful deleteItem:.....	33
91	12 Querying calendar resources.....	35
92	12.1 Calendar Query common types.....	35
93	12.2 CompFilterType.....	35
94	12.3 PropFilterType.....	36
95	12.4 ParamFilterType.....	37
96	12.5 CalendarQueryType elements.....	38
97	12.6 Specifying data to be returned.....	38
98	12.7 Pre/postconditions for calendar queries.....	39
99	12.8 Time range limited queries.....	39
100	12.9 Example: time range limited retrieval.....	39
101	13 Free-busy queries.....	42
102	13.1 Element values .....	42
103	13.1.1 start.....	42
104	13.1.2 end.....	42
105	13.2 Examples.....	42
106	14 Multiple operations.....	45
107	Appendix A. Acknowledgments.....	46
108	Participants:.....	46
109	Appendix B. Revision History.....	47
110		

# 111 1 Introduction

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

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

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

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

## 123 1.1 Terminology

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

## 127 1.2 Normative References

- 128 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*.  
 129 IETF RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 130 **[RFC 2616]** Fielding, et al. *Hypertext Transfer Protocol -- HTTP/1.1*  
 131 <http://tools.ietf.org/html/rfc2616>
- 132 **[RFC 4791]** Daboo, et al. *Calendar Extensions to WebDAV (CalDAV)*.  
 133 <http://www.ietf.org/rfc/rfc4791.txt>.
- 134 **[draft caldav-sched]** Desruisseaux, et al. *CalDAV Scheduling extensions to WebDAV*  
 135 <http://tools.ietf.org/html/draft-desruisseaux-caldav-sched-08>
- 136 **[RFC 5545]** B. Desruisseaux, *Internet Calendaring and Scheduling Core Object*  
 137 *Specification (iCalendar)*  
 138 <http://tools.ietf.org/html/rfc5545>
- 139 **[RFC 6321]** C. Daboo, M. Douglass, S. Lees *xCal: The XML format for iCalendar*  
 140 <http://www.ietf.org/rfc/rfc6321.txt>
- 141 **[draft-timezones]** C. Daboo, M. Douglass: *Timezone Service Protocol*  
 142 <http://tools.ietf.org/html/draft-douglass-timezone-service>
- 143 **[FreeBusy Read URL]** E York. *Freebusy read URL*  
 144 [http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)  
 145 [%20V1.0.pdf](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)
- 146 **[SOAP11]** Simple Object Access Protocol (SOAP) 1.1, 8 May 2000  
 147 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- 148
- 149 **[WSDL11]** Web Services Description Language (WSDL) 1.1, 15 March 2001  
 150 <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- 151 **[WS-Calendar]** *WS-Calendar Version 1.0*. 19 January 2011. OASIS Committee Specification  
 152 [http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-](http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf)  
 153 [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).

10

### 154 1.3 Non-normative References

- 155     **[Web-Linking]**     M. Nottingham *Web linking*
- 156                             <http://tools.ietf.org/html/draft-nottingham-http-link-header>
- 157     **[WS-Addr]**             W3C Recommendation, Web Services Addressing 1.0 - Core, and Web
- 158                             Services Addressing 1.0 - SOAP Binding, 9 May 2006
- 159                             <http://www.w3.org/2002/ws/addr/>
- 160     **[WT-I-Basic]**         Basic Profile Version 1.1, 10 April 2006
- 161                             <http://www.ws-i.org/Profiles/BasicProfile-1.1-2006-04-10.html>
- 162     **[WS-I-Bind]**         Web Services-Interoperability Organization (WS-I) Simple SOAP Binding Profile
- 163                             Version 1.0, 24 August 2004
- 164                             <http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html>

### 165 1.4 Namespace

166 XML namespaces and prefixes used in this standard:

167 Table 1-1: XML Namespaces in this standard

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

168

---

## 169 **2 Issues not addressed by this specification.**

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

### 172 **2.1 Access Control**

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

### 175 **2.2 Provisioning**

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

### 179 **2.3 Copy/Move**

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

### 184 **2.4 Creating Collections**

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

### 187 **2.5 Retrieving collections**

188 This operation is currently undefined.

### 189 **2.6 Setting service and resource properties.**

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

---

## 192 **3 CalWS Glossary**

### 193 **3.1 Calendar Object Resource**

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

### 196 **3.2 Uid**

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

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

### 202 **3.3 Collections**

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

### 206 **3.4 Calendar Collection**

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

### 210 **3.5 Scheduling Calendar Collection**

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

### 213 **3.6 Principal Home**

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

### 216 **3.7 Change token**

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



## 222 4 Overview of the CalWS protocol

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

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

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

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

234 The CalWS responder MUST return either a CalWS response element within the body of another SOAP  
235 message or generate a SOAP fault. The CalWS responder MUST NOT include more than one CalWS  
236 response per SOAP message or include any additional XML elements in the SOAP body. If a CalWS  
237 responder cannot, for some reason, process a CalWS request, it MUST generate a SOAP fault. (SOAP  
238 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)

### 239 4.1 Discovery

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

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

### 245 4.2 Properties

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

### 249 4.3 Operations

250 The following operations are defined by this specification:

- 251 • Retrieval and update of service and resource properties
- 252 • Creation of a calendar object
- 253 • Retrieval of a single calendar object
- 254 • Multiget of one or more calendar objects
- 255 • Update of a calendar object
- 256 • Deletion of a calendar object
- 257 • Query
- 258 • Free-busy query
- 259 • Multiple operations

### 260 4.4 Calendar Object Resources

261 The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC 4791] section  
262 4.2. An additional constraint for CalWS is that no timezone specifications are transferred with the data.

