



WS-Calendar SOAP-based Services Version 1.0

**Committee Specification ~~Draft 02/~~
~~Public Review Draft 0201~~**

~~09 November 2012~~

28 February 2013

Specification URIs

This version:

<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd02/ws-calendar-soap-v1.0-esprd02.pdf>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/cs01/ws-calendar-soap-v1.0-cs01.pdf> (Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/cs01/ws-calendar-soap-v1.0-cs01.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/cs01/ws-calendar-soap-v1.0-cs01.odt>

Previous version:

(Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd02/ws-calendar-soap-v1.0-esprd02.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd02/ws-calendar-soap-v1.0-esprd02.odt>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd02/ws-calendar-soap-v1.0-esprd02.pdf> (Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd02/ws-calendar-soap-v1.0-esprd02.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd02/ws-calendar-soap-v1.0-esprd02.odt>

Previous version:

<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd01/ws-calendar-soap-v1.0-esprd01.pdf> (Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd01/ws-calendar-soap-v1.0-esprd01.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/esprd01/ws-calendar-soap-v1.0-esprd01.odt>

Latest version:

<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/ws-calendar-soap-v1.0.pdf>
(Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/ws-calendar-soap-v1.0.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/ws-calendar-soap-v1.0.odt>

Technical Committee:

OASIS Web Services Calendar (WS-Calendar) TC

Chair:

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

Editor:

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

Related work:

This specification is related to:

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

Abstract:

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

Status:

This document was last revised or approved by the OASIS Web Services Calendar (WS-Calendar) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this Work Product to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at <http://www.oasis-open.org/committees/ws-calendar/>.

For information on whether any patents have been disclosed that may be essential to implementing this Work Product, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/ws-calendar/ipr.php>).

Citation format:

When referencing this Work Product the following citation format should be used:

[WS-Cal-SOAP]

WS-Calendar SOAP-based Services Version 1.0. ~~09 November 2012-28 February 2013~~. OASIS Committee Specification ~~Draft 02 / Public Review Draft 02-01~~. <http://docs.oasis-open.org/ws-calendar/ws-calendar-soap/v1.0/csprd021/ws-calendar-soap-v1.0-csprd021.html>.

Notices

Copyright © OASIS Open 201~~2~~3. All Rights Reserved.

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

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

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

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

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

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

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

The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

Table of Contents

1	Introduction	7
1.1	Terminology	7
1.2	Normative References Normative References	7
1.3	Non-normative References Non-normative References	8
1.4	Namespace.....	8
2	Issues not addressed by this specification.	9
2	Issues not addressed by this specification.	9
2.1	Access Control Access Control	9
2.2	Provisioning	9
2.3	Copy/Move Copy/Move	9
2.4	Creating Collections Creating Collections	9
2.5	Retrieving collections Retrieving collections.....	9
2.6	Setting service and resource properties.	9
2.6	Setting service and resource properties.	9
3	CalWS Glossary CalWS Glossary	10
3.1.1	Calendar Object Resource Calendar Object Resource	10
3.1.2	Uid	10
3.1.3	Collections	10
3.1.4	Calendar Collection Calendar Collection	10
3.1.5	Scheduling Calendar Collection Scheduling Calendar Collection.....	10
3.1.6	Principal Home Principal Home.....	10
3.1.7	Change token Change token.....	10
4	Basic Calendar Access	11
4	Basic Calendar Access	11
4.1	Overview of the CalWS protocol Overview of the CalWS protocol	11
4.1.1	Discovery	11
4.1.2	Properties	11
4.1.3	Operations	11
4.1.4	Calendar Object Resources Calendar Object Resources.....	12
4.1.5	Timezone information Timezone information	12
4.1.6	Error conditions Error conditions	12
4.1.6.1	Example: error with error condition Example: error with error condition.....	12
4.2	CalWS-SOAP Messages CalWS-SOAP Messages	12
4.2.1	Common Elements and types Common Elements and types.....	13
4.2.1.1	ErrorCodeType	13
4.2.1.2	BaseResponseType.....	15
4.3	Properties	15
4.3.1	childCollection	15
4.3.2	creationDateTime	15
4.3.3	displayName.....	16
4.3.4	lastModifiedDateTime.....	16
4.3.5	maxAttendeesPerInstance	16
4.3.6	maxDateTime	16
4.3.7	maxInstances	16
4.3.8	maxResourceSize	16

4.3.9	minDateTime	17
4.3.10	principalHome	17
4.3.11	resourceDescription	17
4.3.12	resourceOwner	17
4.3.13	resourceTimezoneId.....	17
4.3.14	resourceType	17
4.3.15	supportedCalendarComponentSet.....	18
4.3.16	supportedFeatures	18
4.3.17	timezoneServer	18
4.3.18	CalWS:privilege-set XML element CalWS:privilege-set XML element	19
4.4	Retrieving Collection and Service Properties	19
4.4	Retrieving Collection and Service Properties	19
4.4.1	Example—retrieving server properties: Example - retrieving server properties:	19
4.5	Creating Calendar Object Resources	20
4.5	Creating Calendar Object Resources	20
4.5.1	Preconditions for Calendar Object Creation Preconditions for Calendar Object Creation ...	20
4.5.2	Example—successful addItem: Example - successful addItem:	21
4.6	Retrieving resources Retrieving resources	22
4.6.1	Example—successful fetchItem: Example - successful fetchItem:	22
4.6.2	Example—unsuccessful fetchItem: Example - unsuccessful fetchItem:	23
4.7	Updating resources Updating resources	23
4.7.1	Change tokens and concurrent updates Change tokens and concurrent updates	27
4.7.2	Example—successful update: Example - successful update:	27
4.7.3	Other updates: Other updates:	29
4.7.4	Creating an update message: Creating an update message:	30
4.8	Deletion of resources Deletion of resources	30
4.8.1	Example—successful deleteItem: Example - successful deleteItem:	30
4.8.2	Example—unsuccessful deleteItem: Example - unsuccessful deleteItem:	31
4.9	Querying calendar resources Querying calendar resources	31
4.9.1	Calendar Query common types Calendar Query common types	31
4.9.2	CompFilterType	32
4.9.3	PropFilterType	32
4.9.4	ParamFilterType	33
4.9.5	CalendarQueryType elements CalendarQueryType elements	34
4.9.6	Specifying data to be returned Specifying data to be returned	34
4.9.7	Pre/postconditions for calendar queries Pre/postconditions for calendar queries	34
4.9.8	Time range limited queries: Time range limited queries:	35
4.9.9	Example: time range limited retrieval Example: time range limited retrieval	35
4.10	Free-busy queries Free-busy queries	37
4.10.1	Element values Element values	38
4.10.1.1	start.....	38
4.10.1.2	end.....	38
4.10.2	Examples.....	38
4.11	Multiple operations Multiple operations	40
5	Conformance.....	41
5.1	Start, end and duration in calendar components	41
5.1.1	Updating, transporting and maintaining start, and and duration.	41

5.1	<u>Start, end and duration in calendar components</u>	41
5.1.1	<u>Updating, transporting and maintaining start, and and duration.</u>	41
5.1.2	VEVENT:	41
5.1.3	VTODO:	41
5.1.4	VJOURNAL:	41
5.1.5	VAVAILABILITY	42
5.1.6	AVAILABILITY	42
5.2	Recurrences.	42
5.3	Alarms:	42
5.4	Unrecognized or unsupported elements	42
5.4	<u>Unrecognized or unsupported elements</u>	42
Appendix A	Acknowledgments	43
Appendix B	Revision History <u>Revision History</u>	44

