

WS-Calendar Version 1.0

Working Draft 06

19 May 2010

Specification URIs:

This Version:

<http://docs.oasis-open.org/ws-calendar/v1.0/wd06/ws-calendar-1.0-spec-wd-06.pdf>
<http://docs.oasis-open.org/ws-calendar/v1.0/wd06/ws-calendar-1.0-spec-wd-06.html>
<http://docs.oasis-open.org/ws-calendar/v1.0/wd06/ws-calendar-1.0-spec-wd-06.doc>

Previous Version:

N/A

Latest Version:

<http://docs.oasis-open.org/ws-calendar/v1.0/ws-calendar-1.0-spec.pdf>
<http://docs.oasis-open.org/ws-calendar/v1.0/ws-calendar-1.0-spec.html>
<http://docs.oasis-open.org/ws-calendar/v1.0/ws-calendar-1.0-spec.doc>

Technical Committee:

OASIS WS-Calendar TC

Chair(s):

Toby Considine

Editor(s):

Toby Considine

Related work:

This specification replaces or supersedes:

N/A

This specification is related to:

IETF RFC 5545, iCalendar
IETF RFC 5546, iCalendar Transport
IETF RFC 2447, iCalendar Message Based Interoperability
IETF / CalConnect xCal specification in progress
IETF / CalConnect Calendar Resource Schema specification in progress
CalConnect CalWS Web Services specification in progress

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/WS-Calendar/WS-Calendar-201001>

Abstract:

WS-Calendar describes a common set of message components for specifying schedules and intervals to coordinate activities between services. Web services definitions, service definitions consistent with the OASIS SOA Reference Model, and XML vocabularies for the interoperable and standard exchange of:

- Schedules, including sequences of schedules
- Intervals, including sequences of intervals

The definition of the service performed to meet a schedule or interval depends on the market context in which it exists. It is not in scope for this TC to define those markets or services.

Status:

This document was last revised or approved by the WS-Calendar Technical Committee on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification 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 specification, 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>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/WS-Calendar/>.

Notices

Copyright © OASIS® 2010. 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 names "OASIS", [insert specific trademarked names and abbreviations here] are trademarks 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/who/trademark.php> for above guidance.

Table of Contents

1	Introduction	5
1.1	Terminology	5
1.2	Normative References	6
1.3	Non-Normative References	6
1.4	Naming Conventions	6
1.5	Architectural References	7
2	Overview of WS-Calendar	8
2.1	Scope of Work of the WS-Calendar Technical Committee	8
2.2	Deliverables	8
3	WS-Calendar Elements	9
3.1	Core Semantics xCal	9
3.1.1	Time	9
3.1.2	Segmenting Time	9
3.1.3	Alarms	10
3.1.4	Attachments	10
3.2	Time Stamps	10
3.3	Service Execution	11
4	Intervals and Process Synchronization	13
4.1	Intervals	13
4.1.1	vtodo elements	13
4.1.2	Intervals: Unscheduled Time Segments	13
4.1.3	Periods: Scheduled Time Segment	14
4.2	Notification and Synchronization	16
4.2.1	Inter-service Notifications	16
5	Service Interactions	17
5.1	Use of Cal-WS	17
6	WS-Calendar Structures	18
6.1	Partition Set	18
6.2	Period Set	18
7	Sequence of Operations	20
8	Conformance	21
	Acknowledgements	22
	Background	23
	Revision History	24

1 Introduction

One of the most fundamental components of negotiating services is agreeing when something should occur, and in auditing when they did occur. Short running services have traditionally been handled as if they were instantaneous, and thereby dodged this requirement through just-in-time requests. Longer running processes may require significant lead times. When multiple long-running services participate in the same business process, it may be more important to negotiate a common completion time than a common start time. Central coordination of such services reduces interoperability as it requires the coordinating agent to know the lead time of each service.

Other processes may have multiple and periodic occurrence. Identical processes may need to be requested on multiple schedules. Other processes must be requested to coincide with or avoid human interactions. An example is a process that occurs on the first Tuesday of every month. Others may need to be completed on schedules that vary by local time zone.

