

WS-Calendar Calendar Update and Synchronization with REST-based Services Version 1.0

Working Draft ~~0206~~

~~13 February 2012~~

29 January 2013

Technical Committee:

~~OASIS Web Services Calendar (WS-Calendar) TC~~

OASIS Web Services Calendar (WS-Calendar) TC

Chair:

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

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

Editor:

~~Michael Douglass (dougln@rpi.edu|dougln@rpi.edu) Rensselaer Polytechnic Institute~~
Rensselaer Polytechnic Institute

Related work:

This specification is related to:

- *WS-Calendar Version 1.0*. Latest version.
~~<http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.html>~~
<http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.html>

Abstract:

Summary of the technical purpose of the document.

Status:

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

Copyright © OASIS Open ~~2011~~2013. 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~~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.

Table of Contents

1	Introduction	4
1.1	Terminology	4
1.2	Normative References	4
1.3	Non-Normative References	4
2	Calendar Services	5
2.1	Overview of the protocol	5
2.1.1	Calendar Object Resources	5
2.1.2	Timezone information	5
2.1.3	Issues not addressed by this specification	6
2.1.3.1	Access Control	6
2.1.3.2	Provisioning	6
2.1.3.3	Copy/Move	6
2.1.3.4	Creating Collections	6
2.1.3.5	Retrieving collections	6
2.1.3.6	Setting service and resource properties	6
2.1.4	CalWS Glossary	6
2.1.4.1	Hrefs	6
2.1.4.2	Calendar Object Resource	6
2.1.4.3	Calendar Collection	6
2.1.4.4	Scheduling Calendar Collection	7
2.1.4.5	Principal Home	7
2.2	Error conditions	7
2.2.1	Example: error with CalDAV error condition	7
3	Properties and link relations	8
3.1	Property and relation-type URIs	8
3.2	supported-features property	8
3.3	max-attendees-per-instance	8
3.4	max-date-time	8
3.5	max-instances	8
3.6	max-resource-size	8
3.7	min-date-time	9
3.8	description	9
3.9	timezone-service relation	9
3.10	principal-home relation	9
3.11	current-principal-freebusy relation	9
3.12	principal-freebusy relation	9
3.13	child-collection relation	9
3.14	created link property	10
3.15	last-modified property	10
3.16	displayname property	10
3.17	timezone property	10
3.18	owner property	10
3.19	collection link property	10
3.20	calendar-collection link property	10

3.21 CalWS:privilege-set XML element.....	11
4 Retrieving Collection and Service Properties.....	12
4.1 Request parameters.....	12
4.2 Responses:.....	12
4.3 Example – retrieving server properties:.....	12
5 Creating Calendar Object Resources.....	13
5.1 Request parameters.....	13
5.2 Responses:.....	13
5.3 Preconditions for Calendar Object Creation.....	13
5.4 Example – successful POST:.....	14
5.5 Example – unsuccessful POST:.....	14
6 Retrieving resources.....	16
6.1 Request parameters.....	16
6.2 Responses:.....	16
6.3 Example – successful fetch:.....	16
6.4 Example – unsuccessful fetch:.....	16
7 Updating resources.....	17
7.1 Responses:.....	17
8 Deletion of resources.....	19
8.1 Delete for Collections.....	19
8.2 Responses:.....	19
9 Querying calendar resources.....	20
9.1 Limiting data returned.....	20
9.2 Pre/postconditions for calendar queries.....	20
9.3 Example: time range limited retrieval.....	20
10 Free busy queries.....	25
10.1 ACCEPT header.....	25
10.2 URL Query Parameters.....	25
10.2.1 start.....	25
10.2.2 end.....	26
10.2.3 period.....	26
10.2.4 account.....	26
10.3 URL parameters - notes.....	26
10.4 HTTP Operations.....	26
10.5 Response Codes.....	26
10.6 Examples.....	27
11 Conformance.....	30
Appendix A. Acknowledgments.....	31
Appendix B. Revision History.....	32

1 Introduction

[All text is normative unless otherwise labeled]

1	Introduction	3
2	Calendar Services	5
3	Properties and link relations	8
4	Retrieving Collection and Service Properties	12
5	Creating Calendar Object Resources	14
6	Retrieving resources	16
7	Updating resources	17
8	Deletion of resources	19
9	Querying calendar resources	20
10	Free-busy queries	25
11	Conformance	30
12	Appendix A. Acknowledgments	33
13	Appendix B. An Introduction to Internet Calendaring	34
14	Appendix C. Revision History	37

1 Introduction

The CalWS REST 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 REST services will be built without any CalDAV support.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt><http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [RFC4791] C. Daboo, B. Desruisseaux, L. Dusseault, *Calendaring Extensions to WebDAV (CalDAV)*, <http://www.ietf.org/rfc/rfc4791.txt><http://www.ietf.org/rfc/rfc4791.txt>, IETF RFC4791, March 1997.
- [WS-Calendar-1.0] *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>.<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf>.
- [XRD] *Extensible Resource Descriptor (XRD) Version 1.0*, 1 November 2010, OASIS Standard, <http://docs.oasis-open.org/xri/xrd/v1.0/os/xrd-1.0-os.xml>

1.3 Non-Normative References

- REST T Fielding, *Architectural Styles and the Design of Network-based Software Architectures*, <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm><http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

1.4 Namespace

XML namespaces and prefixes used in this standard:

Table 1 1: XML Namespaces in this standard

Prefix	Namespace
--------	-----------

<u>xcal</u>	<u>urn:ietf:params:xml:ns:icalendar-2.0</u>
<u>calWS</u>	<u>http://docs.oasis-open.org/ws-calendar/ns/REST</u>
<u>xrd</u>	<u>http://docs.oasis-open.org/ns/xri/xrd-1.0</u>

58

59

2 Calendar Services

61 The Service interactions are built upon and make the same assumptions about structure as the CalDAV
62 protocol defined in **[RFC4791]** and related specifications. It does NOT require nor assume the WebDAV
63 nor CalDAV protocol but does make use of some of the same elements and structures in the CalDAV
64 XML namespace.

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

67 These services can be looked upon as a layer built on top of CalDAV and defines the basic operations
68 which allow creation, retrieval, update and deletion. In addition, query, and free-busy operations are
69 defined to allow efficient, partial retrieval of calendar data.

70 These services assume a degree of conformity with CalDAV is established such that services built in that
71 manner do not have a significant mismatch. It is assumed that some WS-Calendar services will be built
72 without any CalDAV support.

2.1 Overview of the protocol

74 The protocol is an HTTP based RESTfull protocol using a limited set of methods. Each request may be
75 followed by a response containing status information.

76 The following methods are specified in the protocol description, PUT, POST, GET, DELETE. To avoid
77 various issues with certain methods being blocked clients may use the X-HTTP-Method-Override: header
78 to specify the intended operation. Servers SHOULD behave as if the named method was used.

```
79 POST /user/fred/calendar/ HTTP/1.1  
80 ...  
81 X-HTTP-Method-Override: PUT  
82 Properties
```

83 A service or resource will have a number of properties which describe the current state of that service or
84 resource. These properties are accessed through a GET on the target resource or service with an
85 ACCEPT header specifying application/xrd+xml. See Section 2.1.3.6

86 The following operations are defined by this specification:

- 87 • Retrieval and update of service and resource properties
- 88 • Creation of a calendar object
- 89 • Retrieval of a calendar object
- 90 • Update of a calendar object
- 91 • Deletion of a calendar object
- 92 • Query
- 93 • Free-busy query

2.1.1 Calendar Object Resources

95 The same restrictions apply to Calendar Object Resources as specified in CalDAV **[RFC4791]** section
96 4.2. An additional constraint for CalWS is that no timezone specifications are transferred.

2.1.2 Timezone information

98 It is assumed that the client and server each have access to a full set of up to date timezone information.
99 Timezones will be referenced by a timezone identifier from the full set of Olson data together with a set of
100 well-known aliases defined **[TZDB]**. CalWS services may advertise themselves as timezone servers
101 through the server properties object.

102 **2.1.3 Issues not addressed by this specification.**

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

105 **2.1.3.1 Access Control**

106 It is assumed that the targeted server will set an appropriate level of access based on authentication. This
107 specification will not attempt to address the issues of sharing or Access Control Lists (ACLs).