1 Introduction

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

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

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

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

1.1 Terminology

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

1.2 Normative References

- [RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- [RFC 2616] Fielding, et al. *Hypertext Transfer Protocol -- HTTP/1.1*. <http://tools.ietf.org/html/rfc2616>
- [RFC 4791] Daboo, et al. *Calendaring Extensions to WebDAV (CalDAV)*. <http://www.ietf.org/rfc/rfc4791>
- [RFC 6638] Desruisseaux, et al. *CalDAV Scheduling extensions to CalDAV*. <http://tools.ietf.org/html/rfc6638>
- [RFC 5545] B. Desruisseaux, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. <http://tools.ietf.org/html/rfc5546>
- [RFC 5546] C. Daboo, *iCalendar Transport-Independent Interoperability Protocol (iTIP)*. <http://tools.ietf.org/html/rfc5545>
- [RFC 6321] C. Daboo, M. Douglass, S. Lees *xCal: The XML format for iCalendar*. <http://www.ietf.org/rfc/rfc6321>
- [draft-timezones] C. Daboo, M. Douglass: *Timezone Service Protocol*. <http://tools.ietf.org/html/draft-douglass-timezone-service>
- [FreeBusy Read URL] E York. *Freebusy read URL*. <http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf>
- [SOAP11] Simple Object Access Protocol (SOAP) 1.1, 8 May 2000. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [WSDL11] Web Services Description Language (WSDL) 1.1, 15 March 2001. <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [WS-Calendar] *WS-Calendar Version 1.0*. 19 January 2011. OASIS Committee Specification. <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf>

18 **1.3 Non-normative References**

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

19 **1.4 Namespace**

20 XML namespaces and prefixes used in this standard:

21 Table 1-1: XML Namespaces in this standard

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

22

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

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

26 **2.1 Access Control**

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

29 **2.2 Provisioning**

30 The protocol will not provide any explicit provisioning operations. If it is possible to authenticate or address a prin-
31 cipals calendar resources then they **MUST** be automatically created if necessary or appropriate

32 **2.3 Copy/Move**

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

36 **2.4 Creating Collections**

37 We will not address the issue of creating collections within the address space. The initial set is created by provision-
38 ing.

39 **2.5 Retrieving collections**

40 This operation is currently undefined.

41 **2.6 Setting service and resource properties.**

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

44 3 CalWS Glossary

45 3.1.1 Calendar Object Resource

46 A calendar object resource is an event, meeting or a task. Attachments are resources but NOT calendar object re-
47 sources. An event or task with overrides is a single calendar resource entity.

48 3.1.2 Uid

49 The UID of an event is defined in [RFC 5545] as a "persistent, globally unique identifier for the calendar compo-
50 nent". It is in fact, slightly more complicated in that all overrides to a recurring event have the same UID as the mas-
51 ter event. Copies of a meeting invitation sent to attendees must also have the same UID.

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

54 3.1.3 Collections

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

58 3.1.4 Calendar Collection

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

62 3.1.5 Scheduling Calendar Collection

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

65 3.1.6 Principal Home

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

68 3.1.7 Change token

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

73 4 Basic Calendar Access

74 This section defines properties, messages and operations sufficient to provide basic access and operations on a calendar store. These are sufficient to store, retrieve and update calendaring entities and to obtain various reports on the
75 current state of the store.
76

77 Any service supporting this protocol **MUST** return a calendarAccessFeature element in the supportedFeatures property in the getPropertiesResponse message as specified in supportedFeatures
78

79 4.1 Overview of the CalWS protocol

80 CalWS operations and data elements are defined in this specification. Many of the operations result in the transmission of data as defined in [RFC 5545].
81

82 SOAP 1.1 messages consist of three elements: an envelope, header data, and a message body. CalWS request-response elements **MUST** be enclosed within the SOAP message body. CalWS SOAP messages **MUST** conform to [WT-I-Basic] and [WS-I-Bind]. A single CalWS SOAP message **MUST** contain only one service request or a single
83 service response).
84

85 The basic process for using SOAP for CalWS operations is:
86

87 A system entity acting as a CalWS requester transmits a CalWS request element within the body of a SOAP message to a system entity acting as a CalWS responder. The CalWS requester **MUST NOT** include more than one CalWS request per SOAP message or include any additional XML elements in the SOAP body (though see Section 4.1.1 for multiple messages packaged in one request).
88

89 The CalWS responder **MUST** return either a CalWS response element within the body of another SOAP message or generate a SOAP fault. The CalWS responder **MUST NOT** include more than one CalWS response per SOAP message or include any additional XML elements in the SOAP body. If a CalWS responder cannot, for some reason, process a CalWS request, it **MUST** generate a SOAP fault. (SOAP 1.1 faults and fault codes are discussed in [SOAP11] section 5.1.)
90
91
92
93
94
95

96 4.1.1 Discovery

97 CalWS implementers (service providers) **MUST** provide a WSDL **WSDL11** to describe their implementations. This WSDL **MAY** or may not be made public via a standard discovery mechanism (such as UDDI) or other method.
98

99 In addition, it is **REQUIRED** that the CalWS implementation include the Properties operation to provide dynamic information regarding CalWS capabilities, options, etc. that are supported.
100

101 4.1.2 Properties

102 A service or resource will have a number of properties which describe the current state of that service or resource. These properties are accessed through the execution of a properties operation specifying the target resource. See Retrieving Collection and Service Properties below
103
104

105 4.1.3 Operations

106 The following operations are defined by this specification:

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

116 4.1.4 Calendar Object Resources

117 The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC 4791] section 4.2. An additional constraint for CalWS is that no timezone specifications are transferred with the data.
118

119 4.1.5 Timezone information

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

125 4.1.6 Error conditions

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

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

132 Each method specification defines the preconditions that must be satisfied before the method can succeed. A number
133 of postconditions are generally specified which define the state that must exist after the execution of the operation.

134 Preconditions and postconditions are defined as error elements in the CalWS-SOAP XML namespace,
135 "http://docs.oasis-open.org/ws-calendar/ns/soap".

136 4.1.6.1 Example: error with error condition

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

145 4.2 CalWs-SOAP Messages.

146 This section describes the common elements and structure of CalWs-SOAP messages. The conventions followed are
147 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		

148 Table 1: Field column descriptions

149 4.2.1 Common Elements and types

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

152 All requests must include an href which specifies the target for the request. There is also an id attribute which will
153 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.

154 *Table 2: BaseRequestType elements*

155 A response may include an error response element of type ErrorResponse. This element will be returned in
 156 response messages when some form of processing error occurs and provides further information on the error beyond
 157 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

158 *Table 3: ErrorResponse elements*

159 4.2.1.1 **ErrorCodeType**

160 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: 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

Field	Type	Description
		many attendees. The maximum number is set by the target system and can be determined from the properties.
partialSuccess	PartialSuccessType	Indicates that a MultiOpType operation was partially successful. Returned when the operation is marked as non-atomic and one or more sub-operations failed. The entire response needs to be examined to determine failing operations.
missingChangeToken	MissingChangeTokenType	An operation was attempted which required a change token but none was supplied. Note that it appears that the marshalling or demarshalling should handle this as the token is required. It doesn't.
mismatchedChangeToken	MismatchedChangeTokenType	An update operation was attempted with a change token value which does not match that held by the service. The client must refetch the entity to refresh its cached value and token. Note that matching of tokens is a server 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.

161 Table 4: ErrorCodeType definitions

162 4.2.1.2 BaseResponseType

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

163 Table 5: BaseResponseType elements

164 4.3 Properties

165 The getPropertiesResponse message contains 0 or more properties defined below. Some properties apply to the service as a whole while others apply only to the targeted resource. The targeted resource may have property values
166 which override those for the service. For example, the timezone identifier for a particular collection may differ from
167 the default timezone identifier for the system.
168

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

170 4.3.1 childCollection

171 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

172 Table 6: ChildCollectionType fields

173 See resourceType for descriptions of CollectionType and Calendar CollectionType.

174 **4.3.2 creationDateTime**

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

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

176 *Table 7: CreationDateTimeType fields*

177 **4.3.3 displayName**

178 This property SHOULD be returned for any targeted resource.

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

179 *Table 8: DisplayNameType fields*

180 **4.3.4 lastModifiedDateTime**

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

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

182 *Table 9: LastModifiedDateTimeType fields*

183 **4.3.5 maxAttendeesPerInstance**

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

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

185 *Table 10: MaxAttendeesPerInstanceType fields*

186 **4.3.6 maxDateTime**

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

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

188 *Table 11: MaxDateTimeType fields*

189 **4.3.7 maxInstances**

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

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

191 *Table 12: MaxInstancesType fields*

192 **4.3.8 maxResourceSize**

193 This property SHOULD be returned for the service and MAY be returned for any targeted collection 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

Field	Type	#	?	Description
				accept when a calendar object resource is stored in a calendar collection.

