



# WS-Calendar Version 1.0

## Working Draft 09

15 August 2010

### Specification URIs:

#### This Version:

<http://docs.oasis-open.org/WS-Calendar/v1.0/wd09/WS-Calendar-1.0-spec-wd-09.pdf>  
<http://docs.oasis-open.org/WS-Calendar/v1.0/wd09/WS-Calendar-1.0-spec-wd-09.html>  
<http://docs.oasis-open.org/WS-Calendar/v1.0/wd09/WS-Calendar-1.0-spec-wd-09.doc>

#### Previous Version:

<http://docs.oasis-open.org/WS-Calendar/v1.0/wd08/WS-Calendar-1.0-spec-wd-08.pdf>  
<http://docs.oasis-open.org/WS-Calendar/v1.0/wd08/WS-Calendar-1.0-spec-wd-08.html>  
<http://docs.oasis-open.org/WS-Calendar/v1.0/wd08/WS-Calendar-1.0-spec-wd-08.doc>

#### 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 limited set of message components and interactions providing a common basis for specifying schedules and intervals to coordinate activities between services. The specification includes 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

46 These message components describe schedules and intervals future, present, or past (historical). The  
47 definition of the services performed to meet a schedule or interval depends on the market context in  
48 which that service exists. It is not in scope for this TC to define those markets or services.

49 Status:

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

53 Technical Committee members should send comments on this specification to the Technical Committee's  
54 email list. Others should send comments to the Technical Committee by using the "Send A Comment"  
55 button on the Technical Committee's web page at <http://www.oasis-open.org/committees/WS-Calendar/>.

56 For information on whether any patents have been disclosed that may be essential to implementing this  
57 specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights  
58 section of the Technical Committee web page ([http://www.oasis-open.org/committees/WS-](http://www.oasis-open.org/committees/WS-Calendar/ipr.php)  
59 [Calendar/ipr.php](http://www.oasis-open.org/committees/WS-Calendar/ipr.php)).

60 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/WS-Calendar/)  
61 [open.org/committees/WS-Calendar/](http://www.oasis-open.org/committees/WS-Calendar/).

---

62 **Notices**

63 Copyright © OASIS® 2010. All Rights Reserved.

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

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

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

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

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

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

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

102 The names "OASIS", [insert specific trademarked names and abbreviations here] are trademarks of  
103 OASIS, the owner and developer of this specification, and should be used only to refer to the organization  
104 and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications,  
105 while reserving the right to enforce its marks against misleading uses. Please see [http://www.oasis-](http://www.oasis-open.org/who/trademark.php)  
106 [open.org/who/trademark.php](http://www.oasis-open.org/who/trademark.php) for above guidance.

107

# 108 Table of Contents

109	1	Introduction.....	7
110	1.1	Terminology.....	8
111	1.2	Normative References.....	8
112	1.3	Non-Normative References.....	8
113	1.4	Naming Conventions.....	8
114	1.5	Architectural References.....	9
115	2	Overview of WS-Calendar.....	10
116	2.1	Approach taken by the WS-Calendar Technical Committee.....	10
117	2.2	Specification Deliverables.....	10
118	3	WS-Calendar Definitions.....	11
119	3.1	Scheduling Service Performance.....	11
120	3.2	Core Semantics xCal.....	11
121	3.2.1	Time.....	12
122	3.2.2	The iCalendar Components (VObjects).....	12
123	3.2.3	Intervals.....	12
124	3.2.4	Alarms.....	13
125	3.2.5	Related Components.....	13
126	3.3	Services and Service Characteristics.....	14
127	3.3.1	Attachments.....	14
128	3.3.2	Specifying Timely Performance.....	15
129	3.3.3	Combining Service and Performance.....	16
130	3.4	Time Stamps.....	17
131	4	Intervals, Partitions, Sequences, Processes, and Process Synchronization.....	18
132	4.1	Use of VTODO elements.....	18
133	4.1.1	Use of VTODO elements.....	18
134	4.1.2	Relationships between VTODO Objects.....	19
135	4.2	Intervals and Sequences.....	19
136	4.2.1	Intervals: the Basic Time Segment.....	20
137	4.2.2	Sequences: Putting things together.....	21
138	4.2.3	Related Components and Sequences.....	22
139	4.2.4	Partitions: Regular repeating sets.....	23
140	4.3	Notification and Synchronization.....	25
141	5	Calendar Service Interactions: Overview.....	27
142	5.1	Glossary.....	27
143	5.2	Issues not addressed by this specification.....	27
144	6	Service Interactions: Protocol Overview.....	28
145	7	Service Capabilities.....	29
146	8	Creating Calendar Resources.....	30
147	9	Retrieving Calendar Resources.....	31
148	10	Updating Calendar Resources.....	32
149	11	Deletion of Calendar Resources.....	33
150	12	Querying Calendar Resources.....	34
151	13	Conformance.....	35