22

## 263 4.5 Timezone information

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

## 269 4.6 Error conditions

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

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

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

### 281 4.6.1 Example: error with error condition

```
282 <?xml version="1.0" encoding="utf-8"  
283     xmlns:CW="http://docs.oasis-open.org/ws-calendar/ns/soap" ?>  
284 <CW:error>  
285   <CW:uidConflict>  
286     <CW:href>/user/mike/calendar/abcd-0123456789.ics</CW:href>  
287   </CW:uidConflict>  
288   <CW:description>Unknown property </CW:description>  
289 </CW:error>
```

## 290 5 CalWs-SOAP Messages.

291 This section describes the common elements and structure of CalWs-SOAP messages. The conventions  
292 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		

293 *Table 1: Field column descriptions*

### 294 5.1 Common Elements and types

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

297 All requests must include an href which specifies the target for the request. There is also an id attribute  
298 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.

299 *Table 2: BaseRequestType elements*

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

28

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

303 *Table 3: ErrorResponse elements*

### 304 **5.1.1 ErrorCodeType**

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

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

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

306 Table 4: ErrorCodeType definitions

307 **5.1.2 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.

308 Table 5: BaseResponseType elements

## 309 6 Properties

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

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

### 315 6.1 childCollection

316 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

317 *Table 6: ChildCollectionType fields*

318 See `resourceType` for descriptions of `CollectionType` and `Calendar CollectionType`.

### 319 6.2 creationDateTime

320 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.

321 *Table 7: CreationDateTimeType fields*

### 322 6.3 displayName

323 This property SHOULD be returned for any targeted resource.

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

324 *Table 8: DisplayNameType fields*

### 325 6.4 lastModifiedDateTime

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

40

Field	Type	#	?	Description
dateTime	dateTime	1	Y	A date-time as defined in [WS-Calendar].

327 *Table 9: LastModifiedDateType fields*328 **6.5 maxAttendeesPerInstance**

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

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

331 *Table 10: MaxAttendeesPerInstanceType fields*332 **6.6 maxDateTime**

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

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

335 *Table 11: MaxDateTimeType fields*336 **6.7 maxInstances**

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

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

339 *Table 12: MaxInstancesType fields*340 **6.8 maxResourceSize**

341 This property SHOULD be returned for the service and MAY be returned for any targeted collection  
342 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.

343 *Table 13: MaxResourceSizeType fields*



43

## 344 6.9 minDateTime

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

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

347 Table 14: MinDateTimeType fields

## 348 6.10 principalHome

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

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

351 Table 15: PrincipalHomeType fields

## 352 6.11 resourceDescription

353 Provides some descriptive text for the targeted collection.

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

354 Table 16: ResourceDescriptionType fields

## 355 6.12 resourceOwner

356 This property SHOULD be returned for any targeted resource.

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

357 Table 17: ResourceownerType fields

## 358 6.13 resourceTimezoneId

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

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

361 Table 18: ResourceTimezoneIdType fields

362 **6.14 resourceType**

363 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.

364 *Table 19: ResourceType fields*

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

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

366 *Table 20: XresourceType fields*367 **6.15 supportedCalendarComponentSet**368 This property identifies which component types the service is prepared to store. The allowable  
369 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.

370 *Table 21: SupportedCalendarComponentSetType fields*371 **6.16 supportedFeatures**372 This property SHOULD be returned for the service and MAY be returned for any targeted collection  
373 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.

374 *Table 22: SupportedFeaturesType fields*

375 **6.17 timezoneServer**

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

378 *Table 23: TimezoneServerType fields*

379 **6.18 CalWS:privilege-set XML element**

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

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

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

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

- 386 • CalWS: Read - current principal has read access  
 387 • CalWS: Write - current principal has write access

```
388 <calws:privilege-set>
389 <calws:privilege><calws:read></calws:privilege>
390 <calws:privilege><calws:write></calws:privilege>
391 </calws:privilege-set>
```

## 392 7 Retrieving Collection and Service Properties

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

395 The service properties define the global limits and defaults. Any properties defined on collections within  
396 the service hierarchy override those service defaults. The service may choose to prevent such overriding  
397 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.

398 *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

399 *Table 25: GetPropertiesResponseType fields*

### 400 7.1 Example - retrieving server properties:

```

401 >>Request
402
403 <?xml version="1.0" encoding="UTF-8"?>
404 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
405   <SOAP-ENV:Header />
406   <SOAP-ENV:Body>
407     <ns2:getProperties xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
408       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
409       <ns2:href />
410     </ns2:getProperties>
411   </SOAP-ENV:Body>
412 </SOAP-ENV:Envelope>
413
414 >>Response
415
416 <?xml version="1.0" encoding="UTF-8"?>
417 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
418   <SOAP-ENV:Header />
419   <SOAP-ENV:Body>
420     <ns2:getPropertiesResponse
421       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
422       xmlns:ns4="urn:ietf:params:xml:ns:icalendar-2.0"
423       id="0" >
424       <ns2:href />
425       <ns2:lastModifiedDate>
426         <ns2:dateTime>2012-01-04T18:21:14Z</ns2:dateTime>
427       </ns2:lastModifiedDate>
428       <ns2:supportedCalendarComponentSet>
429         <ns4:vevent />
430         <ns4:vtodo />
431         <ns4:vavailability />
432       </ns2:supportedCalendarComponentSet>

```

```
55
433     <ns2:resourceType>
434         <ns2:collection />
435     </ns2:resourceType>
436     <ns2:supportedFeatures>
437         <ns2:calendarAccessFeature />
438     </ns2:supportedFeatures>
439     <ns2:maxInstances>
440         <ns2:integer>1000</ns2:integer>
441     </ns2:maxInstances>
442     <ns2:maxResourceSize>
443         <ns2:integer>1000000</ns2:integer>
444     </ns2:maxResourceSize>
445 </ns2:getPropertiesResponse>
446 </SOAP-ENV:Body>
447 </SOAP-ENV:Envelope>
448
449
```