194 Table 13: MaxResourceSizeType fields

195 4.3.9 minDateTime

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

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

197 Table 14: MinDateTimeType fields

198 4.3.10 principalHome

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

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

200 Table 15: PrincipalHomeType fields

201 4.3.11 resourceDescription

202 Provides some descriptive text for the targeted collection.

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

203 Table 16: ResourceDescriptionType fields

204 4.3.12 resourceOwner

205 This property SHOULD be returned for any targeted resource.

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

206 Table 17: ResourceownerType fields

207 4.3.13 resourceTimezoneId

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

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

209 Table 18: ResourceTimezoneIdType fields

210 4.3.14 resourceType

211 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

Field	Type	#	?	Description
inbox	InboxType	0..1	C	If present this is a scheduling inbox
xresource	XresourceType	0..1	C	If present provides further type information.

212 *Table 19: ResourceTypeType fields*

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

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

214 *Table 20: XresourceType fields*

215 **4.3.15 supportedCalendarComponentSet**

216 This property identifies which component types the service is prepared to store. The allowable components may be
217 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.

218 *Table 21: SupportedCalendarComponentSetType fields*

219 **4.3.16 supportedFeatures**

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

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

222 *Table 22: SupportedFeaturesType fields*

223 **4.3.17 timezoneServer**

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

Field	Type	#	?	Description
string	string	1	Y	The location of a timezone service used to retrieve timezone information and specifications. This may be an absolute URL referencing some other service or a relative URL if the current server also provides a timezone service.

225 *Table 23: TimezoneServerType fields*

226 **4.3.18 CalWS:privilege-set XML element**

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

228 Appears within a link relation describing collections or entities and specifies the set of privileges allowed to the cur-
229 rent authenticated principal for that collection or entity.

230 `<!ELEMENT calws:privilege-set (calws:privilege*)>`

231 `<!ELEMENT calws:privilege ANY>`

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

233 • CalWS: Read - current principal has read access

234 • CalWS: Write - current principal has write access

235 `<calWS:privilege-set>`

236 `<calWS:privilege><calWS:read></calWS:privilege>`

237 `<calWS:privilege><calWS:write></calWS:privilege>`

238 `</calWS:privilege-set>`

239 4.4 Retrieving Collection and Service Properties

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

242 The service properties define the global limits and defaults. Any properties defined on collections within the service
 243 hierarchy override those service defaults. The service may choose to prevent such overriding of defaults and limits
 244 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.

245 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

246 Table 25: GetPropertiesResponse fields

247 4.4.1 Example - retrieving server properties:

```

248 >>Request
249
250 <?xml version="1.0" encoding="UTF-8"?>
251 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
252   <SOAP-ENV:Header />
253   <SOAP-ENV:Body>
254     <ns2:getProperties xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
255       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
256       <ns2:href/></ns2:href>
257     </ns2:getProperties>
258   </SOAP-ENV:Body>
259 </SOAP-ENV:Envelope>
260
261 >>Response
262
263 <?xml version="1.0" encoding="UTF-8"?>
264 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
265   <SOAP-ENV:Header />
266   <SOAP-ENV:Body>
267     <ns2:getPropertiesResponse
268       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
269       xmlns:ns4="urn:ietf:params:xml:ns:icalendar-2.0"
270       id="0" >
271     <ns2:href/></ns2:href>
272     <ns2:lastModifiedDate>
273       <ns2:dateTime>2012-01-04T18:21:14Z</ns2:dateTime>
274     </ns2:lastModifiedDate>
275     <ns2:supportedCalendarComponentSet>
276       <ns4:vevent />
277       <ns4:vtodo />
278       <ns4:vavailability />
279     </ns2:supportedCalendarComponentSet>
280     <ns2:resourceType>
281       <ns2:collection />
282     </ns2:resourceType>
283     <ns2:supportedFeatures>
284       <ns2:calendarAccessFeature />
285     </ns2:supportedFeatures>
286     <ns2:maxInstances>
287       <ns2:integer>1000</ns2:integer>
288     </ns2:maxInstances>
289     <ns2:maxResourceSize>
290       <ns2:integer>100000</ns2:integer>
291     </ns2:maxResourceSize>
292   </ns2:getPropertiesResponse>
293 </SOAP-ENV:Body>

```

294
295
296

</SOAP-ENV:Envelope>

297 4.5 Creating Calendar Object Resources

298 Creating calendar object resources is carried out by using a CalWS-SOAP addItem request targeted at the parent
299 collection and containing the resource to be created. The response will contain the href of the newly created object.
300 The icalendar entity in the request MUST contain only a single calendaring entity with any related overrides.

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

301 Table 26: AddItem Type fields

302 The service will respond with an AddItemResponseType giving either the href and change token of the new entity
303 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

304 Table 27: AddItemResponseType additional fields

305 4.5.1 Preconditions for Calendar Object Creation

- 306 • **CalWS:target-exists:** The entity already exists.
- 307 • **CalWS:not-calendar-data:** The resource submitted MUST be a supported media type (i.e., iCalendar) for calen-
308 dar object resources;
- 309 • **CalWS:invalid-calendar-data:** The resource submitted MUST be valid data for the media type being specified
310 (i.e., MUST contain valid iCalendar data);
- 311 • **CalWS:invalid-calendar-object-resource:** The resource submitted in the request MUST obey all restrictions
312 specified in Calendar Object Resources (e.g., calendar object resources MUST NOT contain more than one type
313 of calendar component, calendar object resources MUST NOT specify the iCalendar METHOD property, etc.);
- 314 • **CalWS:unsupported-calendar-component:** The resource submitted in the request MUST contain a type of cal-
315 endar component that is supported in the targeted calendar collection;
- 316 • **CalWS:uid-conflict:** The resource submitted in the request MUST NOT specify an iCalendar UID property
317 value already in use in the targeted calendar collection or overwrite an existing calendar object resource with one
318 that has a different UID property value. Servers SHOULD report the URL of the resource that is already making
319 use of the same UID property value in the CalWS:href element
320 <!ELEMENT uid-conflict (CalWS:href)>
- 321 • **CalWS:exceeds-max-resource-size:** The resource submitted in the request MUST have an octet size less than or
322 equal to the value of the CalDAV:max-resource-size property value on the calendar collection where the resource
323 will be stored;
- 324 • **CalWS:before-min-date-time:** The resource submitted in the request MUST have all of its iCalendar DATE or
325 DATE-TIME property values (for each recurring instance) greater than or equal to the value of the CalDAV:min-
326 date-time property value on the calendar collection where the resource will be stored;
- 327 • **CalWS:after-max-date-time:** The resource submitted in the request MUST have all of its iCalendar DATE or
328 DATE-TIME property values (for each recurring instance) less than the value of the CalDAV:max-date-time
329 property value on the calendar collection where the resource will be stored;
- 330 • **CalWS:too-many-instances:** The resource submitted in the request MUST generate a number of recurring in-
331 stances less than or equal to the value of the CalDAV: max-instances property value on the calendar collection
332 where the resource will be stored;
- 333 • **CalWS:too-many-attendees-per-instance:** The resource submitted in the request MUST have a number of AT-
334 TENDEE properties on any one instance less than or equal to the value of the CalDAV:max-attendees-per-
335 instance property value on the calendar collection where the resource will be stored;