152	Acknowledgements .....	36
153	Revision History .....	37
154		

---

155 **Tables**

156 **Index to Tables**

157 Table 1: Defining Time Segments for WS-Calendar..... 12  
158 Table 2: RelatedComponent elements in WS-Calendar ..... 13  
159 Table 3: Elements of a WS-Calendar Attachments ..... 14  
160 Table 4: Performance Characteristics..... 15  
161 Table 5: Aspects of Time Stamps ..... 17  
162 Table 6: VTODO elements in WS-Calendar ..... 18  
163 Table 7: Use of Inter-component Relationships in WS-Calendar ..... 19  
164 Table 8: Interval Data Elements..... 20

---

167 **Table of Examples**

168 Example 1: Use of an Attachment with inline XML artifact ..... 14  
169 Example 2: Use of an Attachment with external reference ..... 14  
170 Example 3: Performance Component ..... 16  
171 Example 5: Use of an Attachment with external reference and optional specified performance ..... 16  
172 Example 6: Vobject Relationship ..... 19  
173 Example 7: An Interval ..... 20  
174 Example 8: Simple sequence with three intervals ..... 21  
175 Example 9: Scheduled Sequence with shared performance definition ..... 22  
176 Example 10: Scheduled Partition with common contract..... 23  
177 Example 11: Scheduled Sequence with common contract, common duration..... 24

178  
179

---

# 180 1 Introduction

181 One of the most fundamental components of negotiating services is agreeing when something should  
182 occur, and in auditing when they did occur. Short running services traditionally have been handled as if  
183 they were instantaneous, and have handled scheduling through just-in-time requests. Longer running  
184 processes, including physical processes, may require significant lead times. When multiple long-running  
185 services participate in the same business process, it may be more important to negotiate a common  
186 completion time than a common start time. Pre-existing approaches that rely on direct control of such  
187 services by a central system increases integration costs and reduce interoperability as they require the  
188 controlling agent to know and manage multiple lead times.

189 Not all services are requested one time as needed. Processes may have multiple and periodic  
190 occurrences. An agent may need to request identical processes on multiple schedules. An agent may  
191 request services to coincide with or to avoid human interactions. Service performance be required on the  
192 first Tuesday of every month, or in weeks in which there is no payroll, to coordinate with existing business  
193 processes. Service performance requirements may vary by local time zone. A common schedule  
194 communication must support diverse requirements.

195 Physical processes are already being coordinated by web services. Building systems and industrial  
196 processes are operated using oBIX, BACnet/WS, LON-WS, OPC XML, and a number of proprietary  
197 specifications including TAC-WS, Gridlogix EnNet, and MODBUS.NET. In particular, if building systems  
198 coordinate with the schedules of the building's occupants, they can reduce energy use while improving  
199 performance.

200 An increasing number of specifications envision synchronization of processes through mechanisms  
201 including broadcast scheduling. Efforts to build an intelligent power grid (or smart grid) rely on  
202 coordinating processes in homes, offices, and industry with projected and actual power availability;  
203 mechanisms proposed include communicating different prices at different times. Several active OASIS  
204 Technical Committees require a common means to specify schedule and interval: Energy Interoperation  
205 (EITC) and Energy Market Information Exchange (EMIX). Emergency management coordinators wish to  
206 inform geographic regions of future events, such as a projected tornado touchdown, using EDXL. The  
207 open Building Information Exchange specification (OBIX) lacks a common schedule communications for  
208 interaction with enterprise activities. These and other efforts would benefit from a common cross-domain,  
209 cross specification standard for communicating schedule and interval.