## 450 8 Creating Calendar Object Resources

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

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

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

456 *Table 26: AddItemType fields*

457 The service will respond with an AddItemResponseType giving either the href and change token of the  
 458 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

459 *Table 27: AddItemResponseType additional fields*

### 460 8.1 Preconditions for Calendar Object Creation

- 461 • **CalWS:target-exists:** The entity already exists.
- 462 • **CalWS:not-calendar-data:** The resource submitted MUST be a supported media type (i.e., iCalendar)  
 463 for calendar object resources;
- 464 • **CalWS:invalid-calendar-data:** The resource submitted MUST be valid data for the media type being  
 465 specified (i.e., MUST contain valid iCalendar data);
- 466 • **CalWS:invalid-calendar-object-resource:** The resource submitted in the request MUST obey all  
 467 restrictions specified in Calendar Object Resources (e.g., calendar object resources MUST NOT  
 468 contain more than one type of calendar component, calendar object resources MUST NOT specify  
 469 the iCalendar METHOD property, etc.);
- 470 • **CalWS:unsupported-calendar-component:** The resource submitted in the request MUST contain a  
 471 type of calendar component that is supported in the targeted calendar collection;
- 472 • **CalWS:uid-conflict:** The resource submitted in the request MUST NOT specify an iCalendar UID  
 473 property value already in use in the targeted calendar collection or overwrite an existing calendar  
 474 object resource with one that has a different UID property value. Servers SHOULD report the URL  
 475 of the resource that is already making use of the same UID property value in the CalWS:href  
 476 element  
 477 <!ELEMENT uid-conflict (CalWS:href)>
- 478 • **CalWS:exceeds-max-resource-size:** The resource submitted in the request MUST have an octet size  
 479 less than or equal to the value of the CalDAV:max-resource-size property value on the calendar  
 480 collection where the resource will be stored;
- 481 • **CalWS:before-min-date-time:** The resource submitted in the request MUST have all of its iCalendar  
 482 DATE or DATE-TIME property values (for each recurring instance) greater than or equal to the

- 61  
483 value of the CalDAV:min-date-time property value on the calendar collection where the resource  
484 will be stored;
- 485 • **CalWS:after-max-date-time:** The resource submitted in the request MUST have all of its iCalendar  
486 DATE or DATE-TIME property values (for each recurring instance) less than the value of the  
487 CalDAV:max-date-time property value on the calendar collection where the resource will be stored;
  - 488 • **CalWS:too-many-instances:** The resource submitted in the request MUST generate a number of  
489 recurring instances less than or equal to the value of the CalDAV: max-instances property value on  
490 the calendar collection where the resource will be stored;
  - 491 • **CalWS:too-many-attendees-per-instance:** The resource submitted in the request MUST have a  
492 number of ATTENDEE properties on any one instance less than or equal to the value of the  
493 CalDAV:max-attendees-per-instance property value on the calendar collection where the resource  
494 will be stored;

## 495 8.2 Example - successful addItem:

```

496 >>Request
497
498 <?xml version="1.0" encoding="UTF-8"?>
499 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
500   <SOAP-ENV:Header/>
501   <SOAP-ENV:Body>
502     <ns2:addItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
503       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
504       <ns2:href>/user/douglm/calendar</ns2:href>
505       <ns3:icalendar>
506         <ns3:vcalendar>
507           <ns3:components>
508             <ns3:vevent>
509               <ns3:properties>
510                 <ns3:uid>
511                   <ns3:text>1302064354993</ns3:text>
512                 </ns3:uid>
513                 <ns3:summary>
514                   <ns3:text>try this</ns3:text>
515                 </ns3:summary>
516                 <ns3:dtstart>
517                   <ns3:date-time>20110406T150000Z</ns3:date-time>
518                 </ns3:dtstart>
519                 <ns3:dtend>
520                   <ns3:date-time>20110406T160000Z</ns3:date-time>
521                 </ns3:dtend>
522               </ns3:properties>
523             </ns3:vevent>
524           </ns3:components>
525         </ns3:vcalendar>
526       </ns3:icalendar>
527     </ns2:addItem>
528   </SOAP-ENV:Body>
529 </SOAP-ENV:Envelope>
530
531 >>Response
532
533 <?xml version="1.0" encoding="UTF-8"?>
534 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
535   <SOAP-ENV:Header/>
536   <SOAP-ENV:Body>
537     <ns2:addItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
538       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
539       <ns2:status>OK</ns2:status>
540       <ns2:href>/user/douglm/calendar/1302064354993.ics</ns2:href>
541       <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
542     </ns2:addItemResponse>
543   </SOAP-ENV:Body>
544 </SOAP-ENV:Envelope>

```

## 545 9 Retrieving resources

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

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

549 *Table 28: FetchItemType fields*

550 The service will respond with a FetchItemResponseType containing either the change token, its href and  
551 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