108 **2.1.3.2 Provisioning**

109 The protocol will not provide any explicit provisioning operations. If it is possible to authenticate or
110 address a principal's calendar resources then they **MUST** be automatically created if necessary or
111 appropriate

112 **2.1.3.3 Copy/Move**

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

117 **2.1.3.4 Creating Collections**

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

120 **2.1.3.5 Retrieving collections**

121 This operation is currently undefined. A GET on a collection may fail or return a complete calendar object
122 representing the collection.

123 **2.1.3.6 Setting service and resource properties.**

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

126 **2.1.4 CalWS Glossary**

127 **2.1.4.1 Hrefs**

128 An href is a URI reference to a resource, for example

129 `"http://example.org/user/fred/calendar/event1.ics".`

130 The URL above reflects a possible structure for a calendar server. All URLs should be absolute or path-
131 absolute following the rules defined in ~~Error! Reference source not found.~~ Section 8.3.3.1: Property and
132 relation-type URIs

133 **2.1.4.2 Calendar Object Resource**

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

136 **2.1.4.3 Calendar Collection**

137 A folder only allowed to contain calendar object resources.

138 2.1.4.4 Scheduling Calendar Collection

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

141 2.1.4.5 Principal Home

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

144 2.2 Error conditions

145 Each operation on the calendar system has a number of pre-conditions and post-conditions that apply.
146 A "precondition" for a method describes the state of the server that must be true for that method to be
147 performed. A "post-condition" of a method describes the state of the server that must be true after that
148 method has been completed. Any violation of these conditions will result in an error response in the form
149 of a CalWS XML error element containing the violated condition and an optional description. \
150 Each method specification defines the preconditions that must be satisfied before the method can
151 succeed. A number of post-conditions are generally specified which define the state that must exist after
152 the execution of the operation. Preconditions and post-conditions are defined as error elements in the
153 CalWS XML namespace.

154 2.2.1 Example: error with CalDAV error condition

```
155 <?xml version="1.0" encoding="utf-8"  
156     xmlns:CW="Error! Reference source not found."  
157     xmlns:C="urn:ietf:params:xml:http://docs.oasis-open.org/ws-  
158 calendar/ns+caldav/REST" ?>  
159 <CW:error>  
160   <C:supported-filter>  
161     <C:prop-filter name="X-ABC-GUID"/>  
162   </C:supported-filter>  
163   <CW:description>Unknown property </CW:description>  
164 </CW:error>
```

165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203

3 Properties and link relations

3.1 Property and relation-type URIs

In the XRD entity returned properties and related services and entities are defined by absolute URIs which correspond to the extended relation type defined in **[web linking]** Section 4.2. These URIs do NOT correspond to any real entity on the server and clients should not attempt to retrieve any data at that target.

Certain of these property URIs correspond to CalDAV preconditions. Each URL is prefixed by the CalWS relations and properties namespace <http://docs.oasis-open.org/ns/wscal/calws>. <http://docs.oasis-open.org/ws-calendar/ns/REST/>. Those properties which correspond to CalDAV properties have the additional path element "caldav/", for example

```
http://docs.oasis-open.org/ns/wscal/calws/caldavhttp://docs.oasis-open.org/ws-calendar/ns/REST/supported-calendar-data
```

corresponds to

```
CalDAV:supported-calendar-data
```

In addition to those CalDAV properties, the CalWS specification defines a number of other properties and link relations with the URI prefix of <http://docs.oasis-open.org/ns/wscal/calws>. <http://docs.oasis-open.org/ws-calendar/ns/REST/>.

3.2 supported-features property.

```
http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-calendar/ns/REST/supported-features
```

This property defines the features supported by the target. All resources contained and managed by the service should return this property. The value is a comma separated list containing one or more of the following

- calendar-access - the service supports all MUST requirements in this specification

```
<Property type="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-calendar/ns/REST/supported-features">  
  >calendar-access</Property>
```

3.3 max-attendees-per-instance

```
http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-calendar/ns/REST/max-attendees-per-instance
```

Defines the maximum number of attendees allowed per event or task.

3.4 max-date-time

```
http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-calendar/ns/REST/max-date-time
```

Defines the maximum date/time allowed on an event or task

3.5 max-instances

```
http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-calendar/ns/REST/max-instances
```

Defines the maximum number of instances allowed per event or task

204 3.6 max-resource-size

205 | <http://docs.oasis-open.org/ns/wscal/calws/> <http://docs.oasis-open.org/ws-calendar/ns/REST/>max-
206 resource-size

207 Provides a numeric value indicating the maximum size of a resource in octets that the server is willing to
208 accept when a calendar object resource is stored in a calendar collection.

209 3.7 min-date-time

210 | <http://docs.oasis-open.org/ns/wscal/calws/> <http://docs.oasis-open.org/ws-calendar/ns/REST/>min-
211 date-time

212 Provides a DATE-TIME value indicating the earliest date and time (in UTC) that the server is willing to
213 accept for any DATE or DATE-TIME value in a calendar object resource stored in a calendar collection.

214 3.8 description

215 | <http://docs.oasis-open.org/ns/wscal/calws/> [calendar/ns/REST/](http://docs.oasis-open.org/ws-
216 calendar/ns/REST/)description

217 Provides some descriptive text for the targeted collection.

218 3.9 timezone-service relation.

219 | <http://docs.oasis-open.org/ns/wscal/calws/> [calendar/ns/REST/](http://docs.oasis-open.org/ws-
220 calendar/ns/REST/)timezone-service

221 The location of a timezone service used to retrieve timezone information and specifications. This may be
222 an absolute URL referencing some other service or a relative URL if the current server also provides a
223 timezone service.

```
224 | <Link rel="http://docs.oasis-open.org/ns/wscal/calws/calwshttp://docs.oasis-open.org/ws-  
225 | calendar/ns/REST/timezone-service"  
226 | href="http://example.com/tz" />
```

227 3.10 principal-home relation.

228 | <http://docs.oasis-open.org/ns/wscal/calws/> [calendar/ns/REST/](http://docs.oasis-open.org/ws-
229 | calendar/ns/REST/)principal-home

230 Provides the URL to the user home for the currently authenticated principal.

```
231 | <Link rel="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-  
232 | calendar/ns/REST/principal-home"  
233 | href="http://example.com/user/fred" />
```

234 3.11 current-principal-freebusy relation.

235 | <http://docs.oasis-open.org/ns/wscal/calws/> [calendar/ns/REST/](http://docs.oasis-open.org/ws-
236 | calendar/ns/REST/)current-principal-freebusy

237 Provides the URL to use as a target for freebusy requests for the current authenticated principal.

```
238 | <Link rel="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-  
239 | calendar/ns/REST/current-principal-freebusy"  
240 | href="http://example.com/freebusy/user/fred" />
```

241 3.12 principal-freebusy relation.

242 | <http://docs.oasis-open.org/ns/wscal/calws/> [calendar/ns/REST/](http://docs.oasis-open.org/ws-
243 | calendar/ns/REST/)principal-freebusy

244 Provides the URL to use as a target for freebusy requests for a different principal.

```
245 | <Link rel="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-
246 | calendar/ns/REST/principal-freebusy"
247 | href="http://example.com/freebusy" />
```

248 3.13 child-collection relation.

```
249 | http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-calendar/ns/REST/child-
250 | collection
```

251 Provides information about a child collections for the target. The href attribute gives the URI of the
252 collection. The element should only have CalWS child elements giving the type of the collection, that is
253 the **CalWS**:collection link property and the CalWS-calendar-collection link property. This allows
254 clients to determine the structure of a hierarchical system by targeting each of the child collections in turn.