210 For human interactions and human scheduling, the well-known iCalendar format is used to address these  
211 problems. Prior to WS-Calendar, there has been no comparable standard for web services. As an  
212 increasing number of physical processes become managed by web services, the lack of a similar  
213 standard for scheduling and coordination of services becomes critical.

214 The intent of the WS-Calendar technical committee was to adapt the existing specifications for  
215 calendaring and apply them to develop a standard for how schedule and event information is passed  
216 between and within services. The standard adopts the semantics and vocabulary of iCalendar for  
217 application to the completion of web service contracts. WS Calendar is built on work done and ongoing in  
218 The Calendaring and Scheduling Consortium (CalConnect), which works to increase interoperation  
219 between calendaring systems.

220 A calendar communication without a real world effect is of little interest. That real world effect is the result  
221 of a services execution context within a policy context<sup>1</sup>. Practitioners can use WS-Calendar to add  
222 communication of schedule and interval to the execution context of a service. Use of WS-Calendar will  
223 align the performance expectations between execution contexts in different domains. The Technical  
224 Committee intends for other specifications and standards to incorporate WS-Calendar, bringing a  
225 common scheduling context to diverse interactions in different domains.

226 Everything with the exception of all examples, all appendices, and the introduction is normative.

---

<sup>1</sup> See “*Reference Model for Service Oriented Architecture 1.0*” for definitions of all terms used herein to describe service interactions.

## 227 1.1 Terminology

228 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD  
229 NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described  
230 in **RFC2119**.

## 231 1.2 Normative References

- 232 **RFC2119** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
233 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 234 **SOA-RM** OASIS Standard, *Reference Model for Service Oriented Architecture 1.0*,  
235 October 2006.  
236 <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>
- 237 **RFC5545** B. Desruisseaux *Internet Calendaring and Scheduling Core Object*  
238 *Specification (iCalendar)*, <http://www.ietf.org/rfc/rfc5545.txt>, IETF RFC  
239 5545, September 2009.
- 240 **RFC5546** C. Daboo *iCalendar Transport-Independent Interoperability Protocol*  
241 *(iTIP)*, <http://www.ietf.org/rfc/rfc5546.txt>, IETF RFC 5546, January 1999.
- 242 **RFC2447** F. Dawson, S. Mansour, S. Silverberg, *iCalendar Message-Based*  
243 *Interoperability Protocol (iMIP)*, <http://www.ietf.org/rfc/rfc2447.txt>, IETF  
244 RFC 2447, December 2009.
- 245 **xCal** C. Daboo, M Douglas, S Lees *xCal: The XML format for iCalendar*,  
246 <http://tools.ietf.org/html/draft-daboo-et-al-icalendar-in-xml-03>, Internet-  
247 Draft, April 2010.
- 248 **Calendar Resource Schema** C. Joy, C. Daboo, M Douglas, *Schema for representing*  
249 *resources for calendaring and scheduling services*,  
250 <http://tools.ietf.org/html/draft-cal-resource-schema-00>, (Internet-Draft),  
251 April 2010.
- 252 **XPATH** A Berglund, S Boag, D Chamberlin, MF Fernández, M Kay, J Robie, J  
253 Siméon *XML Path Language (XPath) 2.0*, <http://www.w3.org/TR/xpath20/>  
254 January 2007.
- 255 **XLINK** S DeRose, E Maler, D Orchard, N Walsh *XML Linking Language (XLink)*  
256 *Version 1.1.*, <http://www.w3.org/TR/xlink11/> May 2010.
- 257 **XPOINTER** S DeRose, E Maler, R Daniel Jr. *XPointer xpointer Scheme*,  
258 <http://www.w3.org/TR/xptr-xpointer/> December 2002.
- 259 **XML Schema** PV Biron, A Malhotra, *XML Schema Part 2: Datatypes Second Edition*,  
260 <http://www.w3.org/TR/xmlschema-2/> October 2004.