Physical processes are now being coordinated by web services. Building systems and industrial processes are operated using oBIX, BACnet/WS, LON-WS, OPC XML, and a number of proprietary specifications including TAC-WS, Gridlogix EnNet, and MODBUS.NET. Energy use in buildings can be reduced while improving performance if building systems are coordinated with the schedules of the buildings occupants.

An increasing number of specifications envision synchronization of processes through mechanisms including broadcast scheduling. Efforts to build an intelligent power grid (or smart grid) rely on coordinating processes in homes, offices, and industry with projected and actual power availability, including different prices at different times. Two active OASIS Technical Committees require a common means to specify schedule and interval: Energy Interoperation (EITC) and Energy Market Information Exchange (EMIX). Emergency management coordinators wish to inform geographic regions of future events, such as a projected tornado touchdown, using EDXL. These efforts would benefit from a common standard for transmitting schedule and interval.

For human interactions and human scheduling, the well-known iCalendar format is used. Today, there is no equivalent standard for web services. As an increasing number of physical processes are managed by web services, the lack of a similar standard for scheduling and coordination of services becomes critical.

The goal of WS-Calendar is to adapt the existing specifications for calendaring and apply them to develop a standard for how schedule and event information is passed between and within services. The standard should adopt the semantics and vocabulary of iCalendar for application to the completion of web service contracts. WS Services will be built on work done and ongoing in The Calendaring and Scheduling Consortium (CalConnect), in particular the xCal, Calendar Resource Schema, and CalWS specifications.

A calendar event without an associated contract is of little use. WS-Calendar will be a micro-specification, and then a micro-standard. WS-Calendar is unlikely to be used by itself. WS-Calendar will instead be used inside other specifications and standards, bringing a common scheduling function to diverse interactions in different domains.

Special impetus to develop the WS-Calendar Specification is derived from the National Institute of Standards and Technology (NIST) Smart Grid Interoperability Road Map prepares in support of the US Department of Energy (DOE) as described in the Energy Independence and Security Act of 2007 (EISA 2007). The roadmap stated that common communication of schedule and interval is a cross-cutting critical to overall roadmap success.

A. All examples and all Appendices are non-normative.

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**.

48 1.2 Normative References

- 49 **RFC2119** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
50 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 51 **SOA-RM** OASIS Standard, *Reference Model for Service Oriented Architecture 1.0*,
52 October 2006.
53 <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>
- 54 **RFC5545** B. Desruisseaux *Internet Calendaring and Scheduling Core Object*
55 *Specification (iCalendar)*, <http://www.ietf.org/rfc/rfc5545.txt>, IETF RFC
56 5545, September 2009.
- 57 **RFC5546** C. Daboo *iCalendar Transport-Independent Interoperability Protocol*
58 *(iTIP)*, <http://www.ietf.org/rfc/rfc5546.txt>, IETF RFC 5546, January 1999.
- 59 **RFC2447** F. Dawson, S. Mansour, S. Silverberg, *iCalendar Message-Based*
60 *Interoperability Protocol (iMIP)*, <http://www.ietf.org/rfc/rfc2447.txt>, IETF
61 RFC 2447, December 2009.
- 62 **xCal** C. Daboo, M Douglas, S Lees *xCal: The XML format for iCalendar*,
63 <http://tools.ietf.org/html/draft-daboo-et-al-icalendar-in-xml-03>, Internet-
64 Draft, April 2010.
- 65 **Calendar Resource Schema** C. Joy, C. Daboo, M Douglas, *Schema for representing*
66 *resources for calendaring and scheduling services*,
67 <http://tools.ietf.org/html/draft-cal-resource-schema-00>, (Internet-Draft),
68 April 2010.
- 69 **CalWS** CalConnect draft in Process.
- 70 **XPATH** A Berglund, S Boag, D Chamberlin, MF Fernández, M Kay, J Robie, J
71 Siméon *XML Path Language (XPath) 2.0*, <http://www.w3.org/TR/xpath20/>
72 January 2007.
- 73 **XLINK** S DeRose, E Maler, D Orchard, N Walsh *XML Linking Language (XLink)*
74 *Version 1.1.*, <http://www.w3.org/TR/xlink11/> May 2010.
- 75 **XPOINTER** S DeRose, E Maler, R Daniel Jr. *XPointer xpointer Scheme*,
76 <http://www.w3.org/TR/xptr-xpointer/> December 2002.
- 77 **XML Schema** PV Biron, A Malhotra, *XML Schema Part 2: Datatypes Second Edition*,
78 <http://www.w3.org/TR/xmlschema-2/> October 2004.