255 The xrd:title child element of the link element provides a description for the child-collection.

```
256 | <Link rel="_ http://http://docs.oasis-open.org/ns/wscal/calws://docs.oasis-
257 | open.org/ws-calendar/ns/REST/child-collection"
258 | href="http://example.com/ealwscalWS/user/fred/calendar">
259 | <Title xml:lang="en">Calendar</Title>
260 | <Property type="http://docs.oasis-open.org/ns/wscal/calws/
261 | http://docs.oasis-open.org/ws-calendar/ns/REST/collection"
262 | xsi:nil="true" />
263 | <Property type="http://docs.oasis-open.org/ns/wscal/calws/
264 | http://docs.oasis-open.org/ws-calendar/ns/REST/calendar-collection"
265 | xsi:nil="true" />
266 | </Link>
```

267 3.14 created link property

```
268 | http://docs.oasis-open.org/ns/wscal/calws/created
```

```
269 | http://docs.oasis-open.org/ws-calendar/ns/REST/created
```

270 Appears within a link relation describing collections or entities. The value is a date-time as defined in
271 **Error! Reference source not found.[WS-Calendar]** Section 5.6

```
272 | <Property type="http://docs.oasis-open.org/ns/wscal/calws/created"
273 | http://docs.oasis-open.org/ws-calendar/ns/REST/created"
274 | >1985-04-12T23:20:50.52Z</Property>
```

275 3.15 last-modified property

```
276 | http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-calendar/ns/REST/last-
277 | modified
```

278 Appears within an xrd object describing collections or entities. The value is the same format as would
279 appear in the Last-Modified header and is defined in **[RFC2616]**, Section 3.3.1

```
280 | <Property type="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-
281 | open.org/ws-calendar/ns/REST/last-modified"
282 | >Mon, 12 Jan 1998 09:25:56 GMT</Property>
```

283 3.16 displayname property

```
284 | http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-
285 | calendar/ns/REST/displayname
```

286 Appears within an xrd object describing collections or entities. The value is a localized name for the entity
287 or collection.

```
288 | <Property type="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-
289 | open.org/ws-calendar/ns/REST/displayname"
290 | >My Calendar</Property>
```

291 3.17 timezone property

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

293 <http://docs.oasis-open.org/ws-calendar/ns/REST/timezone>

294 Appears within an xrd object describing collections. The value is a text timezone identifier.

```
295 <Property type="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-  
296 open.org/ws-calendar/ns/REST/timezone"  
297 >America/New_York</Property>
```

298 3.18 owner property

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

300 <http://docs.oasis-open.org/ws-calendar/ns/REST/owner>

301 Appears within an xrd object describing collections or entities. The value is a server specific uri.

```
302 <Property type="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-  
303 open.org/ws-calendar/ns/REST/owner"  
304 >/principals/users/mike</Property>
```

305 3.19 collection link property

306 [http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-
calendar/ns/REST/collection](http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-
307 calendar/ns/REST/collection)

308 Appears within a link relation describing collections or entities. The property takes no value and indicates
309 that this child element is a collection.

```
310 <Property type="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-  
311 open.org/ws-calendar/ns/REST/collection"  
312 xsi:nil="true" />
```

313 3.20 calendar-collection link property

314 [http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-
calendar/ns/REST/calendar-collection](http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-open.org/ws-
315 calendar/ns/REST/calendar-collection)

316 Appears within a link relation describing collections or entities. The property takes no value and indicates
317 that this child element is a calendar collection.

```
318 <Property type="http://docs.oasis-open.org/ns/wscal/calws/ http://docs.oasis-  
319 open.org/ws-calendar/ns/REST/calendar-collection"  
320 xsi:nil="true" />
```

321 3.21 CalWScaIWS:privilege-set XML element

322 [http://docs.oasis-open.org/ns/wscal/calws: http://docs.oasis-open.org/ws-
calendar/ns/REST/privilege-set](http://docs.oasis-open.org/ns/wscal/calws: http://docs.oasis-open.org/ws-
323 calendar/ns/REST/privilege-set)

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

```
326 <!ELEMENT ealwscalWS:privilege-set (ealwscalWS:privilege*)>  
327 <!ELEMENT ealwscalWS:privilege ANY>
```

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

- 329 • **CalWScaIWS**: Read - current principal has read access
- 330 • **CalWScaIWS**: Write - current principal has write access

```
331 <calWS:privilege-set>  
332 <calWS:privilege><calWS:read></calWS:privilege>  
333 <calWS:privilege><calWS:write></calWS:privilege>  
334 </calWS:privilege-set>
```

4 Retrieving Collection and Service Properties

336 Properties, related services and locations are obtained from the service or from service resources in the
337 form of an XRD document as defined by [XRD-1.0].

338 Given the URL of a CalWS service a client retrieves the service XRD document through a GET on the
339 service URL with an ACCEPT header specifying application/xrd+xml.

340 Retrieving resource properties is identical to obtaining service properties, that is, execute a GET on the
341 target URL with an ACCEPT header specifying application/xrd+xml.

342 The service properties define the global limits and defaults. Any properties defined on collections within
343 the service hierarchy override those service defaults. The service may choose to prevent such overriding
344 of defaults and limits when appropriate.

4.1 Request parameters

- None

4.2 Responses:

- 200: OK
- 403: Forbidden
- 404: Not found

4.3 Example - retrieving server properties:

```
>>Request
GET / HTTP/1.1
Host: example.com
ACCEPT:application/xrd+xml

>>Response
<XRD xmlns="http://docs.oasis-open.org/ns/xri/xrd-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Expires>1970-01-01T00:00:00Z</Expires>
  <Subject>http://example.com/ealwscalWS</Subject>
  <Property type="http://docs.oasis-open.org/ns/wscal/calws/="
http://docs.oasis-open.org/ws-calendar/ns/REST/created"
    >1970-01-01</Property>

  <Link rel="http://docs.oasis-open.org/ns/wscal/calws/=" http://docs.oasis-open.org/ws-calendar/ns/REST/timezone-service"
    href="http://example.com/tz" />

  <calWS:privilege-set>
  <calWS:privilege><calWS:read></calWS:privilege>
  </calWS:privilege-set>

  <Link rel="http://docs.oasis-open.org/ns/wscal/calws/=" http://docs.oasis-open.org/ws-calendar/ns/REST/principal-home"
    type="collection"
    href="http://example.com/ealwscalWS/user/fred">
  <Title xml:lang="en">Fred's calendar home</Title>
  </Link>

  <Link rel="http://docs.oasis-open.org/ns/wscal/calws/=" http://docs.oasis-open.org/ws-calendar/ns/REST/child-collection"
    type="calendar,scheduling"
```

```
385 |         href="http://example.com/ealwscalWS/user/fred/calendar">
386 |     <Title xml:lang="en">Calendar</Title>
387 |     </Link>
388 |
389 |     <Property type="http://docs.oasis-open.org/ns/wscal/calws/=""
390 | http://docs.oasis-open.org/ws-calendar/ns/REST/max-instances"
391 |         >1000</Property>
392 |
393 |     <Property type="http://docs.oasis-open.org/ns/wscal/calws/=""
394 | http://docs.oasis-open.org/ws-calendar/ns/REST/max-attendees-per-instance"
395 |         >100</Property>
396 |
397 | </XRD>
398 |
```


399

5 Creating Calendar Object Resources

400 Creating calendar object resources is carried out by a POST on the parent collection. The body of the
401 request will contain the resource being created. The request parameter "action=create" indicates this
402 POST is a create. The location header of the response gives the URL of the newly created object.

5.1 Request parameters

- action=create

5.2 Responses:

- 201: created
- 403: Forbidden - no access

5.3 Preconditions for Calendar Object Creation

- **CalWScaIWS:target-exists:** The target of a PUT must exist. Use POST to create entities and PUT to update them.
- **CalWScaIWS:not-calendar-data:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST be a supported media type (i.e., iCalendar) for calendar object resources;
- **CalWScaIWS:invalid-calendar-data:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST be valid data for the media type being specified (i.e., MUST contain valid iCalendar data);
- **CalWScaIWS:invalid-calendar-object-resource:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST obey all restrictions specified in Calendar Object Resources (e.g., calendar object resources MUST NOT contain more than one type of calendar component, calendar object resources MUST NOT specify the iCalendar METHOD property, etc.);
- **CalWScaIWS:unsupported-calendar-component:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST contain a type of calendar component that is supported in the targeted calendar collection;
- **CalWScaIWS:uid-conflict:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST NOT specify an iCalendar UID property value already in use in the targeted calendar collection or overwrite an existing calendar object resource with one that has a different UID property value. Servers SHOULD report the URL of the resource that is already making use of the same UID property value in the **CalWScaIWS:href** element
<!ELEMENT uid-conflict (**CalWScaIWS:href**)>
- **CalWScaIWS:invalid-calendar-collection-location:** In a COPY or MOVE request, when the Request-URI is a calendar collection, the Destination-URI MUST identify a location where a calendar collection can be created;
- **CalWScaIWS:exceeds-max-resource-size:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have an octet size less than or equal to the value of the CalDAV:max-resource-size property value on the calendar collection where the resource will be stored;
- **CalWScaIWS:before-min-date-time:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have all of its iCalendar DATE or DATE-TIME property values (for each recurring instance) greater than or equal to the value of the CalDAV:min-date-time property value on the calendar collection where the resource will be stored;
- **CalWScaIWS:after-max-date-time:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST have all of its iCalendar DATE or DATE-TIME property values (for each recurring instance) less than the value of the CalDAV:max-date-time property value on the calendar collection where the resource will be stored;