## 261 1.3 Non-Normative References

- 262 **NIST Framework and Roadmap for Smart Grid Interoperability Standards**, Office of the  
263 National Coordinator for Smart Grid Interoperability, Release 1.0, NIST  
264 Special Publication 1108,  
265 [http://www.nist.gov/public\\_affairs/releases/upload/smartgrid\\_interoperabili](http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf)  
266 [ty\\_final.pdf](http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf) January 2010.
- 267 **NAESB Smart Grid Requirements** (*dunno what reference I need here*)

## 268 1.4 Naming Conventions

269 This specification follows some naming conventions for artifacts defined by the specification, as follows:

270 For the names of elements and the names of attributes within XSD files, the names follow the CamelCase  
271 convention, with all names starting with a lower case letter, eg

```
272 <element name="componentType" type="WS-Calendar:ComponentType"/>
```

273 For the names of types within XSD files, the names follow the CamelCase convention with all names  
274 starting with an upper case letter, e.g.,

```
275 <complexType name="ComponentService">
```

276 For the names of intents, the names follow the CamelCase convention, with all names starting with a  
277 lower case letter, EXCEPT for cases where the intent is to represent an established acronym, in which  
278 case the entire name follows the usage of the established acronym.

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

## 280 **1.5 Architectural References**

281 WS-Calendar assumes incorporation into services. Accordingly it assumes a certain amount of definitions  
282 of roles, names, and interaction patterns. This document relies heavily on roles and interactions as  
283 defined in the OASIS Standard *Reference Model for Service Oriented Architecture*.

---

## 284 2 Overview of WS-Calendar

285 A calendar communication without a real world effect<sup>2</sup> is of little interest. That real world effect is the result  
286 of a services execution context within a policy context. Practitioners can use WS-Calendar to add  
287 communication of schedule and interval to the execution context of a service. Use of WS-Calendar will  
288 align the performance expectations between execution contexts in different domains. The Technical  
289 Committee intends for other specifications and standards to incorporate WS-Calendar, bringing a  
290 common scheduling context to diverse interactions in different domains

### 291 2.1 Approach taken by the WS-Calendar Technical Committee

292 The Committee based its work upon the iCalendar specification as updated in 2009 (IETF RFC 5545) and  
293 its the XML serialization xCal, currently (2010-07) on a standards track in the IETF. Both updates were to  
294 IETF specifications were developed by members of the Calendaring and Scheduling Consortium  
295 (CalConnect.org). This work provides the vocabulary for use in this specification.

296 The committee solicited requirements from a range of interests, notably the NIST Smart Grid Roadmap  
297 and the requirements if the Smart Grid Interoperability Panel (SGIP) as developed by the North American  
298 Energy Standards Board (NAESB). Others submitting requirements included members of the oBIX  
299 technical committee and representative of the FIX Protocol Association

300 Based on these requirements, the technical committee developed the semantic elements in sections  
301 three and four.

302 In a parallel effort, the CalConnect TC-XML committee developed a number of schedule and calendar-  
303 related services. CalConnect drew on its experience in interoperability between enterprise calendaring  
304 systems as well as interactions with web-based calendars and personal digital assistants (PDAs). These  
305 services, which CalConnect refers to as CalWS, provide the basic interactions for querying, creating,  
306 updating, and deleting calendar events that are common to all calendars and schedules. CalConnect  
307 donated CalWS to WS-Calendar to make up the service interactions in section 5.

### 308 2.2 Specification Deliverables

309 The specification consists of a standard schema and semantics for schedule and interval information. The  
310 specification also includes standard service calls for calendar inquiries, event scheduling, event updating,  
311 and event cancelation. Finally, the specification includes rules for delivering a sequence of operations,  
312 i.e., a representation of several services that are scheduled as a single event.

313 The standard also includes guidance for including geo-location within an event.

---

<sup>2</sup> This paragraph includes a number of terms of art used in service oriented architecture (SOA). In all cases, the terms are as defined in the *Reference Model for Service Oriented Architecture*, found in the normative references.

---

## 314 **3 WS-Calendar Definitions**

315 WS-Calendar Elements are semantic elements derived from the xCal specification. These elements are  
316 smaller than a full schedule interaction, and describe the intervals, durations, and time-related events that  
317 are relevant to service interactions. In effect, the Elements are used to build a precise vocabulary of time,  
318 duration, sequencing, and schedule.