80 1.3 Non-Normative References

81

82 1.4 Naming Conventions

- 83 This specification follows some naming conventions for artifacts defined by the specification, as follows:
84 For the names of elements and the names of attributes within XSD files, the names follow the CamelCase
85 convention, with all names starting with a lower case letter.
86 eg <element name="componentType" type="WS-Calendar:ComponentType"/>
- 87 For the names of types within XSD files, the names follow the CamelCase convention with all names
88 starting with an upper case letter.
89 eg. <complexType name="ComponentService">
- 90 For the names of intents, the names follow the CamelCase convention, with all names starting with a
91 lower case letter, EXCEPT for cases where the intent represents an established acronym, in which case
92 the entire name is in upper case.

93 An example of an intent which is an acronym is the "SOAP" intent.

94 **1.5 Architectural References**

95 Energy Interoperability defines a service oriented approach to energy interactions. Accordingly it assumes
96 a certain amount of definitions of roles, names, and interaction patterns. This document relies heavily on
97 roles and interactions as defined in the OASIS Standard *Reference Model for Service Oriented*
98 *Architecture*.

99 2 Overview of WS-Calendar

100 A calendar event without an associated contract is of little use. WS-Calendar will be a micro-specification,
101 and then a micro-standard. WS-Calendar is unlikely to appear by itself. Other specifications and
102 standards will use WS-Calendar, bringing a common scheduling function to diverse interactions in
103 different domains. WS-Calendar components may also appear inside xCal messages as the schedule
104 payload.

105 2.1 Scope of Work of the WS-Calendar Technical Committee

106 The Committee will start work with the canonical XML serialization of the updated iCalendar (IETF RFC
107 5545), hereafter referred to as xCal, as developed by the Calendaring and Scheduling Consortium
108 (CalConnect.org). This work will provide the vocabulary for use in this web service component.

109 The committee will also use as a starting point the forthcoming calendaring web service specifications
110 being developed by CalConnect. This specification provides the basic mechanism for creating, updating,
111 and deleting calendar events that are common to all calendars and schedules.

112 The committee expects that use cases and requirements will be contributed by other groups, including
113 the NAESB task forces on smart grid prices and schedules for DR/DER as well as the work done within
114 the oBIX TC to schedule building systems. These use cases will test the completeness and functionality
115 of the specification.

116 The committee will develop additional Calendar functionality in later work, both in revisions and new
117 specifications. The committee will also accept additional use cases for profile development, centralizing
118 profile development to ensure consistency and interoperation of the additional schedule- and interval-
119 related components.

120 2.2 Deliverables

121 The committee will deliver a standard schema and semantics for schedule and interval information for use
122 in other web services. This specification will be derived from and compatible with the existing xCal
123 specification and offer some or all of the functionality of that specification.

124 The completed specification should include a standard for referring to instances of xCal within a larger
125 payload, as well as a means to refer to objects external to the xCal instances. A single calendar object
126 may be referenced by multiple contracts. A series of calendar events may reference a single contract.

127 The committee will deliver a specification for creating, retrieving, updating, and deleting calendar events
128 on a schedule. This specification will be based on the forthcoming calendar web service specifications
129 developed by CalConnect.

130 Geoposition is an optional component of the iCalendar structure. Several of the use cases would benefit
131 from geo-location. Some would benefit more from a point, and some from region or polygon. The
132 committee work will include recommendations on how to reference geospatial information, both point and
133 polygon, from within schedule and interval components. Preference will be given to specifications
134 promulgated by the Open Geospatial Consortium (OGC)¹.

¹ <http://www.opengeospatial.org/>

135 3 WS-Calendar Elements

136 WS-Calendar elements are semantic elements derived from xCal to standardize data and interval outside
137 of scheduling interactions. One should think of these elements as degenerate versions of xCal, as a blind
138 cave fish is a degenerate derivation of a fish outside the cave. The eyes may be gone, but all remaining
139 elements are the same.