4.5.2 Example - successful addItem:

```

336 >>Request
337
338
339 <?xml version="1.0" encoding="UTF-8"?>
340 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
341   <SOAP-ENV:Header/>
342   <SOAP-ENV:Body>
343     <ns2:addItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
344       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
345       <ns2:href>/user/douglm/calendar</ns2:href>
346       <ns3:icalendar>
347         <ns3:vcalendar>
348           <ns3:components>
349             <ns3:vevent>
350               <ns3:properties>
351                 <ns3:uid>
352                   <ns3:text>1302064354993</ns3:text>
353                 </ns3:uid>
354                 <ns3:summary>
355                   <ns3:text>try this</ns3:text>
356                 </ns3:summary>
357                 <ns3:dtstart>
358                   <ns3:date-time>20110406T150000Z</ns3:date-time>
359                 </ns3:dtstart>
360                 <ns3:dtend>
361                   <ns3:date-time>20110406T160000Z</ns3:date-time>
362                 </ns3:dtend>
363               </ns3:properties>
364             </ns3:vevent>
365           </ns3:components>
366         </ns3:vcalendar>
367       </ns3:icalendar>
368     </ns2:addItem>
369   </SOAP-ENV:Body>
370 </SOAP-ENV:Envelope>
371
372 >>Response
373
374 <?xml version="1.0" encoding="UTF-8"?>
375 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
376   <SOAP-ENV:Header/>
377   <SOAP-ENV:Body>
378     <ns2:addItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
379       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
380       <ns2:status>OK</ns2:status>
381       <ns2:href>/user/douglm/calendar/1302064354993.ics</ns2:href>
382       <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
383     </ns2:addItemResponse>
384   </SOAP-ENV:Body>
385 </SOAP-ENV:Envelope>

```

4.6 Retrieving resources

386 Fetching calendar object resources is carried out by using a CalWS-SOAP fetchItem request with an href specifying
 387 the entity to be fetched. The response will contain the calendaring entity with any related overrides.
 388

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

389 *Table 28: FetchItem Type fields*

390 The service will respond with a FetchItemResponseType containing either the change token, its href and the entity
 391 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

4.6.1 Example - successful fetchItem:

```

393 >>Request
394
395
396 <?xml version="1.0" encoding="UTF-8"?>
397 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
398   <SOAP-ENV:Header/>
399   <SOAP-ENV:Body>
400     <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
401       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
402       <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
403     </ns2:fetchItem>
404   </SOAP-ENV:Body>
405 </SOAP-ENV:Envelope>
406
407 >>Response
408
409 <?xml version="1.0" encoding="UTF-8"?>
410 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
411   <SOAP-ENV:Header/>
412   <SOAP-ENV:Body>
413     <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
414       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
415       <ns2:status>OK</ns2:status>
416       <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
417       <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
418       <ns3:icalendar>
419         <ns3:vcalendar>
420           <ns3:properties>
421             <ns3:prodid>
422               <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
423             </ns3:prodid>
424             <ns3:version>
425               <ns3:text>2.0</ns3:text>
426             </ns3:version>
427           </ns3:properties>
428           <ns3:components>
429             <ns3:vevent>
430               <ns3:properties>
431                 <ns3:created>
432                   <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
433                 </ns3:created>
434                 <ns3:dtend>
435                   <ns3:date-time>20110406T160000Z</ns3:date-time>
436                 </ns3:dtend>
437                 <ns3:dtstamp>
438                   <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
439                 </ns3:dtstamp>
440                 <ns3:dtstart>
441                   <ns3:date-time>20110406T150000Z</ns3:date-time>
442                 </ns3:dtstart>
443                 <ns3:last-modified>
444                   <ns3:utc-date-time>20110406T155741Z</ns3:utc-date-time>
445                 </ns3:last-modified>
446                 <ns3:summary>
447                   <ns3:text>try this</ns3:text>
448                 </ns3:summary>
449                 <ns3:uid>
450                   <ns3:text>1302105461170</ns3:text>
451                 </ns3:uid>
452               </ns3:properties>
453             </ns3:vevent>
454           </ns3:components>
455         </ns3:vcalendar>
456       </ns3:icalendar>
457     </ns2:fetchItemResponse>
458   </SOAP-ENV:Body>
459 </SOAP-ENV:Envelope>

```

4.6.2 Example - unsuccessful fetchItem:

```
460 >>Request
461
462
463 <?xml version="1.0" encoding="UTF-8"?>
464 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
465   <SOAP-ENV:Header/>
466   <SOAP-ENV:Body>
467     <ns2:fetchItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
468       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
469       <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
470     </ns2:fetchItem>
471   </SOAP-ENV:Body>
472 </SOAP-ENV:Envelope>
473
474 >>Response
475
476 <?xml version="1.0" encoding="UTF-8"?>
477 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
478   <SOAP-ENV:Header/>
479   <SOAP-ENV:Body>
480     <ns2:fetchItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
481       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
482       <ns2:status>Error</ns2:status>
483       <ns2:errorResponse>
484         <ns2:targetDoesNotExist/>
485       </ns2:errorResponse>
486     </ns2:fetchItemResponse>
487   </SOAP-ENV:Body>
488 </SOAP-ENV:Envelope>
```

4.7 Updating resources

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

- 491 • An iCalendar element contains...
- 492 • a single vCalendar element which contains...
- 493 • one or more calendaring components, event, task etc each of which contain...
- 494 • zero or more components, alarms etc or one or more properties each of which contains...
- 495 • zero or more parameters and one or more values.

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

```
497   <icalendar>
498     <vcalendar>
499       <components>
500         <vevent>
501           <properties>
502             <uid>
503               <text>1302064354993-a</text>
504             </uid>
505             <summary>
506               <text>try this</text>
507             </summary>
508             <dtstart>
509               <date-time>2011-07-18T15:00:00Z</date-time>
510             </dtstart>
511             <dtend>
512               <date-time>2011-07-18T16:00:00Z</date-time>
513             </dtend>
514           </properties>
515         </vevent>
516       </components>
517     </vcalendar>
518   </icalendar>
```

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

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

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

- 527 • Remove - components, properties or parameters
- 528 • Add - components, properties or parameters
- 529 • Change - property or parameter values

530 Removals MUST be processed ahead of additions

531 Preconditions as specified in Preconditions for Calendar Object Creation are applicable. The response will indicate
 532 success or failure of the update. If the change token value does not match that held by the service a mismatched-
 533 ChangeToken error status will be returned. The client should re-fetch the entity to refresh its cache and then retry the
 534 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

535 *Table 30: UpdateItem fields*

536 The ComponentSelectionType contains three repeating child elements. The first allows for selection of nested
 537 components which can then be updated. The next allows addition of entire components and the last allows for the
 538 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

539 *Table 31: ComponentSelectionType fields*

540 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

541 *Table 32: PropertiesSelectionType fields*

542 To complete that pattern there is also a ParametersSelectionType used to select property parameters for update or
 543 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

544 *Table 33: ParametersSelectionType fields*

545 Each of these refers to a reference type. These either provide a complete entity for addition or identify the entity for
 546 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.

547 *Table 34: ComponentReferenceType fields*

Field	Type	#	?	Description
Any valid iCalendar	xcal:BasePropertyType	1	Y	Either a complete property or sufficient to identify it

Field	Type	#	?	Description
property name				or provide a new value, depending on usage.

548 *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.

549 *Table 36: ParameterReferenceType fields*

550 To complete the picture we have three selection types for component, property and parameter. Each of these identifies the entity to be updated, possible selections of the sub-elements and a possible change to values.
551
552 ComponentSelectionType contains three child elements. The first is any valid icalendar component element which
553 is to be matched at the current level.
554 The optional properties selection allows selection and possible updates to the properties of the component. An iCalendar
555 properties element cannot take a value so the only updates possible are addition and removal of properties.
556 Nested properties may be selected for updates.
557 The optional components selection allows selection and possible updates to the nested icalendar components element
558 of the component. An iCalendar components element cannot take a value so the only updates possible are addition
559 and removal of components. Nested components may be selected for 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

560 *Table 37: ComponentSelectionType fields*

561 PropertySelectionType contains three child elements. The first is any valid icalendar property element which is to
562 be matched at the current level.
563 The optional parameters selection allows selection and possible updates to the parameters of the property.
564 The optional change element allows a change to the value of the property. The new value is specified by supplying
565 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

566 *Table 38: PropertySelectionType fields*

567 Lastly, there is the ParameterSelectionType which contains two child elements. The first is any valid icalendar parameter
568 element which is to be matched at the current level.
569 The optional change element allows a change to the value of the parameter. The new value is specified by supplying
570 an iCalendar parameter with the desired value(s).