- 445 • **CalWS:too-many-instances:** The resource submitted in the PUT request, or targeted by a
446 COPY or MOVE request, MUST generate a number of recurring instances less than or equal to the
447 value of the CalDAV: max-instances property value on the calendar collection where the resource
448 will be stored;
- 449 • **CalWS:too-many-attendees-per-instance:** The resource submitted in the PUT request, or
450 targeted by a COPY or MOVE request, MUST have a number of ATTENDEE properties on any one
451 instance less than or equal to the value of the CalDAV:max-attendees-per-instance property value
452 on the calendar collection where the resource will be stored;

453 5.4 Example - successful POST:

```
454 >>Request
455
456 POST /user/fred/calendar/?action=create HTTP/1.1
457 Host: example.com
458 Content-Type: application/xml+calendar; charset="utf-8"
459 Content-Length: ?
460
461 <?xml version="1.0" encoding="utf-8" ?>
462 <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
463   <vcalendar>
464     ...
465   </vcalendar>
466 </icalendar>
467
468 >>Response
469
470 HTTP/1.1 201 Created
471 Location: http://example.com/user/fred/calendar/event1.ics
```

472 5.5 Example - unsuccessful POST:

```
473 >>Request
474
475 POST /user/fred/readcalendar/?action=create HTTP/1.1
476 Host: example.com
477 Content-Type: text/text; charset="utf-8"
478 Content-Length: ?
479
480 This is not an xml calendar object
481
482 >>Response
483
484 HTTP/1.1 403 Forbidden
485 <?xml version="1.0" encoding="utf-8"
486   xmlns:D="DAV:"
487   xmlns:C="urn:ietf:params:xml:ns:caldav" ?>
488 <D:error>
489   <C:supported-calendar-data/>
490   <D:description>Not an icalendar object</D:description>
491 </D:error>
```

492 6 Retrieving resources

493 A simple GET on the href will return a named resource. If that resource is a recurring event or task with
494 overrides, the entire set will be returned. The desired format is specified in the ACCEPT header. The
495 default form is application/xml+calendar

496 6.1 Request parameters

- 497 • none

498 6.2 Responses:

- 499 • 200: OK
- 500 • 403: Forbidden - no access
- 501 • 406 The requested format specified in the accept header is not supported.

502 6.3 Example - successful fetch:

```
503 >>Request
504 GET /user/fred/calendar/event1.ics HTTP/1.1
505 Host: example.com
506
507 >>Response
508
509 HTTP/1.1 200 OK
510 Content-Type: application/xml+calendar; charset="utf-8"
511 Content-Length: ?
512
513 <?xml version="1.0" encoding="utf-8" ?>
514 <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
515   <vcalendar>
516     ...
517   </vcalendar>
518 </icalendar>
```

520 6.4 Example - unsuccessful fetch:

```
521 >>Request
522 PUT /user/fred/calendar/noevent1.ics HTTP/1.1
523 Host: example.com
524
525 >>Response
526
527 HTTP/1.1 404 Not found
```

529

7 Updating resources

530 Resources are updated with the PUT method targeted at the resource href. The body of the request
531 contains a complete new resource which effectively replaces the targeted resource. To allow for optimistic
532 locking of the resource use the if-match header.

533 When updating a recurring event all overrides and master must be supplied as part of the content.

534 Preconditions as specified in Section 5.3 are applicable.

535 7.1 Responses:

- 536 • 200: OK
- 537 • 304: Not modified - entity was modified by some other request
- 538 • 403: Forbidden - no access, does not exist etc. See error response

539

540 *Example 7-41: Successful Update*

```
541 >>Request
542
543 PUT /user/fred/calendar/event1.ics HTTP/1.1
544 Host: example.com
545 Content-Type: application/xml+calendar; charset="utf-8"
546 Content-Length: ?
547
548 <?xml version="1.0" encoding="utf-8" ?>
549 <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
550   <vcalendar>
551     ...
552   </vcalendar>
553 </icalendar>
554
555 >>Response
556
557 HTTP/1.1 200 OK
```

558 *Example 7-22: Unsuccessful Update*

```
559 >>Request
560
561 PUT /user/fred/readcalendar/event1.ics HTTP/1.1
562 Host: example.com
563 Content-Type: application/xml+calendar; charset="utf-8"
564 Content-Length: ?
565
566 <?xml version="1.0" encoding="utf-8" ?>
567 <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
568   <vcalendar>
569     ...
570   </vcalendar>
571 </icalendar>
572
573 >>Response
574
575 HTTP/1.1 403 Forbidden
576 Content-Type: application/xml; charset="utf-8"
577 Content-Length: xxxx
578
579 <?xml version="1.0" encoding="utf-8"
580   xmlns:D="DAV:"
581   xmlns:CW=" http://docs.oasis-open.org/ws-calendar/CalWS-ns/REST/" ?>
```

```
582 <CW:error>  
583   <CW:target-exists/>  
584   <CW:description>Target of update must exist</C:description>  
585 </CW:error>
```

586 8 Deletion of resources

587 Delete is defined in [RFC 2616] Section 9.7. In addition to conditions defined in that specification, servers
588 must remove any references from the deleted resource to other resources. Resources are deleted with
589 the DELETE method targeted at the resource URL. After a successful completion of a deletion a GET on
590 that URL must result in a 404 - Not Found status.

591 8.1 Delete for Collections

592 Delete for collections may or may not be supported by the server. Certain collections are considered
593 undeletable. On a successful deletion of a collection all contained resources to any depth must also be
594 deleted.

595 8.2 Responses:

- 596 • 200: OK
- 597 • 403: Forbidden - no access
- 598 • 404: Not Found

602 9 Querying calendar resources

603 Querying provides a mechanism by which information can be obtained from the service through possibly
604 complex queries. A list of icalendar properties can be specified to limit the amount of information returned
605 to the client. A query takes the parts

- 603 • Limitations on the data returned
- 604 • Selection of the data
- 605 • Optional timezone id for floating time calculations.

606 The current specification uses CalDAV multiget and calendar-query XML bodies as specified in **[RFC**
607 **4791]** with certain limitations and differences.

- 608 1. The POST method is used for all requests, the action being identified by the outer element.
- 609 2. While CalDAV servers generally only support [RFC 5545] and assume that as the default, the
610 delivery format for CalWS will, by default, be [draft-xcal].
- 611 3. The CalDAV query allows the specification of a number of DAV properties. Specification of these
612 properties, with the exception of DAV:getetag, is considered an error in CalWS.
- 613 4. The CalDAV:propnames element is invalid

614 With those differences, the CalDAV specification is the normative reference for this operation.

615 9.1 Limiting data returned

616 This is achieved by specifying one of the following

- 617 • CalDAV:allprop return all properties (some properties are specified as not being part of the allprop
618 set so are not returned)
- 619 • CalDAV:prop An element which contains a list of properties to be returned . May only contain
620 DAV:getetag and CalDAV:calendar-data

621 Of particular interest, and complexity, is the calendar-data property which can contain a time range to limit
622 the range of recurrences returned and/or a list of calendar properties to return.

623 9.2 Pre/postconditions for calendar queries

624 The preconditions as defined in in **[RFC 4791]** Section 7.8 apply here. CalDav errors may be reported by
625 the service when preconditions or postconditions are violated.