140 WS-Calendar elements also elaborate the objects defined in iCalendar, to make interaction requirements
141 explicit. For example, in different organizations, meetings may start on the hour or within 5 minutes of the
142 hour. As agents scheduled in those organizations, people learn the expected precision. In WS-Calendar,
143 that precision must be explicit.

144 3.1 Core Semantics xCal

145 The iCalendar data format [RFC5545] is a widely deployed interchange format for calendaring and
146 scheduling data. The xCal specification standardizes the XML representation of iCalendar information.
147 WS-Calendar relies on xCal standards and data representation to develop its service components.

148 <http://ietfreport.isoc.org/idref/draft-daboo-et-al-icalendar-in-xml/>

149 3.1.1 Time

150 Time is an ISO8601 compliant time string with the optional accompaniment of a duration interval to define
151 times of less than 1 second.

152 3.1.2 Segmenting Time

153 Segmenting time is a critical component of service alignment using WS-Calendar. There are many
154 overloaded uses of terms about time, and within a particular time segment, there may be many of them.
155 Within WS-Calendar, we distinguish between them as defined in Table 1, below.

156 The base data type is xs:duration, which, as its name suggests, quantifies the passage of time. Within
157 iCalendar, Duration appears the component objects.

158 *Table 1: Defining Time Segments for WS-Calendar*

Time Segments defined	Definition
Duration	Well-known element from iCalendar and xCal, Duration is the length of a meeting scheduled using iCalendar or any of its derivatives. Duration is an optional parameter of xCal objects vevent and vtodo.
Interval	An interval is a fully qualified duration, meaning it begins with the duration and adds supporting information to define the service performance. An interval is an array of consecutive duration (which in the trivial case is a single duration). An interval is relocatable, i.e., it does not have a specific date and time.
Period	A period is an interval that is anchored to a specific date and time. Specific performance of a service contract always occurs in a period.

159 A fully qualified interval can have many further refinements of time-related performance communication.
160 There may be requirements of required precision. Minimal notification before performance may be
161 specified.

162 For service to service communications, WS-Calendar can communicate these expectations precisely. A
163 timely response may be within a few minutes of the target time, or it may require Tolerance in
164 milliseconds. As defined by the W3C, "All *·minimally conforming·* processors *·must·* support year values
165 *with a minimum of 4 digits (i.e., YYYY) and a minimum fractional second precision of milliseconds or three*

166 decimal digits (i.e. s.sss). However, *·minimally conforming· processors ·may· set an application-defined*
 167 *limit on the maximum number of digits they are prepared to support in these two cases, in which case that*
 168 *application-defined maximum number ·must· be clearly documented.”*

169 *Table 2: Response Characteristics*

Examples of Time Segments in WS-Calendar	Note (Non-Normative)
Tolerance	The time segment in which a response is required. If you must arrive at a meeting within 5 minutes of the scheduled time, then the Tolerance is expressed as a duration of 5 minutes.
RampTime	How long after a service begins responding can the service be fully operational.
ResponseTime	How long after a service is requested can it acknowledge its ability to perform. Alternately, what lead time is required to for demand of contract performance before performance begins.
ExerciseInterval	How much time does the recipient have to exercise an agreement
DeliveryInterval	How much time does the recipient have to deliver the product after an agreement is executed.

170 Time Segments are a necessary component of all specifications derived from or incorporating the WS-
 171 Calendar specification. The base iCalendar object for all time segments is the vtodo object as expressed
 172 in the xCal serialization. Some additional fields for precisions and performance management are defined.

173 3.1.3 Alarms

174 Alarms in WS-Calendar declare when to send notifications between services. Within a single service,
 175 alarms declare milestones and target times. The base iCalendar object for all alarms is the valarm object.

176 3.1.4 Attachments

177 iCalendar uses Attachments to carry unstructured information elaborating the event or alarm
 178 communication. Attachments can also be in the form of URIs pointing outside the iCalendar structure. WS-
 179 Calendar uses structured XML to communicate the substance of the request. The details of that xml
 180 artifact are domain-specific and are outside the scope of this document.