552 *Table 29: FetchItemResponseType additional fields*

### 553 9.1 Example - successful fetchItem:

```
554 >>Request
555
556 <?xml version="1.0" encoding="UTF-8"?>
557 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
558   <SOAP-ENV:Header/>
559   <SOAP-ENV:Body>
560     <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
561       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
562       <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
563     </ns2:fetchItem>
564   </SOAP-ENV:Body>
565 </SOAP-ENV:Envelope>
566
567 >>Response
568
569 <?xml version="1.0" encoding="UTF-8"?>
570 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
571   <SOAP-ENV:Header/>
572   <SOAP-ENV:Body>
573     <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
574       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
575       <ns2:status>OK</ns2:status>
576       <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
577       <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
578       <ns3:icalendar>
579         <ns3:vcalendar>
580           <ns3:properties>
581             <ns3:prodid>
582               <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
583             </ns3:prodid>
584             <ns3:version>
585               <ns3:text>2.0</ns3:text>
586             </ns3:version>
```



```

67
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
    </ns3:properties>
    <ns3:components>
      <ns3:vevent>
        <ns3:properties>
          <ns3:created>
            <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
          </ns3:created>
          <ns3:dtend>
            <ns3:date-time>20110406T160000Z</ns3:date-time>
          </ns3:dtend>
          <ns3:dtstamp>
            <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
          </ns3:dtstamp>
          <ns3:dtstart>
            <ns3:date-time>20110406T150000Z</ns3:date-time>
          </ns3:dtstart>
          <ns3:last-modified>
            <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
          </ns3:last-modified>
          <ns3:summary>
            <ns3:text>try this</ns3:text>
          </ns3:summary>
          <ns3:uid>
            <ns3:text>1302105461170</ns3:text>
          </ns3:uid>
        </ns3:properties>
      </ns3:vevent>
    </ns3:components>
  </ns3:vcalendar>
</ns3:icalendar>
</ns2:fetchItemResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

## 620 9.2 Example - unsuccessful fetchItem:

```

621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
  >>Request
  <?xml version="1.0" encoding="UTF-8"?>
  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
      <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
        xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
        <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
      </ns2:fetchItem>
    </SOAP-ENV:Body>
  </SOAP-ENV:Envelope>

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

```

## 649 10 Updating resources

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

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

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

```

657     <icalendar>
658         <vcalendar>
659             <components>
660                 <vevent>
661                     <properties>
662                         <uid>
663                             <text>1302064354993-a</text>
664                         </uid>
665                         <summary>
666                             <text>try this</text>
667                         </summary>
668                         <dtstart>
669                             <date-time>2011-07-18T15:00:00Z</date-time>
670                         </dtstart>
671                         <dtend>
672                             <date-time>2011-07-18T16:00:00Z</date-time>
673                         </dtend>
674                     </properties>
675                 </vevent>
676             </components>
677         </vcalendar>
678     </icalendar>

```

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

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

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

- 687 • Remove - components, properties or parameters
- 688 • Add - components, properties or parameters
- 689 • Change - property or parameter values

690 Removals MUST be processed ahead of additions

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

73

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

695 *Table 30: UpdateItem Type fields*

696 The ComponentsSelectionType contains three repeating child elements. The first allows for selection of  
697 nested components which can then be updated. The next allows addition of entire components and the  
698 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

699 *Table 31: ComponentsSelectionType fields*

700 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

701 *Table 32: PropertiesSelectionType fields*

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

76

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

704 *Table 33: ParametersSelectionType fields*

705 Each of these refers to a reference type. These either provide a complete entity for addition or identify the  
706 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.

707 *Table 34: ComponentReferenceType fields*

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

708 *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.

709 *Table 36: ParameterReferenceType fields*

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

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

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

718 The optional components selection allows selection and possible updates to the nested icalendar  
719 components element of the component. An iCalendar components element cannot take a value so the  
720 only updates possible are addition and removal of components. Nested components may be selected for  
721 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

722 *Table 37: ComponentSelectionType fields*

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

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

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

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

729 *Table 38: PropertySelectionType fields*

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

732 The optional change element allows a change to the value of the parameter. The new value is specified  
733 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

734 *Table 39: ParameterSelectionType fields*

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

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

737 Table 40: UpdateItemResponseType additional fields

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

## 739 10.1 Change tokens and concurrent updates

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

## 751 10.2 Example - successful update:

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

```

753 <ns3:icalendar>
754   <ns3:vcalendar>
755     <ns3:components>
756       <ns3:vevent>
757         <ns3:properties>
758           <ns3:uid>
759             <ns3:text>1302064354993-a</ns3:text>
760           </ns3:uid>
761           <ns3:summary>
762             <ns3:text>try this</ns3:text>
763           </ns3:summary>
764           <ns3:dtstart>
765             <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
766           </ns3:dtstart>
767           <ns3:dtend>
768             <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
769           </ns3:dtend>
770         </ns3:properties>
771       </ns3:vevent>
772     </ns3:components>
773   </ns3:vcalendar>
774 </ns3:icalendar>

```

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

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

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

```

781 >>Request
782
783 <?xml version="1.0" encoding="UTF-8"?>
784 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
785   <SOAP-ENV:Header/>
786   <SOAP-ENV:Body>
787     <ns2:updateItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"

```