626 9.3 Example: time range limited retrieval

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

```
629 >> Request <<
630
631 POST /user/fred/calendar/ HTTP/1.1
632 Host: ealws-calWS.example.com
633 Depth: 1
634 Content-Type: application/xml; charset="utf-8"
635 Content-Length: xxxx
636
637 <?xml version="1.0" encoding="utf-8" ?>
638 <C:calendar-query xmlns:D="DAV:"
639     xmlns:C="urn:ietf:params:xml:ns:caldav">
640   <D:prop>
641     <D:getetag/>
642     <C:calendar-data content-type="application/xml+calendar" >
643       <C:comp name="VCALENDAR">
```

```

644     <C:prop name="VERSION"/>
645     <C:comp name="VEVENT">
646         <C:prop name="SUMMARY"/>
647         <C:prop name="UID"/>
648         <C:prop name="DTSTART"/>
649         <C:prop name="DTEND"/>
650         <C:prop name="DURATION"/>
651         <C:prop name="RRULE"/>
652         <C:prop name="RDATE"/>
653         <C:prop name="EXRULE"/>
654         <C:prop name="EXDATE"/>
655         <C:prop name="RECURRENCE-ID"/>
656     </C:comp>
657 </C:comp>
658 </C:calendar-data>
659 </D:prop>
660 <C:filter>
661     <C:comp-filter name="VCALENDAR">
662         <C:comp-filter name="VEVENT">
663             <C:time-range start="20060104T000000Z"
664                 end="20060105T000000Z"/>
665         </C:comp-filter>
666     </C:comp-filter>
667 </C:filter>
668 </C:calendar-query>
669
670 >> Response <<
671
672 HTTP/1.1 207 Multi-Status
673 Date: Sat, 11 Nov 2006 09:32:12 GMT
674 Content-Type: application/xml; charset="utf-8"
675 Content-Length: xxxx
676
677 <?xml version="1.0" encoding="utf-8" ?>
678 <D:multistatus xmlns:D="DAV:"
679     xmlns:C="urn:ietf:params:xml:ns:caldav">
680     <D:response>
681         <D:href>http://cal.example.com/bernard/work/abcd2.ics</D:href>
682         <D:propstat>
683             <D:prop>
684                 <D:getetag>"fffff-abcd2"</D:getetag>
685                 <C:calendar-data content-type="application/xml+calendar" >
686                     <xc:icalendar
687                         xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
688                         <xc:vcalendar>
689                             <xc:properties>
690                                 <xc:calscale><text>GREGORIAN</text></xc:calscale>
691                                 <xc:prodid>
692                                     <xc:text>-//Example Inc.//Example Calendar//EN</xc:text>
693                                 </xc:prodid>
694                                 <xc:version><xc:text>2.0</xc:text></xc:version>
695                             </xc:properties>
696                             <xc:components>
697                                 <xc:vevent>
698                                     <xc:properties>
699                                         <xc:dtstart>
700                                             <xc:parameters>
701                                                 <xc:tzid>US/Eastern<xc:tzid>
702                                             <xc:parameters>
703                                                 <xc:date-time>20060102T120000</xc:date-time>
704                                         </xc:dtstart>
705                                         <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
706                                         <xc:summary>
707                                             <xc:text>Event #2</xc:text>

```



```

708     </xc:summary>
709     <xc:uid>
710       <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
711     </xc:uid>
712     <xc:rrule>
713       <xc:recur>
714         <xc:freq>DAILY</xc:freq>
715         <xc:count>5</xc:count>
716       </xc:recur>
717     </xc:rrule>
718   </xc:properties>
719 </xc:vevent>
720
721 <xc:vevent>
722   <xc:properties>
723     <xc:dtstart>
724       <xc:parameters>
725         <xc:tzid>US/Eastern<xc:tzid>
726       <xc:parameters>
727         <xc:date-time>20060104T140000</xc:date-time>
728     </xc:dtstart>
729     <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
730     <xc:summary>
731       <xc:text>Event #2 bis</xc:text>
732     </xc:summary>
733     <xc:uid>
734       <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
735     </xc:uid>
736     <xc:recurrence-id>
737       <xc:parameters>
738         <xc:tzid>US/Eastern<xc:tzid>
739       <xc:parameters>
740         <xc:date-time>20060104T120000</xc:date-time>
741     </xc:recurrence-id>
742     <xc:rrule>
743       <xc:recur>
744         <xc:freq>DAILY</xc:freq>
745         <xc:count>5</xc:count>
746       </xc:recur>
747     </xc:rrule>
748   </xc:properties>
749 </xc:vevent>
750
751 <xc:vevent>
752   <xc:properties>
753     <xc:dtstart>
754       <xc:parameters>
755         <xc:tzid>US/Eastern<xc:tzid>
756       <xc:parameters>
757         <xc:date-time>20060106T140000</xc:date-time>
758     </xc:dtstart>
759     <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
760     <xc:summary>
761       <xc:text>Event #2 bis bis</xc:text>
762     </xc:summary>
763     <xc:uid>
764       <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
765     </xc:uid>
766     <xc:recurrence-id>
767       <xc:parameters>
768         <xc:tzid>US/Eastern<xc:tzid>
769       <xc:parameters>
770         <xc:date-time>20060106T120000</xc:date-time>
771     </xc:recurrence-id>

```

```

772     <xc:rrule>
773         <xc:recur>
774             <xc:freq>DAILY</xc:freq>
775             <xc:count>5</xc:count>
776         </xc:recur>
777     </xc:rrule>
778 </xc:properties>
779 </xc:vevent>
780 </xc:components>
781 </xc:vcalendar>
782 </xc:icalendar>
783     </C:calendar-data>
784 </D:prop>
785     <D:status>HTTP/1.1 200 OK</D:status>
786 </D:propstat>
787 </D:response>
788 <D:response>
789     <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
790     <D:propstat>
791         <D:prop>
792             <D:getetag>"fffff-abcd3"</D:getetag>
793             <C:calendar-data content-type="application/xml+calendar" >
794                 <xcal:icalendar
795                     xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
796 <xc:vcalendar>
797     <xc:properties>
798         <xc:calscale><text>GREGORIAN</text></xc:calscale>
799         <xc:prodid>
800             <xc:text>-//Example Inc.//Example Calendar//EN</xc:text>
801         </xc:prodid>
802         <xc:version><xc:text>2.0</xc:text></xc:version>
803     </xc:properties>
804 <xc:components>
805     <xc:vevent>
806         <xc:properties>
807             <xc:dtstart>
808                 <xc:parameters>
809                     <xc:tzid>US/Eastern<xc:tzid>
810                 <xc:parameters>
811                     <xc:date-time>20060104T100000</xc:date-time>
812             </xc:dtstart>
813             <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
814             <xc:summary>
815                 <xc:text>Event #3</xc:text>
816             </xc:summary>
817             <xc:uid>
818                 <xc:text>DC6C50A017428C5216A2F1CD@example.com</xc:text>
819             </xc:uid>
820         <xc:rrule>
821             <xc:recur>
822                 <xc:freq>DAILY</xc:freq>
823                 <xc:count>5</xc:count>
824             </xc:recur>
825         </xc:rrule>
826     </xc:properties>
827 </xc:vevent>
828 </xc:components>
829 </xc:vcalendar>
830 </xc:icalendar>
831     </C:calendar-data>
832 </D:prop>
833     <D:status>HTTP/1.1 200 OK</D:status>
834 </D:propstat>
835 </D:response>

```

837 10 Free-busy queries

838 Free-busy queries are used to obtain free-busy information for a calendar-collection or principals. The
839 result contains information only for events to which the current principal has sufficient access.

840 When targeted at a calendar collection the result is based only on the calendaring entities contained in
841 that collection. When targeted at a principal free-busy URL the result will be based on all information
842 which affect the principals free-busy status, for example availability.

843 The possible targets are:

- 844 • A calendar collection URL
- 845 • The XRD link with relation CalWS/current-principal-freebusy
- 846 • The XRD link with relation CalWS/principal-freebusy with a principal given in the request.

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

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

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

856 10.1 ACCEPT header

857 The Accept header is used to specify the format for the returned data. In the absence of a header the
858 data should be returned as specified in **[draft-xcal]**, that is, as if the following had been specified

```
859 ACCEPT: application/xml+calendar
```

860 10.2 URL Query Parameters

861 None of these parameters are required except for the conditions noted below. Appropriate defaults will be
862 supplied by the server.