181 The XML artifact in the attachment may be in-line, i.e., contained within the vtodo or valarm object, or it
 182 may be found in another section of the same XML object, sharing the same message as WS-Calendar
 183 element, or it may be discovered by external reference. In essence, attachments request “perform as
 184 described here”, or “perform as described below”, or “perform as described elsewhere.”

185 WS-Calendar users XPointer to make attachment referrals. XPointer is a general purpose URI and XML
 186 traversal standard. It is recommended that specifications that incorporate WS-Calendar create rules
 187 restricting the general purpose artifact to meet the needs of the market or service being supported. These
 188 limiting choices will always reflect a balance of parsimony of expression vs. efficiency of operation.

189 3.2 Time Stamps

190 Time stamps are used everywhere in inter-domain service performance analysis and have particular use
 191 in smart grids to support event forensics. Time stamps must be assembled across multiple time zones.

192 Different systems may track time and therefore record events with different levels of Tolerance. It is
 193 entirely possible that a time stamp with a low Tolerance may appear to have occurred after events with
 194 high-Tolerance time-stamps that it caused. A fully qualified time-stamp includes the granularity measure.

Table 3: Aspects of Time Stamps

WS-Calendar Time Stamp Element	Specification (Normative)	Note (Non-Normative)
TimeStamp	WS-Calendar:time A fully qualified date and time of event	May include two objects as defined above.
Precision	As defined in granularity above, expressed in xs:duration.	Identifies whether one hour interval is indeed one hour or plus or minus some number of seconds and minutes
TimeStampRealm	The realm identifies where the time stamp originated	Within a realm, one can assume that time-stamped objects sorted by time are in the order of their occurrence. Between realms, this assumption is rebuttable.
LeapSecondsKnown	Xs:boolean	
ClockFailure	xs:boolean	Indicates that the time source of the sending device is unreliable
ClockNotSynchronized	xs:boolean	indicates that the time source of the sending device is not synchronized with the external UTC time
Accuracy	Xs:duration	represents the time accuracy class of the time source of the sending device relative to the external UTC time.
Attach	WS-Calendar:associatedArtifact, defined as above.	Contains either local description of service or reference to xml document describing service

196 3.3 Service Execution

197 Time semantics are critical to WS-Calendar. Services requested differently can have different effects on
 198 performance even though they appear to request the same time interval. This is inherent in the in the
 199 concept of a service oriented architecture.

200 As defined in the OASIS Reference Model for Service Oriented Architecture 1.0², service requests access
 201 the capability of a remote system.

202 *The purpose of using a capability is to realize one or more **real world effects**. At its core, an*
 203 *interaction is “an act” as opposed to “an object” and the result of an interaction is an effect (or a*
 204 *set/series of effects). This effect may be the return of information or the change in the state of*
 205 *entities (known or unknown) that are involved in the interaction.*

206 *We are careful to distinguish between public actions and private actions; private actions are*
 207 *inherently unknowable by other parties. On the other hand, public actions result in changes to the*
 208 *state that is shared between at least those involved in the current execution context and possibly*
 209 *shared by others. Real world effects are, then, couched in terms of changes to this **shared state***

² <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>

210 When one accesses a remote capability one is requesting specifically the real world effects. Consider
211 two service providers that offer the same service. One must start planning an hour or more in advance.
212 The second may be able to achieve the service in five minutes. The service start time is the time when
213 that service becomes available. If this were not true, then the two providers would provide quite different
214 services.

215 The complement of this is the scheduled end time. The party offering the service may need to ramp down
216 long running processes. Using for example energy demand response, if a system contracts to end energy
217 use by 3:00, it assumes the onus of turning everything off before 3:00.

218 Duration is how long a behavior is continued. If a service contracts to provide shed load for an hour, it is
219 not necessary for it to stop shedding load 65 minutes later (which may be the end of the work day). It
220 must, however, shed the agreed upon load during all of the 60 minutes.

221 In this way, the service scheduled to shed load from 4:00 ending at 5:00 may be quite different than the
222 one scheduled to shed load for an hour beginning at 4:00.