```

85      xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
788      <ns2:href>/user/douglm/calendar/1302064354993-a.ics</ns2:href>
789      <ns2:changeToken>"20110802T032608Z-0"      null</ns2:changeToken>
790      <ns2:select>
791      <ns3:vcalendar/>
792      <ns2:components>
793      <ns2:component>
794      <ns3:vevent>
795      <ns3:properties>
796      <ns3:uid>
797      <ns3:text>1302064354993-a</ns3:text>
798      </ns3:uid>
799      </ns3:properties>
800      </ns3:vevent>
801      <ns2:properties>
802      <ns2:property>
803      <ns3:dtstart>
804      <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
805      </ns3:dtstart>
806      <ns2:parameters>
807      <ns2:add>
808      <ns3:tzid>
809      <ns3:text>America/New_York</ns3:text>
810      </ns3:tzid>
811      </ns2:add>
812      </ns2:parameters>
813      <ns2:change>
814      <ns3:dtstart>
815      <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
816      </ns3:dtstart>
817      </ns2:change>
818      </ns2:property>
819      <ns2:property>
820      <ns3:summary>
821      <ns3:text>try this</ns3:text>
822      </ns3:summary>
823      <ns2:change>
824      <ns3:summary>
825      <ns3:text>A changed summary - again and again and again</ns3:text>
826      </ns3:summary>
827      </ns2:change>
828      </ns2:property>
829      <ns2:add>
830      <ns3:categories>
831      <ns3:text>newcategory-2</ns3:text>
832      <ns3:text>resources</ns3:text>
833      <ns3:text>paper</ns3:text>
834      </ns3:categories>
835      </ns2:add>
836      </ns2:properties>
837      </ns2:component>
838      </ns2:components>
839      </ns2:select>
840      </ns2:updateItem>
841      </SOAP-ENV:Body>
842      </SOAP-ENV:Envelope>
843
844  >>Response
845
846  <?xml version="1.0" encoding="UTF-8"?>
847  <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
848  <SOAP-ENV:Header/>
849  <SOAP-ENV:Body>
850  <ns2:updateItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
851  xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
852  id="0">
853  <ns2:status>OK</ns2:status>
854  </ns2:updateItemResponse>
855  </SOAP-ENV:Body>
856  </SOAP-ENV:Envelope>
857

```

## 858 10.3 Other updates:

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

- 861 • Addition of properties
- 862 • Removal of properties
- 863 • Addition of parameters to properties
- 864 • Removal of parameters from properties
- 865 • Changing parameter values.

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

```

869     <ns2:properties>
870       <ns2:property>
871         <ns3:dtstart>
872           <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
873         </ns3:dtstart>
874         <ns2:parameters>
875           <ns2:add>
876             <ns3:tzid>
877               <ns3:text>America/New_York</ns3:text>
878             </ns3:tzid>
879           </ns2:add>
880         </ns2:parameters>
881         <ns2:change>
882           <ns3:dtstart>
883             <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
884           </ns3:dtstart>
885         </ns2:change>
886       </ns2:property>
887     </ns2:properties>

```

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

```

889     <ns2:properties>
890       <ns2:add>
891         <ns3:categories>
892           <ns3:text>paper</ns3:text>
893         </ns3:categories>
894       </ns2:add>
895       <ns2:add>
896         <ns3:categories>
897           <ns3:text>resources</ns3:text>
898         </ns3:categories>
899       </ns2:add>
900     </ns2:properties>

```

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

```

902     <ns2:properties>
903       <ns2:remove>
904         <ns3:dtend>
905           <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
906         </ns3:dtend>
907       </ns2:remove>
908       <ns2:add>
909         <ns3:duration>
910           <ns3:duration>PT1H</ns3:duration>
911         </ns3:duration>
912       </ns2:add>
913     </ns2:properties>

```

914 In this example we change the dtstart timezone identifier.

```

915     <ns2:properties>
916       <ns2:property>
917         <ns3:dtstart>
918           <ns3:parameters>

```



91  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938

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

939

## 940 10.4 Creating an update message.

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

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

950 These properties are

- 951 • categories
- 952 • exdate
- 953 • freebusy
- 954 • rdate

955 This normalization can take place before comparison.

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

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

## 961 11 Deletion of resources

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

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

965 *Table 41: DeleteItem Type fields*

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

### 968 11.1 Example - successful deleteItem:

```

969 >>Request
970
971 <?xml version="1.0" encoding="UTF-8"?>
972 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
973   <SOAP-ENV:Header/>
974   <SOAP-ENV:Body>
975     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
976       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
977       <ns2:href>/user/douglm/calendar/1302620814655.ics</ns2:href>
978     </ns2:deleteItem>
979   </SOAP-ENV:Body>
980 </SOAP-ENV:Envelope>
981
982 >>Response
983
984 <?xml version="1.0" encoding="UTF-8"?>
985 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
986   <SOAP-ENV:Header/>
987   <SOAP-ENV:Body>
988     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
989       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
990       <ns2:status>OK</ns2:status>
991     </ns2:deleteItemResponse>
992   </SOAP-ENV:Body>
993 </SOAP-ENV:Envelope>

```

### 994 11.2 Example - unsuccessful deleteItem:

```

995 >>Request
996
997 <?xml version="1.0" encoding="UTF-8"?>
998 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
999   <SOAP-ENV:Header/>
1000   <SOAP-ENV:Body>
1001     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1002       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1003       <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
1004     </ns2:deleteItem>
1005   </SOAP-ENV:Body>
1006 </SOAP-ENV:Envelope>
1007
1008 >>Response
1009
1010 <?xml version="1.0" encoding="UTF-8"?>
1011 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

```