863 10.2.1 start

864 **Default:** The default value is left up to the server. It may be the current day, start of the current
865 month, etc.

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

868 **Format:** A profile of an **[RFC3339]** Date/Time. Fractional time is not supported. The server MUST
869 support the expanded version e.g.

```
870 2007-01-02T13:00:00-08:00
```

871 It is up to the server to interpret local date/times.

872 **Example:**

```
873 2007-02-03T15:30:00-0800
```

```
874 2007-12-01T10:15:00Z
```

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

877 Date-only values are disallowed as the server cannot determine the correct start of the day. Only
878 UTC or date/time with offset values are permitted.

879 10.2.2 end

880 **Default:** Same as start

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

882 **Format:** Same as start

883 **Example:** Same as start

884 10.2.3 period

885 **Default:** The default value is left up to the server. The recommended value is "P42D".

886 **Description:** Specifies the amount of Freebusy data to return. A client cannot specify both a period
887 and an end date. Period is relative to the start parameter.

888 **Format:** A duration as defined in section 4.3.6 of [RFC 5545]

889 **Example:**

890 P42D

891 10.2.4 account

892 **Default:** none

893 **Description:** Specifies the principal when the request is targeted at the XRD CalWS/principal-
894 freebusy. Specification of this parameter is an error otherwise.

895 **Format:** Server specific

896 **Example:**

897 fred
898 /principals/users/jim
899 user1@example.com

900 10.3 URL parameters - notes

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

903 A client MUST check the time range in the VFREEBUSY response as a server may return a different time
904 range than the requested range.

905 10.4 HTTP Operations

906 The server SHOULD return an Etag response header for a successful GET request targeting a Freebusy
907 read URL. Clients MAY use the Etag response header value to do subsequent "conditional" GET
908 requests that will avoid re-sending the Freebusy data again if it has not changed.

909 10.5 Response Codes

910 Below are the typical status codes returned by a GET request targeting a Free-busy URL. Note that other
911 HTTP status codes not listed here might also be returned by a server.

- 912 • 200 OK
- 913 • 302 Found
- 914 • 400 Start parameter could not be understood / End parameter could not be understood / Period
915 parameter could not be understood
- 916 • 401 Unauthorized
- 917 • 403 Forbidden
- 918 • 404 The data for the requested principal is not currently available, but may be available later.
- 919 • 406 The requested format in the accept header is not supported.
- 920 • 410 The data for the requested principal is no longer available

921 | • 500 General server error

922 10.6 Examples

923 The following are examples of URLs used to retrieve Free-busy data for a user:

```
924 http://www.example.com/freebusy/user1@example.com?
925 start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00
926
927 http://www.example.com/freebusy/user1@example.com?
928 start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00
929
930 http://www.example.com/freebusy/user1@example.com
931
932 http://www.example.com/freebusy?user=user%201@example.com&
933 start=2008-01-01T00:00:00Z&end=2008-12-31T00:00:00Z
```

934 Some Request/Response Examples:

935 A URL with no query parameters:

```
936 >> Request <<
937 GET /freebusy/bernard/ HTTP/1.1
938 Host: www.example.com
939
940 >> Response <<
941 HTTP/1.1 200 OK
942 Content-Type: application/xml+calendar; charset="utf-8"
943 Content-Length: xxxx
944
945 <xc:icalendar xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
946   <xc:vcalendar>
947     <xc:properties>
948       <xc:calscale><text>GREGORIAN</text></xc:calscale>
949       <xc:prodid>
950         <xc:text>-//Example Inc.//Example Calendar//EN</xc:text>
951       </xc:prodid>
952       <xc:version><xc:text>2.0</xc:text></xc:version>
953     </xc:properties>
954     <xc:components>
955       <xc:vfreebusy>
956         <xc:properties>
957           <xc:uid>
958             <xc:text>76ef34-54a3d2@example.com</xc:text>
959           </xc:uid>
960           <xc:dtstart>
961             <xc:date-time>20060101T000000Z</xc:date-time>
962           </xc:dtstart>
963           <xc:dtend>
964             <xc:date-time>20060108T000000Z</xc:date-time>
965           </xc:dtend>
966           <xc:dtstamp>
967             <xc:date-time>20050530T123421Z</xc:date-time>
968           </xc:dtstamp>
969           <xc:freebusy>
970             <xc:parameters>
971               <xc:fbs-type>BUSY TENTATIVE<xc:fbs-type>
972             </xc:parameters>
973             <xc:period>20060102T100000Z/20060102T120000Z</xc:period>
974           </xc:freebusy>
975           <xc:freebusy>
976             <xc:period>20060103T100000Z/20060103T120000Z</xc:period>
977           </xc:freebusy>
978           <xc:freebusy>
979             <xc:period>20060104T100000Z/20060104T120000Z</xc:period>
980           </xc:freebusy>
```

```

981     <xc:freebusy>
982         <xc:parameters>
983             <xc:fbtype>BUSYUNAVAILABLE<xc:fbtype>
984         <xc:parameters>
985         <xc:period>20060105T100000Z/20060105T120000Z</xc:period>
986     </xc:freebusy>
987     <xc:freebusy>
988         <xc:period>20060106T100000Z/20060106T120000Z</xc:period>
989     </xc:freebusy>
990 </xc:vfreebusy>
991 </xc:components>
992 </xc:vcalendar>
993 <xc:icalendar>

```

994 A URL with start and end parameters:

```

995 >> Request <<
996 GET /freebusy/user1@example.com?start=2007-09-01T00:00:00-08:00&end=2007-09-
997 31T00:00:00-08:00
998 HTTP/1.1
999 Host: www.example.com
1000
1001 >> Response <<
1002 HTTP/1.1 200 OK
1003 Content-Type: application/xml+calendar; charset="utf-8"
1004 Content-Length: xxxx
1005
1006 <xc:icalendar xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
1007   <xc:vcalendar>
1008     <xc:properties>
1009       <xc:calscale><text>GREGORIAN</text></xc:calscale>
1010       <xc:prodid>
1011         <xc:text>-//Example Inc.//Example Calendar//EN</xc:text>
1012       </xc:prodid>
1013       <xc:version><xc:text>2.0</xc:text></xc:version>
1014     </xc:properties>
1015     <xc:components>
1016       <xc:vfreebusy>
1017         <xc:properties>
1018           <xc:uid>
1019             <xc:text>76ef34-54a3d2@example.com</xc:text>
1020           </xc:uid>
1021           <xc:dtstart>
1022             <xc:date-time>20070901T000000Z</xc:date-time>
1023           </xc:dtstart>
1024           <xc:dtend>
1025             <xc:date-time>20070931T000000Z</xc:date-time>
1026           </xc:dtend>
1027           <xc:dtstamp>
1028             <xc:date-time>20050530T123421Z</xc:date-time>
1029           </xc:dtstamp>
1030           <xc:freebusy>
1031             <xc:period>20070915T230000Z/20070916T010000Z</xc:period>
1032           </xc:freebusy>
1033         </xc:vfreebusy>
1034       </xc:components>
1035     </xc:vcalendar>
1036   </xc:icalendar>

```

1037 A URL for which the server does not have any data for that user:

```

1038 >> Request <<
1039 GET /freebusy/user1@example.com?start=2012-12-01T00:00:00-08:00&end=2012-12-
1040 31T00:00:00-08:00
1041 HTTP/1.1
1042 Host: www.example.com
1043

```

1044 >> Response <<
1045 HTTP/1.1 404 No data
1046

1047 11 Conformance

1048 ~~The last numbered section in the specification must be the Conformance section. Conformance~~
1049 ~~Statements/Clauses go here.~~

1050 ~~Certain calendaring properties and components are interrelated and it is necessary to have knowledge of~~
1051 ~~all these properties and their current values to allow consistent update and understanding of a target~~
1052 ~~component. The normative definition for these relationships is RFC5445, RFC5446 and related RFCs.~~

1053 ~~As in those specifications this REST-ful protocol assumes a complete view of entities being fetched or~~
1054 ~~updated. This is necessary to ensure that properties are not lost when round tripped through a service or~~
1055 ~~client. To this end all parties in any RESTful transaction MUST preserve any data they do not understand.~~
1056 ~~This allows the data model to be updated by the addition of properties, parameters and value types.~~