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

571 *Table 39: ParameterSelectionType fields*

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

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

574 Table 40: UpdateItemResponseType additional fields

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

576 4.7.1 Change tokens and concurrent updates

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

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

587 4.7.2 Example - successful update:

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

```
589 <ns3:icalendar>
590 <ns3:vcalendar>
591 <ns3:components>
592 <ns3:vevent>
593 <ns3:properties>
594 <ns3:uid>
595 <ns3:text>1302064354993-a</ns3:text>
596 </ns3:uid>
597 <ns3:summary>
598 <ns3:text>try this</ns3:text>
599 </ns3:summary>
600 <ns3:dtstart>
601 <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
602 </ns3:dtstart>
603 <ns3:dtend>
604 <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>
605 </ns3:dtend>
606 </ns3:properties>
607 </ns3:vevent>
608 </ns3:components>
609 </ns3:vcalendar>
610 </ns3:icalendar>
```

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

- 612 • Change the summary
- 613 • Change the dtstart - add a tzid and change the value to local time
- 614 • Add some categories

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

```
617 >>Request
618
619 <?xml version="1.0" encoding="UTF-8"?>
620 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
621 <SOAP-ENV:Header/>
622 <SOAP-ENV:Body>
623 <ns2:updateItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
624 <ns2:href>/user/douglm/calendar/1302064354993-a.ics</ns2:href>
625 <ns2:changeToken>"20110802T032608Z-0" null</ns2:changeToken>
626 <ns2:select>
627 <ns3:vcalendar/>
628 <ns2:components>
629 <ns2:component>
630 <ns3:vevent>
631 <ns3:properties>
632 <ns3:uid>
```

```

634         <ns3:text>1302064354993-a</ns3:text>
635     </ns3:uid>
636 </ns3:properties>
637 </ns3:vevent>
638 <ns2:properties>
639   <ns2:property>
640     <ns3:dtstart>
641       <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>
642     </ns3:dtstart>
643     <ns2:parameters>
644       <ns2:add>
645         <ns3:tzid>
646           <ns3:text>America/New_York</ns3:text>
647         </ns3:tzid>
648       </ns2:add>
649     </ns2:parameters>
650     <ns2:change>
651       <ns3:dtstart>
652         <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>
653       </ns3:dtstart>
654     </ns2:change>
655   </ns2:property>
656 </ns2:properties>
657 <ns3:summary>
658   <ns3:text>try this</ns3:text>
659 </ns3:summary>
660 <ns2:change>
661   <ns3:summary>
662     <ns3:text>A changed summary - again and again and again</ns3:text>
663   </ns3:summary>
664 </ns2:change>
665 </ns2:property>
666 <ns2:add>
667   <ns3:categories>
668     <ns3:text>newcategory-2</ns3:text>
669     <ns3:text>resources</ns3:text>
670     <ns3:text>paper</ns3:text>
671   </ns3:categories>
672 </ns2:add>
673 </ns2:properties>
674 </ns2:component>
675 </ns2:components>
676 </ns2:select>
677 </ns2:updateItem>
678 </SOAP-ENV:Body>
679 </SOAP-ENV:Envelope>
680
681 >>Response
682
683 <?xml version="1.0" encoding="UTF-8"?>
684 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
685   <SOAP-ENV:Header/>
686   <SOAP-ENV:Body>
687     <ns2:updateItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
688                           xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0"
689                           id="0">
690       <ns2:status>OK</ns2:status>
691     </ns2:updateItemResponse>
692   </SOAP-ENV:Body>
693 </SOAP-ENV:Envelope>

```

694 4.7.3 Other updates:

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

- 696 • Addition of properties
- 697 • Removal of properties
- 698 • Addition of parameters to properties
- 699 • Removal of parameters from properties
- 700 • Changing parameter values.

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

```
704     <ns2:properties>  
705       <ns2:property>  
706         <ns3:dtstart>  
707           <ns3:date-time>2011-07-18T15:00:00Z</ns3:date-time>  
708         </ns3:dtstart>  
709         <ns2:parameters>  
710           <ns2:add>  
711             <ns3:tzid>  
712               <ns3:text>America/New_York</ns3:text>  
713             </ns3:tzid>  
714           </ns2:add>  
715         </ns2:parameters>  
716         <ns2:change>  
717           <ns3:dtstart>  
718             <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>  
719           </ns3:dtstart>  
720         </ns2:change>  
721       </ns2:property>  
722     </ns2:properties>
```

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

```
724     <ns2:properties>  
725       <ns2:add>  
726         <ns3:categories>  
727           <ns3:text>paper</ns3:text>  
728         </ns3:categories>  
729       </ns2:add>  
730       <ns2:add>  
731         <ns3:categories>  
732           <ns3:text>resources</ns3:text>  
733         </ns3:categories>  
734       </ns2:add>  
735     </ns2:properties>
```

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

```
737     <ns2:properties>  
738       <ns2:remove>  
739         <ns3:dtend>  
740           <ns3:date-time>2011-07-18T16:00:00Z</ns3:date-time>  
741         </ns3:dtend>  
742       </ns2:remove>  
743       <ns2:add>  
744         <ns3:duration>  
745           <ns3:duration>PT1H</ns3:duration>  
746         </ns3:duration>  
747       </ns2:add>  
748     </ns2:properties>
```

749 In this example we change the dtstart timezone identifier.

```
750     <ns2:properties>  
751       <ns2:property>  
752         <ns3:dtstart>  
753           <ns3:parameters>  
754             <ns3:tzid>  
755               <ns3:text>America/New_York</ns3:text>  
756             </ns3:tzid>  
757           </ns3:parameters>  
758           <ns3:date-time>2011-07-18T11:00:00</ns3:date-time>  
759         </ns3:dtstart>  
760         <ns2:parameters>  
761           <ns2:parameter>  
762             <ns3:tzid>  
763               <ns3:text>America/New_York</ns3:text>  
764             </ns3:tzid>  
765           <ns2:change>  
766             <ns3:tzid>  
767               <ns3:text>America/Montreal</ns3:text>  
768             </ns3:tzid>  
769           </ns2:change>  
770         </ns2:parameter>  
771       </ns2:parameters>
```

772
773
774

```
</ns2:property>
</ns2:properties>
```

775 4.7.4 Creating an update message.

776 The update can be created in many ways but the most common approach is to build the update while modifications
777 take place or to create one as the result of comparing old and new versions. It appears that comparing XML for dif-
778 ferences is difficult. However, we can take advantage of the structure of calendaring entities to simplify the process.
779 There are implementations available which take the diff approach to producing an update stream.
780 There are some special cases to consider when comparing. Some properties are multi-valued and may themselves
781 appear more than once. There is no semantic information implied by any grouping though parameters may need to
782 be taken into account. These properties need to be normalized before comparison and when updating them we pro-
783 duce a change which treats each value as a single property.

784 These properties are

- 785 • categories
- 786 • exdate
- 787 • freebusy
- 788 • rdate

789 This normalization can take place before comparison.

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

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

795 4.8 Deletion of resources

796 Deletion of calendar object resources is carried out by using a CalWS-SOAP deleteItem request with an href specify-
797 ing the entity to be deleted. The deleteItem request is not valid when the href specifies a collection.