```
97
1012 <SOAP-ENV:Header/>
1013 <SOAP-ENV:Body>
1014   <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1015     xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1016     <ns2:status>Error</ns2:status>
1017     <ns2:errorResponse>
1018       <ns2:targetDoesNotExist/>
1019     </ns2:errorResponse>
1020   </ns2:deleteItemResponse>
1021 </SOAP-ENV:Body>
1022 </SOAP-ENV:Envelope>
```

## 1023 12 Querying calendar resources

1024 Querying provides a mechanism by which information can be obtained from the service through possibly  
 1025 complex queries. A skeleton icalendar entity can be provided to limit the amount of information returned to  
 1026 the client. A query takes the parts

- 1027 • Limitations on the data returned
- 1028 • Selection of the data
- 1029 • Optional timezone id for floating time calculations.

### 1030 12.1 Calendar Query common types

1031 The UTCTimeRangeType is used in a number of places to define a time range within which components  
 1032 must appear or property values must lie. The values are UTC time-date, the start is inclusive and the end  
 1033 is exclusive.

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

1034 *Table 42: UTCTimeRangeType elements*

1035 The TextMatchType is used to match text values in properties and parameters. The collation attribute  
 1036 species a collation as defined in Error: Reference source not found.

1037 Servers are REQUIRED to support the "i;ascii-casemap" and "i;octet" collations which provide a basic  
 1038 case insensitive and case sensitive match respectively.

1039 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 Error: Reference source not found. "
#negate-condition	boolean	0..1	N	if "true" negates the condition

1040 *Table 43: TextMatchType attributes*

### 1041 12.2 CompFilterType

1042 This type defines a search query for the calendar query operation. It specifies the component types to  
 1043 return, absence tests or basic matching operations on properties and time ranges.

1044 The top level comp-filter element (which must match a vcalendar component may contain zero or more  
 1045 comp-filter elements to match events, tasks or other contained components. These in turn may contain  
 1046 further nested comp-filter elements to match further levels of nested components.

1047 Each may also contain prop-filter elements to test for the absence of properties or to match values.

1048 Only logical conjunctions are supported, that is, all elements of a comp-filter must match for the  
 1049 expression to match.

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

1050 *Table 44: CompFilterType elements*1051 **12.3 PropFilterType**

1052 The prop-filter element may test for the absence of a property or match values or specify zero or more  
 1053 ParamFilterType elements to match against parameters.

1054 Only logical conjunctions are supported, that is, all elements must match for the full expression to match.

106

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

1055 *Table 45: PropFilterType elements*1056 **12.4 ParamFilterType**

1057 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	

1058 *Table 46: ParamFilterType elements*

## 12.5 CalendarQueryType elements

Field	Type	#	?	Description
href	string	1	Y	Identify the target of the request. "/" for the service.
allprop	empty	0..1	N	If present specifies all properties should be returned One or none of allprop or icalendar
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

1060 Table 47: CalendarQueryType elements

## 1061 12.6 Specifying data to be returned

1062 This is achieved by specifying one of the following

- 1063 • allprop: return all properties and calendar data. (some properties are specified as not being part of the  
1064 allprop set so are not returned)

- 112  
1065 • Set the icalendar element. This is an icalendar valueless pattern entity which provides a map of the  
1066 components and properties to be returned. Neither the pattern nor the returned result need to be  
1067 valid icalendar entities in that required properties may be absent if unselected.

## 1068 12.7 Pre/postconditions for calendar queries

1069 The preconditions as defined in [RFC 4791] Section 7.8 apply here. CalWS errors may be reported by the  
1070 service when preconditions or postconditions are violated.

## 1071 12.8 Time range limited queries.

1072 Time-range limited retrieval has some special characteristics. The simplest case is a single event or task  
1073 which overlaps the requested time-period. Recurring items and other components such as alarms  
1074 complicate the picture.

## 1075 12.9 Example: time range limited retrieval

1076 This example shows the time-range limited retrieval from a calendar which results in 2 events, one a  
1077 recurring event and one a simple non-recurring event.

```
1078 >> Request <<
1079
1080 <?xml version="1.0" encoding="UTF-8"?>
1081 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1082   <SOAP-ENV:Header/>
1083   <SOAP-ENV:Body>
1084     <ns2:calendarQuery xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1085       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1086       <ns2:href>/user/douglm/calendar</ns2:href>
1087       <ns3:icalendar>
1088         <ns3:vcalendar>
1089           <ns3:components>
1090             <ns3:vevent>
1091               <ns3:properties>
1092                 <ns3:summary/>
1093                 <ns3:dtstart/>
1094                 <ns3:dtend/>
1095                 <ns3:duration/>
1096                 <ns3:uid/>
1097                 <ns3:recurrence-id/>
1098                 <ns3:rrule/>
1099                 <ns3:rdate/>
1100                 <ns3:exdate/>
1101               </ns3:properties>
1102             </ns3:vevent>
1103           </ns3:components>
1104         </ns3:vcalendar>
1105       </ns3:icalendar>
1106       <ns2:filter>
1107         <ns2:compFilter test="anyof">
1108           <ns3:vcalendar />
1109           <ns2:compFilter>
1110             <ns3:vevent />
1111             <ns2:time-range end="20110430T040000Z" start="20110401T040000Z"/>
1112           </ns2:compFilter>
1113         </ns2:filter>
1114       </ns2:calendarQuery>
1115     </SOAP-ENV:Body>
1116   </SOAP-ENV:Envelope>
1117
1118 >> Response <<
1119
1120 <?xml version="1.0" encoding="UTF-8"?>
1121 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1122   <SOAP-ENV:Header/>
1123   <SOAP-ENV:Body>
```