1057 ~~Services allowing updates to entities MUST ensure that the result after an update operation is still~~
1058 ~~internally consistent.~~

1059 **11.1 Start, end and duration in calendar components**

1060 ~~A period of time is fully specified by a start and an end or duration.~~

1061 **11.1.1 Updating, transporting and maintaining start, and and duration.**

- 1062 • ~~For all components the calculated or specified start must be at or before the end.~~
- 1063 • ~~When a system updates or stores a calendar component it MUST retain the relationship of start,~~
1064 ~~end and duration. Applications MUST NOT without good cause, change a start and end pair into~~
1065 ~~a start and duration nor the reverse. Semantically they are not equivalent when DST transitions~~
1066 ~~occur during the time of the event.~~
- 1067 • ~~For interoperability, iCalendar based systems SHOULD avoid the use of weekly durations and~~
1068 ~~XML based systems SHOULD avoid the use of yearly durations.~~

1069 **11.1.2 VEVENT:**

- 1070 • ~~The three properties are DTSTART, DTEND and DURATION.~~
- 1071 • ~~DTSTART MUST appear once and only one of DTEND or DURATION MAY be present.~~
- 1072 • ~~The DTSTART property for a VEVENT specifies the inclusive start of the event. For recurring~~
1073 ~~events, it also specifies the very first instance in the recurrence set.~~
- 1074 • ~~The DTEND property for a VEVENT calendar component specifies the non-inclusive end of the~~
1075 ~~event.~~
- 1076 • ~~For cases where a VEVENT calendar component specifies a DTSTART property with a DATE~~
1077 ~~value type but no DTEND nor DURATION property, the event's duration is taken to be one day.~~
- 1078 • ~~For cases where a VEVENT calendar component specifies a DTSTART property with a DATE-~~
1079 ~~TIME value type but no DTEND nor DURATION property, the event ends on the same calendar~~
1080 ~~date and time of day specified by the DTSTART property, that is, it signifies a zero length instant~~
1081 ~~in time.~~

1082 **11.1.3 VTODO:**

- 1083 • ~~The three properties are DTSTART, DUE, DURATION.~~
- 1084 • ~~DTSTART MAY appear once.~~

- 1085
- 1086
- Either DUE or DURATION MAY appear in a VTODO, but DUE and DURATION MUST NOT occur in the same VTODO.
- 1087
- If DURATION does appear in a VTODO, then DTSTART MUST also appear in the same VTODO.
- 1088
- The three properties for a VTODO are related in the same way as for VEVENT. Additionally a VTODO calendar component without the DTSTART and DUE (or DURATION) properties specifies a VTODO that will be associated with each successive calendar date, until it is completed.
- 1089
- 1090
- 1091

1092 **11.1.4 VJOURNAL:**

- 1093
- DTSTART only, which may be a date or date-time value.

1094 **11.1.5 VAVAILABILITY**

- 1095
- DTSTART and DTEND if specified MUST be date-time values.
- 1096
- DTSTART MAY appear once and signifies start of the busy period.
- 1097
- Only one of DTEND or DURATION MAY appear and signify the end of the busy period.
- 1098
- If DURATION does appear in a VAVAILABILITY, then DTSTART MUST also appear in the same VAVAILABILITY.
- 1099

1100 **11.1.6 AVAILABILITY**

- 1101
- DTSTART and DTEND if specified MUST be date-time values.
- 1102
- DTSTART MUST appear once and signifies start of the free period.
- 1103
- Only one of DTEND or DURATION MAY appear and signify the end of the free period.

1104 **11.2 Recurrences.**

- 1105
- The RECURRENCE-ID is a property of each instance of a recurring event. It is calculated from the DTSTART and the recurrence rules or added to the set by the RDATE property.
- 1106
- RDATE, EXDATE and RECURRENCE-ID must take the same form as the DTSTART. That is if DTSTART is a DATE value then the RDATE and EXDATE must be DATE. If DTSTART is a date-time the RDATE and EXDATE values must take the same form, including the same timezone.
- 1107
- Overrides to an instance are specified by completely specifying the instance with the appropriate RECURRENCE-ID property.
- 1108
- An RDATE adds an instance to the recurrence set.
- 1109
- An EXDATE deletes an instance by specifying the recurrence id(s) to be deleted. Applications SHOULD NOT specify overrides for instances so deleted.
- 1110
- The recurrence set is calculated from the RRULE and RDATES and then applying any EXDATE properties. That is EXDATE takes precedence over RDATE and the RRULE.
- 1111
- 1112
- 1113
- 1114
- 1115
- 1116

1117 **11.3 Alarms:**

- 1118
- Alarms are typically anchored to the start or end of an event or task. This is defined by the RELATED parameter to the TRIGGER property.
- 1119

1120

11.4 Unrecognized or unsupported elements

1121

- A system SHOULD reject any attempt to store components which it does not support. A SYSTEM MUST respond with a CalWS:unsupported-calendar-component if such an attempt is made.

1122

1123

- A system MUST ignore but preserve any elements it does not understand.

1124

1125 Appendix A. Acknowledgments

1126 The following individuals have participated in the creation of this specification and are gratefully
1127 acknowledged:

1128 | **Participants:**

1129 | Bruce Bartell, Southern California Edison
1130 | Brad Benson, Trane
1131 | Edward Cazalet, Individual
1132 | Toby Considine, University of North Carolina at Chapel Hill
1133 | William Cox, Individual
1134 | Sharon Dinges, Trane
1135 | Mike, Douglass, Rensselaer Polytechnic Institute
1136 | Craig Gemmill, Tridium, Inc.
1137 | Girish Ghatikar, Lawrence Berkeley National Laboratory
1138 | Gerald Gray, Southern California Edison
1139 | David Hardin, ENERNOC
1140 | Gale Horst, Electric Power Research Institute (EPRI)
1141 | Gershon Janssen, Individual
1142 | Ed Koch, Akuacom Inc.
1143 | Benoit Lepeuple, LonMark International*
1144 | | Carl Mattocks, CheckMi*
1145 | Robert Old, Siemens AG
1146 | Alexander Papaspyrou, Technische Universitat Dortmund
1147 | | Joshua Phillips, ISO/RTO Council (IRC)
1148 | Jeremy J. Roberts, LonMark International
1149 | David Thewlis, CalConnect
1150

1151 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-
1152 Calendar Technical Committee, bridging to developing IETF standards and contributing the services
1153 definitions that make up Services in Section 4. The Technical Committee gratefully acknowledges their
1154 assistance and cooperation as well. Contributors to TC XML include:

1155 | Cyrus Daboo, Apple
1156 | Mike Douglass, Rensselaer Polytechnic Institute
1157 | Steven Lees, Microsoft
1158 | Tong Li, IBM
1159

1160 **Appendix B. An Introduction to Internet Calendaring**

1161 *The WS-Calendar Technical Committee thanks CalConnect for contributing this overview of iCalendar*
1162 *and its use.*

1163 **B.1 icalendar**

1164 **B.1.1 History**

1165 The iCalendar specification was first produced by the IETF in 1998 as RFC 2445 [1]. Since then it has
1166 become the dominant standard for calendar data interchange on the internet and between devices
1167 (desktop computers, mobile phones etc.). The specification was revised in 2009 as RFC 5545 [4].

1168 Alongside iCalendar is the iTIP specification (RFC 2446 [2] and revised as RFC 5546[5]) that defines how
1169 iCalendar is used to carry out scheduling operations (for example, how an organizer can invite attendees
1170 to a meeting and receive their replies). This forms the basis for email-based scheduling using iMIP (the
1171 specification that describes how to use iTIP with email - RFC 6047 [3]).

1172 iCalendar itself is a text-based data format. However, an XML format is also available, providing a one-to-
1173 one mapping to the text format (draft [7]).

1174 iCalendar data files typically have a .ics file name extension. Most desktop calendar clients can import or
1175 export iCalendar data, or directly access such data over the Internet using a variety of protocols.

1176 **B.1.2 Data model**

1177 The iCalendar data format has a well defined data model. "iCalendar objects" encompass a set of
1178 "iCalendar Components" each of which contains a set of "iCalendar properties" and possibly other sub-
1179 Components. An iCalendar property consists of a name, a set of optional parameters (specified as "key-
1180 value" pairs) and a value.