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

798 *Table 41: DeleteItem fields*

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

801 4.8.1 Example - successful deleteItem:

```

802 >>Request
803
804 <?xml version="1.0" encoding="UTF-8"?>
805 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
806   <SOAP-ENV:Header/>
807   <SOAP-ENV:Body>
808     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
809       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
810       <ns2:href>/user/douglm/calendar/1302620814655.ics</ns2:href>
811     </ns2:deleteItem>
812   </SOAP-ENV:Body>
813 </SOAP-ENV:Envelope>
814
815 >>Response
816
817 <?xml version="1.0" encoding="UTF-8"?>
818 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
819   <SOAP-ENV:Header/>
820   <SOAP-ENV:Body>
821     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
822       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
823       <ns2:status>OK</ns2:status>
824     </ns2:deleteItemResponse>
825   </SOAP-ENV:Body>
826 </SOAP-ENV:Envelope>
```

827 4.8.2 Example - unsuccessful deleteItem:

```
828 >>Request
829
830 <?xml version="1.0" encoding="UTF-8"?>
831 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
832   <SOAP-ENV:Header/>
833   <SOAP-ENV:Body>
834     <ns2:deleteItem xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
835       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
836       <ns2:href>/user/douglm/calendar/nosuchevent.ics</ns2:href>
837     </ns2:deleteItem>
838   </SOAP-ENV:Body>
839 </SOAP-ENV:Envelope>
840
841 >>Response
842
843 <?xml version="1.0" encoding="UTF-8"?>
844 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
845   <SOAP-ENV:Header/>
846   <SOAP-ENV:Body>
847     <ns2:deleteItemResponse xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
848       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
849       <ns2:status>Error</ns2:status>
850       <ns2:errorResponse>
851         <ns2:targetDoesNotExist/>
852       </ns2:errorResponse>
853     </ns2:deleteItemResponse>
854   </SOAP-ENV:Body>
855 </SOAP-ENV:Envelope>
```

856 4.9 Querying calendar resources

857 Querying provides a mechanism by which information can be obtained from the service through possibly complex
858 queries. A skeleton icalendar entity can be provided to limit the amount of information returned to the client. A que-
859 ry takes the parts

- 860 • Limitations on the data returned
- 861 • Selection of the data
- 862 • Optional timezone id for floating time calculations.

863 4.9.1 Calendar Query common types

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

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

866 *Table 42: UTCTimeRangeType elements*

867 The TextMatchType is used to match text values in properties and parameters. The collation attribute species a col-
868 lation as defined in [RFC4790].

869 Servers are REQUIRED to support the "i;ascii-casemap" and "i;octet" collations which provide a basic case insensi-
870 tive and case sensitive match respectively.

871 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

872 *Table 43: TextMatchType attributes*

873 **4.9.2 CompFilterType**

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

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

879 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

880 *Table 44: CompFilterType elements*

881 **4.9.3 PropFilterType**

882 The prop-filter element may test for the absence of a property or match values or specify zero or more ParamFilter-
883 Type 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

884 *Table 45: PropFilterType elements*

885 **4.9.4 ParamFilterType**

886 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

Field	Type	#	?	Description
text-match	TextMatchtype	0..1	N	

887 Table 46: ParamFilterType elements

888 4.9.5 CalendarQueryType elements

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

889 Table 47: CalendarQueryType elements

890 4.9.6 Specifying data to be returned

891 This is achieved by specifying one of the following

- 892 • allprop: return all properties and calendar data. (some properties are specified as not being part of the allprop set so are not returned)
- 893
- 894 • Set the icalendar element. This is an icalendar valueless pattern entity which provides a map of the components and properties to be returned. Neither the pattern nor the returned result need to be valid icalendar entities in that
- 895 required properties may be absent if unselected.
- 896

897 4.9.7 Pre/postconditions for calendar queries

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

900 4.9.8 Time range limited queries.

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

903 4.9.9 Example: time range limited retrieval

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

```
906 >> Request <<
907
908 <?xml version="1.0" encoding="UTF-8"?>
909 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
910   <SOAP-ENV:Header/>
911   <SOAP-ENV:Body>
912     <ns2:calendarQuery xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
913       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
914       <ns2:href>/user/douglm/calendar</ns2:href>
915       <ns3:icalendar>
916         <ns3:vcalendar>
917           <ns3:components>
918             <ns3:vevent>
919               <ns3:properties>
920                 <ns3:summary/>
921                 <ns3:dtstart/>
922                 <ns3:dtend/>
923                 <ns3:duration/>
924                 <ns3:uid/>
925                 <ns3:recurrence-id/>
926                 <ns3:rrule/>
927                 <ns3:rdate/>
928                 <ns3:exdate/>
929               </ns3:properties>
930             </ns3:vevent>
931           </ns3:components>
932         </ns3:vcalendar>
933       </ns3:icalendar>
934       <ns2:filter>
935         <ns2:compFilter test="anyof">
936           <ns3:vcalendar />
937           <ns2:compFilter>
938             <ns3:vevent />
939             <ns2:time-range end="20110430T040000Z" start="20110401T040000Z"/>
940           </ns2:compFilter>
941         </ns2:filter>
942       </ns2:calendarQuery>
943     </SOAP-ENV:Body>
944   </SOAP-ENV:Envelope>
945
946 >> Response <<
947
948 <?xml version="1.0" encoding="UTF-8"?>
949 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
950   <SOAP-ENV:Header/>
951   <SOAP-ENV:Body>
952     <ns2:calendarQueryResponse
953       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
954       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
955       <ns2:status>OK</ns2:status>
956       <ns2:response>
957         <ns2:href>/user/douglm/calendar/1302105461170.ics</ns2:href>
958         <ns2:changeToken>"20110406T155741Z-0"</ns2:changeToken>
959         <ns2:propstat>
960           <ns2:prop>
961             <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
962               <ns3:icalendar>
963                 <ns3:vcalendar>
964                   <ns3:properties>
965                     <ns3:prodid>
966                       <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
967                     </ns3:prodid>
968                     <ns3:version>
```



```

969         <ns3:text>2.0</ns3:text>
970     </ns3:version>
971 </ns3:properties>
972 <ns3:components>
973     <ns3:vevent>
974         <ns3:properties>
975             <ns3:dtend>
976                 <ns3:date-time>20110406T160000Z</ns3:date-time>
977             </ns3:dtend>
978             <ns3:dtstart>
979                 <ns3:date-time>20110406T150000Z</ns3:date-time>
980             </ns3:dtstart>
981             <ns3:summary>
982                 <ns3:text>try this</ns3:text>
983             </ns3:summary>
984             <ns3:uid>
985                 <ns3:text>1302105461170</ns3:text>
986             </ns3:uid>
987         </ns3:properties>
988     </ns3:vevent>
989 </ns3:components>
990 </ns3:vcalendar>
991 </ns3:icalendar>
992 </ns2:calendar-data>
993 </ns2:prop>
994     <ns2:status>OK</ns2:status>
995 </ns2:propstat>
996 </ns2:response>
997 <ns2:response>
998     <ns2:href>/user/douglm/calendar/CAL-00f1fc61-2f021bca-012f-022947f8-
999 00000006.ics</ns2:href>
1000     <ns2:changeToken>"20110405T140920Z-0"</ns2:changeToken>
1001     <ns2:propstat>
1002         <ns2:prop>
1003             <ns2:calendar-data content-type="application/xml+calendar" version="2.0">
1004                 <ns3:icalendar>
1005                     <ns3:vcalendar>
1006                         <ns3:properties>
1007                             <ns3:prodid>
1008                                 <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
1009                             </ns3:prodid>
1010                             <ns3:version>
1011                                 <ns3:text>2.0</ns3:text>
1012                             </ns3:version>
1013                         </ns3:properties>
1014                         <ns3:components>
1015                             <ns3:vevent>
1016                                 <ns3:properties>
1017                                     <ns3:duration>
1018                                         <ns3:duration>PT1H</ns3:duration>
1019                                     </ns3:duration>
1020                                     <ns3:dtstart>
1021                                         <ns3:parameters>
1022                                             <ns3:tzid>
1023                                                 <ns3:text>America/New_York</ns3:text>
1024                                             </ns3:tzid>
1025                                         </ns3:parameters>
1026                                         <ns3:date-time>20110412T110000</ns3:date-time>
1027                                     </ns3:dtstart>
1028                                     <ns3:summary>
1029                                         <ns3:text>Test recurring event</ns3:text>
1030                                     </ns3:summary>
1031                                     <ns3:uid>
1032                                         <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1033 00000006demobedework@mysite.edu</ns3:text>
1034                                     </ns3:uid>
1035                                     <ns3:rrule>
1036                                         <ns3:recur>
1037                                             <ns3:freq>WEEKLY</ns3:freq>
1038                                             <ns3:count>2</ns3:count>
1039                                             <ns3:interval>1</ns3:interval>
1040                                         </ns3:recur>
1041                                     </ns3:rrule>