```

115 <ns2:calendarQueryResponse
1124
1125     xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1126     xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1127 <ns2:status>OK</ns2:status>
1128 <ns2:response>
1129   <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
1130   <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
1131   <ns2:propstat>
1132     <ns2:prop>
1133       <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1134         <ns3:icalendar>
1135           <ns3:vcalendar>
1136             <ns3:properties>
1137               <ns3:prodid>
1138                 <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1139               </ns3:prodid>
1140               <ns3:version>
1141                 <ns3:text>2.0</ns3:text>
1142               </ns3:version>
1143             </ns3:properties>
1144             <ns3:components>
1145               <ns3:vevent>
1146                 <ns3:properties>
1147                   <ns3:dtend>
1148                     <ns3:date-time>20110406T160000Z</ns3:date-time>
1149                   </ns3:dtend>
1150                   <ns3:dtstart>
1151                     <ns3:date-time>20110406T150000Z</ns3:date-time>
1152                   </ns3:dtstart>
1153                   <ns3:summary>
1154                     <ns3:text>try this</ns3:text>
1155                   </ns3:summary>
1156                   <ns3:uid>
1157                     <ns3:text>1302105461170</ns3:text>
1158                   </ns3:uid>
1159                 </ns3:properties>
1160               </ns3:vevent>
1161             </ns3:components>
1162           </ns3:vcalendar>
1163         </ns3:icalendar>
1164       </ns2:calendar-data>
1165     </ns2:prop>
1166   <ns2:status>OK</ns2:status>
1167 </ns2:propstat>
1168 </ns2:response>
1169 <ns2:response>
1170   <ns2:href>/user/douglm/calendar/CAL-00f1fc61-2f021bca-012f-022947f8-
1171 00000006.ics</ns2:href>
1172   <ns2:changeToken>"20110405T140920Z-0"</ns2:changeToken>
1173   <ns2:propstat>
1174     <ns2:prop>
1175       <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1176         <ns3:icalendar>
1177           <ns3:vcalendar>
1178             <ns3:properties>
1179               <ns3:prodid>
1180                 <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1181               </ns3:prodid>
1182               <ns3:version>
1183                 <ns3:text>2.0</ns3:text>
1184               </ns3:version>
1185             </ns3:properties>
1186             <ns3:components>
1187               <ns3:vevent>
1188                 <ns3:properties>
1189                   <ns3:duration>
1190                     <ns3:duration>PT1H</ns3:duration>
1191                   </ns3:duration>
1192                   <ns3:dtstart>
1193                     <ns3:parameters>
1194                       <ns3:tzid>

```

```

118         <ns3:text>America/New_York</ns3:text>
1195     </ns3:tzid>
1196     </ns3:parameters>
1197     <ns3:date-time>20110412T110000</ns3:date-time>
1198 </ns3:dtstart>
1200 <ns3:summary>
1201     <ns3:text>Test recurring event</ns3:text>
1202 </ns3:summary>
1203 <ns3:uid>
1204     <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1205 00000006demobedework@mysite.edu</ns3:text>
1206 </ns3:uid>
1207 <ns3:rrule>
1208     <ns3:recur>
1209         <ns3:freq>WEEKLY</ns3:freq>
1210         <ns3:count>2</ns3:count>
1211         <ns3:interval>1</ns3:interval>
1212     </ns3:recur>
1213 </ns3:rrule>
1214 </ns3:properties>
1215 </ns3:vevent>
1216 <ns3:vevent>
1217     <ns3:properties>
1218         <ns3:recurrence-id>
1219             <ns3:parameters>
1220                 <ns3:tzid>
1221                     <ns3:text>America/New_York</ns3:text>
1222                 </ns3:tzid>
1223                 </ns3:parameters>
1224                 <ns3:date-time>20110419T150000Z</ns3:date-time>
1225             </ns3:recurrence-id>
1226             <ns3:duration>
1227                 <ns3:duration>PT1H</ns3:duration>
1228             </ns3:duration>
1229             <ns3:dtstart>
1230                 <ns3:parameters>
1231                     <ns3:tzid>
1232                         <ns3:text>America/New_York</ns3:text>
1233                     </ns3:tzid>
1234                     </ns3:parameters>
1235                 <ns3:date-time>20110419T120000</ns3:date-time>
1236             </ns3:dtstart>
1237             <ns3:summary>
1238                 <ns3:text>Test recurring event</ns3:text>
1239             </ns3:summary>
1240             <ns3:uid>
1241                 <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1242 00000006demobedework@mysite.edu</ns3:text>
1243             </ns3:uid>
1244         </ns3:properties>
1245     </ns3:vevent>
1246 </ns3:components>
1247 </ns3:vcalendar>
1248 </ns3:icalendar>
1249 </ns2:calendar-data>
1250 </ns2:prop>
1251     <ns2:status>OK</ns2:status>
1252 </ns2:propstat>
1253 </ns2:response>
1254 </ns2:calendarQueryResponse>
1255 </SOAP-ENV:Body>
1256 </SOAP-ENV:Envelope>
1257

```

## 1258 13 Free-busy queries

1259 Freebusy queries are used to obtain freebusy information for a principal. The result contains information  
1260 only for events to which the current principal has sufficient access and may be affected by components  
1261 and rules available only to the server (for instance office hours availability).

1262 These queries are carried out by using a CalWS-SOAP freebusyReport request with an href specifying a  
1263 principal. The freebusyReport request is not valid when the href specifies any entity other than a principal.

1264 The query follows the specification defined in [FreeBusy Read URL] with certain limitations. As an  
1265 authenticated user to the CalWS service scheduling read-freebusy privileges must have been granted. As  
1266 an unauthenticated user equivalent access must have been granted to unauthenticated users.

1267 Freebusy information is returned by default as xcalendar vfreebusy components, as defined by [RFC  
1268 6321]. Such a component is not meant to conform to the requirements of VFREEBUSY components in  
1269 Error: Reference source not found. The VFREEBUSY component SHOULD conform to section "4.6.4  
1270 Free/Busy Component" of [RFC 5545]. A client SHOULD ignore the ORGANIZER field.

1271 Since a Freebusy query can only refer to a single user, a client will already know how to match the result  
1272 component to a user. A server MUST only return a single vfreebusy component.

### 1273 13.1 Element values

1274 Three values are provided: href; start; end. Only the href is required. The start and end are in XML UTC  
1275 date/time format and are interpreted as follows:

#### 1276 13.1.1 start

1277 **Default:** If omitted the default value is left up to the server. It may be the current day, start of the  
1278 current month, etc.

1279 **Description:** Specifies the start date for the Freebusy data. The server is free to ignore this value  
1280 and return data in any time range. The client must check the data for the returned time range.

1281 **Format:** An XML UTC date-time

1282 **Example:**

1283 `2011-12-01T10:15:00Z`

1284 **Notes:** Specifying only a start date/time without specifying an end-date/time or period should be  
1285 interpreted as in [RFC 5545]. The effective period should cover the remainder of that day.

#### 1286 13.1.2 end

1287 **Default:** Same as start

1288 **Description:** Specifies the end date for the Freebusy data. The server is free to ignore this value.

1289 **Format:** Same as start

1290 **Example:** Same as start

1291 The server is free to ignore the start, end and period parameters. It is recommended that the server return  
1292 at least 6 weeks of data from the current day.

1293 A client MUST check the time range in the response as a server may return a different time range than  
1294 the requested range.

## 1295 13.2 Examples

1296 The following is an unsuccessful request targeting an invalid resource.