223 4 Intervals and Process Synchronization

224 WS-Calendar-based specifications shall derive objects for communicating intervals and for synchronizing
225 time from the corresponding iCalendar objects. Within an iCalendar message, there is an outer envelope
226 containing transaction and synchronization information. The use of those fields is discussed below in
227 section 5 under Service Interactions.

228 Within this outer envelope there is a components section which can hold one or many iCalendar
229 components. Traditional calendar sharing has tended to use only one or two components, say a single
230 meeting (vevent) or perhaps a task (vtodo) and a request to warn the recipient of the impending due date
231 in advance (valarm). For historic reasons, each of the components is prefixed with the letter "v".

232 Within WS-Calendar, these components can be strung together to create packages of service interactions
233 and market operations.

234 4.1 Intervals

235 An interval specifies a single segment of time. Interval communications are derived from the vtodo object.
236 For ease of reference, the vtodo object is described here. In all cases, implementers SHALL refer to RFC
237 5545 and the xCal specifications for the normative description and definitions.

238 4.1.1 vtodo elements

239 While all elements are legal, certain elements are more critical when invoking services.

240 4.1.2 Intervals: Unscheduled Time Segments

241 An interval specifies how long an activity lasts. An unscheduled Interval is not linked to a specific date
242 and time.

243 *Table 4: Interval Data Elements*

Elements	Use	Discussion
Dtstamp	Mandatory	
Uid	Mandatory	Used to enable unambiguous referencing of each vtodo object
Class	Optional	
Duration	Mandatory	
DurationPrecision	Optional	Duration element used to specify margin of error on Duration
Attach	Mandatory, Multipleoccurs	Contains XML Artifact defining performance or xPointer to artifact defining performance. If repeated, can refer to multiple artifacts
Related	Optional compound element	Defines relations to other components. May be mandatory in derived specifications, esp. if component order is important.

244 The example below shows the components section of a WS-Calendar event containing two consecutive
245 15 minute time segments. They are listed in order. As XML parsing is not guaranteed to result in the
246 same order, each has a UID and a relationship. defining the order that each will occur. No start date is

247 specified in these time segments. As they are linked to each other, they describe a service of 30 minutes
 248 total made up of two consecutive segments.

```

249 <components>
250   <vtodo>
251     <dtstamp></dtstamp>
252     <uid>aaaaaaa1</uid>
253     <description>first contract</description>
254     <priority>high</priority>
255     <summary>defines first behavior to perform in contract with a precisions
256     required of 1 second</summary>
257     <duration>15m</duration>
258     <durationPrecision>1s</durationPrecision>
259     <attach>http://scheduled.services.com/contract1</attach>
260     <related-to>
261       <relationship>
262         <uid>aaaaaaa2</uid><reltype>PRECEDES</reltype>
263       </relationship>
264     </related-to>
265   </vtodo>
266   <vtodo>
267     <dtstamp></dtstamp>
268     <uid>aaaaaaa2</uid>
269     <description>second interval</description>
270     <priority>high</priority>
271     <summary>defines second behavior to perform in contract with a precision
272     required of 1 second</summary>
273     <duration>15m</duration>
274     <durationPrecision>1s</durationPrecision>
275     <attach>http://scheduled.services.com/contract2</attach>
276     <related-to>
277       <relationship>
278         <uid>aaaaaaa1</uid><reltype>FOLLOWS</reltype>
279       </relationship>
280     </related-to>
281   </vtodo>
282 </components>
  
```

```

283
284 <components>
  
```

285 Question for reviewers: we will need to expand the old list of standard relationships. What do we need? I
 286 can think of:

- 287 - PRECEDES
- 288 - FOLLOWS
- 289 - ENDSWHEN
- 290 - STARTSWHEN
- 291 - FINISHBEFORE

292 4.1.3 Periods: Scheduled Time Segment

293 A Period is an interval associated with a specified Time.

Elements	Use	Discussion
Dtstamp	Mandatory	

Uid	Mandatory	Used to enable unambiguous referencing of each vtodo object
Class	Optional	
DtStart	Mandatory	Date and Time segment begins
DtStartPrecision	Optional	Duration element used to specify precision of start time
DtEnd	Optional	iCalendar specifies either DtEnd or Duration, but not both
DtEndPrecision	Optional	Duration element used to specify precision of required end time
Duration	Optional	iCalendar specifies either DtEnd or Duration, but not both
DurationPrecision	Optional	Duration element used to specify margin of error on Duration
Attach	Mandatory, Multipleoccurs	Contains XML Artifact defining performance or xPointer to artifact defining performance. If repeated, can refer to multiple artifacts
Related	Optional compound element	Defines relations to other components. May be mandatory in derived specifications, esp. if component order is important.