1181 iCalendar Components include:

1182 "VEVENT" which represents an event

1183 "VTODO" which represents a task or to-do

1184 "VJOURNAL" which represents a journal entry

1185 "VFREEBUSY" which represents periods of free or busy time information

1186 "VTIMEZONE" which represents a timezone definition (timezone offset and daylight saving rules)

1187 "VALARM" is currently the only defined sub-Component and is used to set alarms or reminders on events
1188 or tasks.

1189 Properties include:

1190 "DTSTART" which represents a start time for a Component

1191 "DTEND" which represents an end time for a Component

1192 "SUMMARY" which represents a title or summary for a Component

1193 "RRULE" which can specify rules for repeating events or tasks (for example, every day, every week on
1194 Tuesdays, etc.)

1195 "ORGANIZER" which represents the calendar user who is organizing an event or assigning a task

1196 "ATTENDEE" which represents calendar users attending an event or assigned a task

1197 In addition to this data model and the pre-defined properties, the specification defines how all those are
1198 used together to define the semantics of calendar objects and scheduling. The semantics are basically a
1199 set of rules stating how all the Components and properties are used together to ensure that all iCalendar
1200 products can work together to achieve good interoperability. For example, a rule requires that all events
1201 must have one and only one "DTSTART" property. The most important part of the iCalendar specification

1202 is the semantics of the calendaring model that it represents. The use of text or XML to encode those is
1203 secondary.

1204 **B.1.3 Scheduling**

1205 The iTIP specification defines how iCalendar objects are exchanged in order to accomplish the key task
1206 needed to schedule events or tasks. An example of a simple workflow is as follows:

- 1207 1. To schedule an event, an organizer creates the iCalendar object representing the event and adds
1208 calendar users as attendees.
- 1209 2. The organizer then sends an iTIP "REQUEST" message to all the attendees.
- 1210 3. Upon receipt of the scheduling message, each attendee can decide whether they want to attend
1211 the meeting or not.
- 1212 4. Each attendee can then respond back to the organizer using an iTIP "REPLY" message
1213 indicating their own attendance status.

1214 iTIP supports other types of scheduling messages, for example, to cancel meetings, add new instances to
1215 a repeating meeting, etc.

1216 **B.1.4 Extensibility**

1217 iCalendar was designed to be extensible, allowing for new Components, properties and parameters to be
1218 defined as needed. A registry exists to maintain the list of standard extensions with references to their
1219 definitions to ensure anyone can use them and work well with others.

1220 **B.2 Calendar data access and exchange protocols**

1221 **B.2.1 Internet Calendar Subscriptions**

1222 An Internet calendar subscription is simply an iCalendar data file made available on a web server. Users
1223 can use this data in two ways:

- 1224 – The data can be downloaded from the web server and then imported directly into an iCalendar
1225 aware client. This solution works well for calendar data that is not likely to change over time (for
1226 example the list of national holidays for the next year).
- 1227 – Calendar clients that support "direct" subscriptions can use the URL to the calendar data on the
1228 web server to download the calendar data themselves. Additionally, the clients can check the web
1229 server on a regular basis for updates to the calendar data, and then update their own cached
1230 copy of it. This allows calendar data that changes over time to be kept synchronized.

1231 **B.2.2 CalDAV**

1232 CalDAV is a calendar access protocol and is defined in RFC 4791 [6]. The protocol is based on WebDAV
1233 which is an extension to HTTP that provides enhanced capabilities for document management on web
1234 servers.

1235 CalDAV is used in a variety of different environments, ranging from very large internet service providers,
1236 to large and small corporations or institutions, and to small businesses and individuals.

1237 CalDAV clients include desktop applications, mobile devices and browser-based solutions. It can also be
1238 used by "applets", for example, a web page panel that displays a user's upcoming events.

1239 One of the key aspects of CalDAV is its data model. Simply put, it defines a "calendar home" for each
1240 calendar user, within which any number of "calendars" can be created. Each "calendar" can contain any
1241 number of iCalendar objects representing individual events, tasks or journal entries. This data model
1242 ensures that clients and servers can interoperate well.

1243 In addition to providing simple operations to read, write and delete calendar data, CalDAV provides a
1244 querying mechanism to allow clients to fetch calendar data matching specific criteria. This is commonly

1245 used by clients to do "time-range" queries, i.e., find the set of events that occur within a given start/end
1246 time period.

1247 CalDAV also supports access control allowing for features such as delegated calendars and calendar
1248 sharing.

1249 CalDAV also specifies how scheduling operations can be done using the protocol. Whilst it uses the
1250 semantics of the iTIP protocol, it simplifies the process by allowing simple calendar data write operations
1251 to trigger the sending of scheduling messages, and it has the server automatically process the receipt of
1252 scheduling messages. Scheduling can be done with other users on the CalDAV server or with calendar
1253 users on other systems (via some form of "gateway").

1254 **B.2.3 ActiveSync/SyncML**

1255 ActiveSync and SyncML are technologies that allow multiple devices to synchronize data with a server,
1256 with calendar data being one of the classes of data supported. These have typically been used for low-
1257 end and high-end mobile devices.

1258 **B.2.4 CalWS**

1259 CalWS ~~is~~ refers to a set of web services calendar access APIs developed by under a cooperative
1260 agreement between The Calendaring and Scheduling Consortium (CalConnect) and ~~the~~ OASIS
1261 organization, to be used, and being published as part a work product of the Oasis-WS-Calendar standard.
1262 ~~It provides~~ Technical Committee. CalWS defines an API to access and manipulate calendar data stored
1263 on a server. It follows a similar data model to CalDAV and has been designed to co-exist with a CalDAV
1264 service offering the same data.

1265 This specification is part of the CalWS set.

1266 **B.2.5 iSchedule**

1267 iSchedule is a protocol to allow scheduling between users on different calendaring systems and across
1268 different internet domains. It transports iTIP scheduling messages using HTTP between servers. Servers
1269 use DNS and various security mechanisms to determine the authenticity of messages received.

1270 It has been specifically designed to be independent of any calendar system in use at the endpoints, so
1271 that it is compatible with many different systems. This allows organizations with different calendar
1272 systems to exchange scheduling messages with each other, and also allows a single organization with
1273 multiple calendar systems (for example due to mergers, or different departmental requirements) to
1274 exchange scheduling messages between users of each system.

1275 **B.3 References**

1276 [1] <https://datatracker.ietf.org/doc/rfc2445/> : 'Internet Calendaring and Scheduling Core Object
1277 Specification'

1278 [2] <https://datatracker.ietf.org/doc/rfc2446/> : 'iCalendar Transport-Independent Interoperability Protocol'

1279 [3] <https://datatracker.ietf.org/doc/rfc6047/> : 'iCalendar Message-Based Interoperability Protocol'

1280 [4] <https://datatracker.ietf.org/doc/rfc5545/> : 'Internet Calendaring and Scheduling Core Object
1281 Specification'

1282 [5] <https://datatracker.ietf.org/doc/rfc5546/> : 'iCalendar Transport-Independent Interoperability Protocol'

1283 [6] <https://datatracker.ietf.org/doc/rfc4791/> : 'Calendaring Extensions to WebDAV'

1284 [7] <https://datatracker.ietf.org/doc/draft-daboo-et-al-icalendar-in-xml/> : 'xCal: The XML format for
1285 iCalendar'

1286

1287

Appendix C. Revision History

Revision	Date	Editor	Changes Made
ws-calendar-wd19	19-Mar-2011	Toby Considine	Originally contributed by Mike Douglass as part of WS-Calendar v1.0 Specification. See full history in that document.
WD02	13-Feb-2012	Toby Considine	Ported to separate document. "Promoted" all section headers.
WD03	15 Feb-2012	Toby Considine	Added Intro, updated namespaces to meet OASIS standard
WD04	17 February 2012	Toby Considine	Additional namespace clean-up in response to Cover comments. Consistent capitalization of calWS when used as a namespace identifier Clean-up of CalWS discussion in appendix
WD05	17 February 2012	Toby Considine	Types, capitalization, missing XRD reference
WD06	29 January 2013	Mike Douglas	Added Conformance