```

124
1297 >> Request <<
1298
1299 <?xml version="1.0" encoding="UTF-8"?>
1300 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1301   <SOAP-ENV:Header/>
1302   <SOAP-ENV:Body>
1303     <ns2:freebusyReport
1304       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1305       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1306       <ns2:href>/user/douglm/calendar</ns2:href>
1307       <ns2:time-range>
1308         <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1309         <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1310       </ns2:time-range>
1311     </ns2:freebusyReport>
1312   </SOAP-ENV:Body>
1313 </SOAP-ENV:Envelope>
1314
1315 >> Response <<
1316
1317 <?xml version="1.0" encoding="UTF-8"?>
1318 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1319   <SOAP-ENV:Header/>
1320   <SOAP-ENV:Body>
1321     <ns2:freebusyReportResponse
1322       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1323       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1324       <ns2:status>Error</ns2:status>
1325       <ns2:message>Only principal href supported</ns2:message>
1326     </ns2:freebusyReportResponse>
1327   </SOAP-ENV:Body>
1328 </SOAP-ENV:Envelope>

```

1329 The following is an example of a request to retrieve Freebusy data for a user:

```

1330 >> Request <<
1331
1332 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1333   <SOAP-ENV:Header/>
1334   <SOAP-ENV:Body>
1335     <ns2:freebusyReport
1336       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1337       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1338       <ns2:href>/principals/users/douglm</ns2:href>
1339       <ns2:time-range>
1340         <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1341         <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1342       </ns2:time-range>
1343     </ns2:freebusyReport>
1344   </SOAP-ENV:Body>
1345 </SOAP-ENV:Envelope>
1346
1347 >> Response <<
1348
1349 <?xml version="1.0" encoding="UTF-8"?>
1350 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1351   <SOAP-ENV:Header/>
1352   <SOAP-ENV:Body>
1353     <ns2:freebusyReportResponse
1354       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1355       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1356       <ns2:status>OK</ns2:status>
1357       <ns3:icalendar>
1358         <ns3:vcalendar>
1359           <ns3:properties>
1360             <ns3:prodid>
1361               <ns3:text>//Bedework.org//Bedework V3.7//EN</ns3:text>
1362             </ns3:prodid>
1363             <ns3:version>
1364               <ns3:text>2.0</ns3:text>
1365             </ns3:version>
1366           </ns3:properties>

```

127  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418

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

1419

## 1420 **14 Multiple operations**

1421 Each of the previously described operations acts upon a single entity or resource only. Frequently we  
 1422 have the need to update an interconnected set of entities so that we maintain the consistency of the  
 1423 structure. This requires an atomic operation which can successfully update all the entities or roll back the  
 1424 operation on failure.

1425 The MultiOpType operation provides such a feature. It is essentially a wrapper around any of the other  
 1426 operations which guarantees the success of the entire set or a roll back. Using the id attribute for  
 1427 requests, each individual response can be located in the result.

1428 The MultiOpType request takes the following elements

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

1429 Table 48: MultiOpType elements

1430 The response type is also simple containing a single element containing all the responses.

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

1431 Table 49: MultiOpResponseType elements

1432

---

## 1433 Appendix A. Acknowledgments

1434 The following individuals have participated in the creation of this specification and are gratefully  
1435 acknowledged:

### 1436 Participants:

1437 Bruce Bartell, Southern California Edison  
1438 Brad Benson, Trane  
1439 Edward Cazalet, Individual  
1440 Toby Considine, University of North Carolina at Chapel Hill  
1441 William Cox, Individual  
1442 Sharon Dinges, Trane  
1443 Mike, Douglass, Rensselaer Polytechnic Institute  
1444 Craig Gemmill, Tridium, Inc.  
1445 Girish Ghatikar, Lawrence Berkeley National Laboratory  
1446 Gerald Gray, Southern California Edison  
1447 David Hardin, ENERNOC  
1448 Gale Horst, Electric Power Research Institute (EPRI)  
1449 Gershon Janssen, Individual  
1450 Ed Koch, Akuacom Inc.  
1451 Benoit Lepeuple, LonMark International\*  
1452 Carl Mattocks, CheckMi\*  
1453 Robert Old, Siemens AG  
1454 Alexander Papaspyrou, Technische Universitat Dortmund  
1455 Joshua Phillips, ISO/RTO Council (IRC)  
1456 Jeremy J. Roberts, LonMark International  
1457 David Thewlis, CalConnect

1458 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-  
1459 Calendar Technical Committee, bridging to developing IETF standards and contributing the services  
1460 definitions that make up Services in Section 4. The Technical Committee gratefully acknowledges their  
1461 assistance and cooperation as well. Contributors to TC XML include:

1462 Cyrus Daboo, Apple  
1463 Mike Douglass, Rensselaer Polytechnic Institute  
1464 Steven Lees, Microsoft  
1465 Tong Li, IBM  
1466

1467 **Appendix B. Revision History**

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



139

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

1468