```

```

1042         </ns3:properties>
1043     </ns3:vevent>
1044     <ns3:vevent>
1045         <ns3:properties>
1046             <ns3:recurrence-id>
1047                 <ns3:parameters>
1048                     <ns3:tzid>
1049                         <ns3:text>America/New York</ns3:text>
1050                     </ns3:tzid>
1051                 </ns3:parameters>
1052                 <ns3:date-time>20110419T150000Z</ns3:date-time>
1053             </ns3:recurrence-id>
1054             <ns3:duration>
1055                 <ns3:duration>PT1H</ns3:duration>
1056             </ns3:duration>
1057             <ns3:dtstart>
1058                 <ns3:parameters>
1059                     <ns3:tzid>
1060                         <ns3:text>America/New_York</ns3:text>
1061                     </ns3:tzid>
1062                 </ns3:parameters>
1063                 <ns3:date-time>20110419T120000</ns3:date-time>
1064             </ns3:dtstart>
1065             <ns3:summary>
1066                 <ns3:text>Test recurring event</ns3:text>
1067             </ns3:summary>
1068             <ns3:uid>
1069                 <ns3:text>CAL-00f1fc61-2f021bca-012f-022947f8-
1070 00000006demobedework@mysite.edu</ns3:text>
1071             </ns3:uid>
1072         </ns3:properties>
1073     </ns3:vevent>
1074 </ns3:components>
1075 </ns3:vcalendar>
1076 </ns3:icalendar>
1077 </ns2:calendar-data>
1078 </ns2:prop>
1079 <ns2:status>OK</ns2:status>
1080 </ns2:propstat>
1081 </ns2:response>
1082 </ns2:calendarQueryResponse>
1083 </SOAP-ENV:Body>
1084 </SOAP-ENV:Envelope>
1085

```

4.10 Free-busy queries

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

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

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

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

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

4.10.1 Element values

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

1104 4.10.1.1 start

1105 **Default:** If omitted the default value is left up to the server. It may be the current day, start of the current month,
1106 etc.

1107 **Description:** Specifies the start date for the Freebusy data. The server is free to ignore this value and return data
1108 in any time range. The client must check the data for the returned time range.

1109 **Format:** An XML UTC date-time

1110 **Example:**

```
1111 2011-12-01T10:15:00Z
```

1112 **Notes:** Specifying only a start date/time without specifying an end-date/time or period should be interpreted as
1113 in [RFC 5545]. The effective period should cover the remainder of that day.

1114 4.10.1.2 end

1115 **Default:** Same as start

1116 **Description:** Specifies the end date for the Freebusy data. The server is free to ignore this value.

1117 **Format:** Same as start

1118 **Example:** Same as start

1119 The server is free to ignore the start, end and period parameters. It is recommended that the server return at least 6
1120 weeks of data from the current day.

1121 A client **MUST** check the time range in the response as a server may return a different time range than the requested
1122 range.

1123 4.10.2 Examples

1124 The following is an unsuccessful request targeting an invalid resource.

```
1125 >> Request <<
1126
1127 <?xml version="1.0" encoding="UTF-8"?>
1128 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1129   <SOAP-ENV:Header/>
1130   <SOAP-ENV:Body>
1131     <ns2:freebusyReport
1132       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1133       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1134       <ns2:href>/user/douglm/calendar</ns2:href>
1135       <ns2:time-range>
1136         <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1137         <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1138       </ns2:time-range>
1139     </ns2:freebusyReport>
1140   </SOAP-ENV:Body>
1141 </SOAP-ENV:Envelope>
1142
1143 >> Response <<
1144
1145 <?xml version="1.0" encoding="UTF-8"?>
1146 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1147   <SOAP-ENV:Header/>
1148   <SOAP-ENV:Body>
1149     <ns2:freebusyReportResponse
1150       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1151       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1152       <ns2:status>Error</ns2:status>
1153       <ns2:message>Only principal href supported</ns2:message>
1154     </ns2:freebusyReportResponse>
1155   </SOAP-ENV:Body>
1156 </SOAP-ENV:Envelope>
```

1157 The following is an example of a request to retrieve Freebusy data for a user:

```
1158 >> Request <<
1159
1160 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1161   <SOAP-ENV:Header/>
1162   <SOAP-ENV:Body>
1163     <ns2:freebusyReport
1164       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1165       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1166       <ns2:href>/principals/users/douglm</ns2:href>
1167       <ns2:time-range>
```

```

1168     <ns2:start>2011-04-01T04:00:00Z</ns2:start>
1169     <ns2:end>2011-04-30T04:00:00Z</ns2:end>
1170     </ns2:time-range>
1171   </ns2:freebusyReport>
1172 </SOAP-ENV:Body>
1173 </SOAP-ENV:Envelope>
1174
1175 >> Response <<
1176
1177 <?xml version="1.0" encoding="UTF-8"?>
1178 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
1179   <SOAP-ENV:Header/>
1180   <SOAP-ENV:Body>
1181     <ns2:freebusyReportResponse
1182       xmlns:ns2="http://docs.oasis-open.org/ws-calendar/ns/soap"
1183       xmlns:ns3="urn:ietf:params:xml:ns:icalendar-2.0">
1184       <ns2:status>OK</ns2:status>
1185       <ns3:icalendar>
1186         <ns3:vcalendar>
1187           <ns3:properties>
1188             <ns3:prodid>
1189               <ns3:text>//Bedework.org//BedeWork V3.7//EN</ns3:text>
1190             </ns3:prodid>
1191             <ns3:version>
1192               <ns3:text>2.0</ns3:text>
1193             </ns3:version>
1194           </ns3:properties>
1195           <ns3:components>
1196             <ns3:vfreebusy>
1197               <ns3:properties>
1198                 <ns3:attendee>
1199                   <ns3:parameters>
1200                     <ns3:partstat>
1201                       <ns3:text>NEEDS-ACTION</ns3:text>
1202                     </ns3:partstat>
1203                   </ns3:parameters>
1204                   <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1205                 </ns3:attendee>
1206               <ns3:created>
1207                 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1208               </ns3:created>
1209               <ns3:dtend>
1210                 <ns3:date-time>2011-04-30T00:00:00Z</ns3:date-time>
1211               </ns3:dtend>
1212               <ns3:dtstamp>
1213                 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1214               </ns3:dtstamp>
1215               <ns3:dtstart>
1216                 <ns3:date-time>2011-04-01T00:00:00Z</ns3:date-time>
1217               </ns3:dtstart>
1218               <ns3:freebusy>
1219                 <ns3:parameters>
1220                   <ns3:fbtype>
1221                     <ns3:text>BUSY</ns3:text>
1222                   </ns3:fbtype>
1223                 </ns3:parameters>
1224                 <ns3:period>
1225                   <ns3:start>2011-04-06T15:00:00Z</ns3:start>
1226                   <ns3:end>2011-04-06T16:00:00Z</ns3:end>
1227                 </ns3:period>
1228               </ns3:freebusy>
1229               <ns3:last-modified>
1230                 <ns3:utc-date-time>2011-06-30T15:45:56Z</ns3:utc-date-time>
1231               </ns3:last-modified>
1232               <ns3:organizer>
1233                 <ns3:parameters/>
1234                 <ns3:cal-address>mailto:douglm@mysite.edu</ns3:cal-address>
1235               </ns3:organizer>
1236               <ns3:uid>
1237                 <ns3:text>2UTDVPZ9H0EQL9QISI44SP5IFPC4N75</ns3:text>
1238               </ns3:uid>
1239             </ns3:properties>
1240           </ns3:vfreebusy>