294 The example below shows a service that must be performed for beginning at 22:00 UCT and continuing
295 until 22:30.

```

296 <components>
297   <vtodo>
298     <dtstamp></dtstamp>
299     <uid>aaaaaaa1</uid>
300     <description>first contract</description>
301     <priority>high</priority>
302     <summary>defines first behavior to perform in contract with a precisions
303     required of 1 second</summary>
304     <dtstart>20100524T220000Z</dtstart>
305     <dtend>20100524T223000Z</dtend>
306     <duration>15m</duration>
307     <durationPrecision>1s</durationPrecision>
308     <attach>http://scheduled.services.com/contract1</attach>
309     <related-to>
310       <relationship>
311         <uid>aaaaaaa2</uid><reltype>PRECEDES</reltype>
312       </relationship>
313     </related-to>
314   </vtodo>
315   <vtodo>
316     <dtstamp></dtstamp>
317     <uid>aaaaaaa1</uid>
318     <description>second interval</description>
319     <priority>high</priority>
320     <summary>defines second behavior to perform in contract with a precision
321     required of 1 second</summary>

```

```
322         <duration>15m</duration>
323         <durationPrecision>1s</durationPrecision>
324         <attach>http://scheduled.services.com/contract2</attach>
325         <related-to>
326             <relationship>
327                 <uid>aaaaaaa1</uid><reltype>FOLLOWS</reltype>
328             </relationship>
329         </related-to>
330     </vtodo>
331 </components>
```

332 Note that the second interval is pure duration, showing both kinds of intervals in the same artifact.

333 4.2 Notification and Synchronization

334 An alarm notifies another party that something has happened. Some alarms, such as alarm clocks, are
335 scheduled in advance. Others arise as a surprise from another system. Actual alarm mechanisms and
336 communications are outside the scope of this document. WS-Eventing, oBIX alarms, and CAP and EDXL
337 alerts are just a few of the already defined mechanisms.

338 This section discusses how the iCalendar valarm object is used in WS-Calendar. Alarms in a client server
339 world are receiving a lot of attention in enterprise scheduling right now and some details may change
340 before final publication.

341 4.2.1 Inter-service Notifications

342 Not all valarm properties make sense within the context of WS-Calendar. For example, audio alarms are
343 of little use in SOA communications. Alarms are communications that take no time, and may be delivered
344 on a schedule or as a notification of an event.

345 Valarm supports the same ATTACH attribute as does vtodo, and the same relationships. Valarm can be
346 included as one of many components within an iCalendar object. Valarm can schedule that a notification
347 be sent between vtodo events, perhaps reporting interim results.

348 Valarm also supports recurring activities. A long-running vtodo service could be started alongside a
349 recurring call-out to a 3rd service providing observation of the service's effects. For example, a Demand
350 Response vtodo could be launched accompanied by a recurring 5 minute request to read the meter from
351 another service.

352 5 Service Interactions

353 5.1 Use of Cal-WS

354 This OASIS Committee has worked closely with the CalConnect TC-XML committee, which publishes its
355 work through the IETF³. CalConnect is defining the core scheduling service interactions, i.e., scheduling
356 an event, determining availability, etc., and publishing them as Cal-WS.

357 It is the intent of this committee to interoperate fully with their specifications. As this draft is published,
358 CalConnect is releasing a draft for review as well.

359

360 The CalWs document can be found at

361

362 [http://www.calconnect.org/pubcomment/CD1004 Cal-WS Web Services API V0.3.pdf](http://www.calconnect.org/pubcomment/CD1004%20Cal-WS%20Web%20Services%20API%20V0.3.pdf)

363

364 Comments can be made at

365

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

367

³ <http://datatracker.ietf.org/wg/calsify/charter/>

368 6 WS-Calendar Structures

369 Structures are the larger assemblies of information built upon and out of the elements.

370 6.1 Partition Set

371 Partitioned sets are a collection of consecutive intervals. All members of a partition set share a common
372 granularity. Partition sets are not bonded to a particular time of occurrence.