```

1241
1242
1243
1244
1245
1246
1247

```
</ns3:components>  
</ns3:vcalendar>  
</ns3:icalendar>  
</ns2:freebusyReportResponse>  
</SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

1248 4.11 Multiple operations

1249 Each of the previously described operations acts upon a single entity or resource only. Frequently we have the need
1250 to update an interconnected set of entities so that we maintain the consistency of the structure. This requires an
1251 atomic operation which can successfully update all the entities or roll back the operation on failure.

1252 The MultiOpType operation provides such a feature. It is essentially a wrapper around any of the other operations
1253 which guarantees the success of the entire set or a roll back. Using the id attribute for requests, each individual re-
1254 sponse can be located in the result.

1255 The MultiOpType request takes the following elements

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

1256 Table 48: MultiOpType elements

1257 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

1258 Table 49: MultiOpResponseType elements

1259
1260
1261
1262

1263

5 Conformance

1264 Certain calendaring properties and components are interrelated and it is necessary to have knowledge of all these
1265 properties and their current values to allow consistent update and understanding of a target component. The norma-
1266 tive definition for these relationships is RFC5445, RFC5446 and related RFCs.

1267 However, those specifications assume a complete view of entities being fetched or updated. This specification al-
1268 lows updates of entities when only a partial view is available. In fact it is the very nature of SOAP based transaction
1269 to provide such a partial view. Given that, parties attempting to update entities MUST have sufficient information to
1270 ensure the end result is consistent. Services allowing updates to entities MUST ensure that the result after an update
1271 operation is still internally consistent.

1272 5.1 Start, end and duration in calendar components

1273 A period of time is fully specified by a start and an end or duration.

1274 5.1.1 Updating, transporting and maintaining start, and and duration.

- 1275 • For all components the calculated or specified start must be at or before the end.
- 1276 • When a system updates or stores a calendar component it MUST retain the relationship of start, end and duration.
1277 Applications MUST NOT without good cause, change a start and end pair into a start and duration nor the re-
1278 verse. Semantically they are not equivalent when DST transitions occur during the time of the event.
- 1279 • For interoperability, iCalendar based systems SHOULD avoid the use of weekly durations and XML based sys-
1280 tems SHOULD avoid the use of yearly durations.

1281 5.1.2 VEVENT:

- 1282 • The three properties are DTSTART, DTEND and DURATION.
- 1283 • DTSTART MUST appear once and only one of DTEND or DURATION MAY be present.
- 1284 • The DTSTART property for a VEVENT specifies the inclusive start of the event. For recurring events, it also
1285 specifies the very first instance in the recurrence set.
- 1286 • The DTEND property for a VEVENT calendar component specifies the non-inclusive end of the event.
- 1287 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE value type but no
1288 DTEND nor DURATION property, the event's duration is taken to be one day.
- 1289 • For cases where a VEVENT calendar component specifies a DTSTART property with a DATE-TIME value type
1290 but no DTEND nor DURATION property, the event ends on the same calendar date and time of day specified by
1291 the DTSTART property, that is, it signifies a zero length instant in time.

1292 5.1.3 VTODO:

- 1293 • The three properties are DTSTART, DUE, DURATION.
- 1294 • DTSTART MAY appear once.
- 1295 • Either DUE or DURATION MAY appear in a VTODO, but DUE and DURATION MUST NOT occur in the
1296 same VTODO.
- 1297 • If DURATION does appear in a VTODO, then DTSTART MUST also appear in the same VTODO.
- 1298 • The three properties for a VTODO are related in the same way as for VEVENT. Additionally a VTODO calendar
1299 component without the DTSTART and DUE (or DURATION) properties specifies a VTODO that will be associ-
1300 ated with each successive calendar date, until it is completed.

1301 5.1.4 VJOURNAL:

- 1302 • DTSTART only, which may be a date or date-time value.

1303 5.1.5 VAVAILABILITY

- 1304 • DTSTART and DTEND if specified MUST be date-time values.
- 1305 • DTSTART MAY appear once and signifies start of the busy period.
- 1306 • Only one of DTEND or DURATION MAY appear and signify the end of the busy period.

- 1307 • If DURATION does appear in a VAVAILABILITY, then DTSTART MUST also appear in the same VAVAIL-
1308 ABILITY.

1309 **5.1.6 AVAILABILITY**

- 1310 • DTSTART and DTEND if specified MUST be date-time values.
1311 • DTSTART MUST appear once and signifies start of the free period.
1312 • Only one of DTEND or DURATION MAY appear and signify the end of the free period.

1313 **5.2 Recurrences.**

- 1314 • The RECURRENCE-ID is a property of each instance of a recurring event. It is calculated from the DTSTART
1315 and the recurrence rules or added to the set by the RDATE property.
1316 • RDATE, EXDATE and RECURRENCE-ID must take the same form as the DTSTART. That is if DTSTART is a
1317 DATE value then the RDATE and EXDATE must be DATE. If DTSTART is a date-time the RDATE and
1318 EXDATE values must take the same form, including the same timezone.
1319 • Overrides to an instance are specified by completely specifying the instance with the appropriate RECUR-
1320 RENCE-ID property.
1321 • An RDATE adds an instance to the recurrence set.
1322 • An EXDATE deletes an instance by specifying the recurrence id(s) to be deleted. Applications SHOULD NOT
1323 specify overrides for instances so deleted.
1324 • The recurrence set is calculated from the RRULE and RDATES and then applying any EXDATE properties. That
1325 is EXDATE takes precedence over RDATE and the RRULE.

1326 **5.3 Alarms:**

- 1327 • Alarms are typically anchored to the start or end of an event or task. This is defined by the RELATED parameter
1328 to the TRIGGER property.

1329 **5.4 Unrecognized or unsupported elements**

- 1330 • A system SHOULD reject any attempt to store components which it does not support. A SYSTEM MUST adver-
1331 tise which components are supported through the use of the supportedCalendarComponentSet property.
1332 • A system MUST ignore any elements it does not understand.

1333

Appendix A Acknowledgments

1334

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

1335

Participants:

1336

Bruce Bartell, Southern California Edison

1337

Brad Benson, Trane

1338

Edward Cazalet, Individual

1339

Toby Considine, University of North Carolina at Chapel Hill

1340

William Cox, Individual

1341

Sharon Dinges, Trane

1342

Mike, Douglass, Rensselaer Polytechnic Institute

1343

Craig Gemmill, Tridium, Inc.

1344

Girish Ghatikar, Lawrence Berkeley National Laboratory

1345

Gerald Gray, Southern California Edison

1346

David Hardin, ENERNOC

1347

Gale Horst, Electric Power Research Institute (EPRI)

1348

Gershon Janssen, Individual

1349

Ed Koch, Akuacom Inc.

1350

Benoit Lepeuple, LonMark International*

1351

Carl Mattocks, CheckMi*

1352

Robert Old, Siemens AG

1353

Alexander Papaspyrou, Technische Universitat Dortmund

1354

Joshua Phillips, ISO/RTO Council (IRC)

1355

Jeremy J. Roberts, LonMark International

1356

David Thewlis, CalConnect

1357

The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-Calendar

1358

Technical Committee, bridging to developing IETF standards and contributing the services definitions that make up

1359

Services in Section 4. The Technical Committee gratefully acknowledges their assistance and cooperation as well.

1360

Contributors to TC XML include:

1361

Cyrus Daboo, Apple

1362

Mike Douglass, Rensselaer Polytechnic Institute

1363

Steven Lees, Microsoft

1364

Tong Li, IBM

1365

Appendix A ~~Appendix B~~ Revision History

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

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

1367