WS-Calendar Partition Set Element	Specification (Normative)	Note (Non-Normative)
PartitionSetID	Identifier for this service interaction	This identifier merely needs to be unique in the context of a service interaction.
PartitionSetName	Optional text name for set	
PartitionSetGranularity	Optional	If present, all granularity on intervals is ignored, and MAY be forbidden.
PartitionIntervals	Array of Duration Intervals.	Sequence is determined by order of serialization.
PartitionSetReference	URI or XRI for interval payload. A Duration SHALL have either a Reference OR a Service	Not used if Service is present.
PartitionSetService	Container for XML component. A Duration SHALL have either a Reference OR a Service	Not used if Reference is present.

373

374 6.2 Period Set

375 A schedule set is a partition set that must being (or end) at a certain time. That time may or not be fully
376 qualified with a date, i.e., a Schedule Set may specify a set of activities that begin at 8:00 each morning.

WS-Calendar PeriodSet	Specification (Normative)	Note (Non-Normative)
PeriodSetID	Identifier for this service interaction	This identifier merely needs to be unique in the context of a service interaction.
PeriodSetName	Optional text name for set	

WS-Calendar PeriodSet	Specification (Normative)	Note (Non-Normative)
PeriodSetBegin	Time Schedule Set must begin	Legal formats can exclude date and time zone if necessary to support repeating or local schedules
PeriodSetEnd	Time Schedule Set must end	Legal formats can exclude date and time zone if necessary to support repeating or local schedules
PeriodSetGranularity	Optional	If present, all granularity on intervals is ignored, and MAY be forbidden.
PartitionedIntervals	Array of Intervals.	
PeriodSetReference	URI or XRI for interval payload. A Period SHALL have either a Reference OR a Service	
PeriodSetService	Container for XML component. A Period SHALL have either a Reference OR a Service	

377

7 Sequence of Operations

378

The committee has not yet developed this work.

379

It is the plan of the committee to investigate adding some interval and schedule attributes from iCalendar to the already existing work of BPELWS. The committee intends to begin this work while chapters 1-6 are out for preliminary public comment.

380

381

382

383 **8 Conformance**

384

385

386

387 *Note: The last numbered section in the specification must be the Conformance section. Conformance*
388 *Statements/Clauses go here.*

389 *All OASIS specifications require conformance*

390 Acknowledgements

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

393 **Participants:**

394 Brad Benson, Trane
395 Edward Cazalet, Individual
396 Toby Considine, University of North Carolina at Chapel Hill
397 William Cox, Individual
398 Craig Gemmill, Tridium, Inc.
399 Girish Ghatikar, Lawrence Berkeley National Laboratory
400 Gale Horst, Electric Power Research Institute (EPRI)
401 Gershon Janssen, Individual
402 Ed Koch, Akuacom Inc.
403 Benoit Lepeuple, LonMark International*
404 Carl Mattocks, CheckMi*
405 Robert Old, Siemens AG
406 Alexander Papaspyrou, Technische Universitat Dortmund
407 Jeremy Roberts, LonMark International*
408 David Thewlis, CalConnect

409

Background

410

411

Revision History

412

Revision	Date	Editor	Changes Made
1.0 WD 01	2010-03-11	Toby Considine	Initial document, largely derived from Charter
1.0 WD 02	2010-03-30	Toby Considine	Straw-man assertion of elements, components to push conversation
1.0 WD 03	2010-04-27	Toby Considine	Cleaned up Elements, added XPOINTER use, xs:duration elements
1.0 WD 04	2010-05-09	Toby Considine	Aligned Chapter 4 with the vAlarm and vToDo objects.
1.0 WD 05	2010-05-18	Toby Considine	Responded to comments, added references, made references to xCal more consistent,
1.0 WD 06	2010-05-10	Toby Considine	Responded to comments from CalConnect, mostly constancy of explanations
1.0 WD 07			
1.0 WD 02			
1.0 WD 09			

413

414