319 The lexicon of Elements is also used to decorate and elaborate the simpler specification of xCal to make  
320 explicit the performance expectations within a scheduled event. xCal to standardize data and interval  
321 outside of scheduling interactions.

322 WS-Calendar elements elaborate the objects defined in iCalendar, to make interaction requirements  
323 explicit. For example, in human schedule interactions, different organizations have their own  
324 expectations. Meetings may start on the hour or within 5 minutes of the hour. As agents scheduled in  
325 those organizations, people learn the expected precision. In WS-Calendar, that precision must be explicit  
326 to prevent interoperability problems.

### 327 **3.1 Scheduling Service Performance**

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

331 As defined in the OASIS Reference Model for Service Oriented Architecture 1.0<sup>3</sup>, service requests access  
332 the capability of a remote system.

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

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

341 A request for remote service performance is a request for specific real world effects. Consider two service  
342 providers that offer the same service. One must start planning an hour or more in advance. The second  
343 may be able to achieve the service in five minutes. The service start time is the time when that service  
344 becomes available. If we do not distinguish these circumstances, then the customer would receive quite  
345 different quite different services with no distinctions in the service contract.

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

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

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

### 354 **3.2 Core Semantics xCal**

355 The iCalendar data format [RFC5545] is a widely deployed interchange format for calendaring and  
356 scheduling data. The xCal specification (in process) standardizes the XML representation of iCalendar

---

<sup>3</sup> See normative references in section 1.2

357 information. WS-Calendar relies on xCal standards and data representation to develop its semantic  
358 components.

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

### 360 3.2.1 Time

361 Time is an ISO 8601 compliant time string with the optional accompaniment of a duration interval to  
362 define times of less than 1 second. Examples of the from the ISO 8601 standard include:

```
363 Year:  
364     YYYY (eg 1997)  
365 Year and month:  
366     YYYY-MM (eg 1997-07)  
367 Complete date:  
368     YYYY-MM-DD (eg 1997-07-16)  
369 Complete date plus hours and minutes:  
370     YYYY-MM-DDThh:mmTZD (eg 1997-07-16T19:20+01:00)  
371 Complete date plus hours, minutes and seconds:  
372     YYYY-MM-DDThh:mm:ssTZD (eg 1997-07-16T19:20:30+01:00)  
373 Complete date plus hours, minutes, seconds and a decimal fraction of a  
374 second  
375     YYYY-MM-DDThh:mm:ss.sTZD (eg 1997-07-16T19:20:30.45+01:00)
```

376 Normative information on ISO 8601 is referenced in section 1.2.

### 377 3.2.2 The iCalendar Components (VObjects)

378 iCalendar and xCal have a number of long defined component objects that comprise the payload inside of  
379 an iCalendar message. These include the VTODO, the VALARM, the VEVENT. These element names  
380 begin with “V” for historic reasons. The definitions and use of each of the VObjects is described in RFC  
381 5545.

382 Because of its flexibility, the VTODO object is the basis for WS-Calendar objects for service performance.  
383 Because WS-Calendar services support all traditional iCalendar-based interactions (CalDAV, et al.) all  
384 VObjects SHALL be supported.

### 385 3.2.3 Intervals

386 Time Segments, i.e., increments of continuous passage of time, are a critical component of service  
387 alignment using WS-Calendar. There are many overloaded uses of terms about time, and within a  
388 particular time segment, there may be many of them. Within this document, we use the term Time  
389 Segments to encompass all the terms in Table 1, below.

390 The base data type for time segments is the Interval. The Interval is a time segment defined by the  
391 Duration element as defined in xCal. The xCAL duration is a data type based upon the string  
392 representation in the iCalendar duration. The Committee listened to arguments that we should redefine  
393 the use and meaning of Duration. Whatever their merit, the iCalendar Duration has a pre-existing  
394 meaning of the length of time of scheduled within an event. In this section, the Duration is enumerated as  
395 one of several time segments.

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

Time Segment	Definition
<b>Duration</b>	Well-known element from iCalendar and xCal, Duration is the length of a meeting scheduled using iCalendar or any of its derivatives. The xCal duration is a data type using the string representation defined in the iCalendar duration. The Duration is the sole descriptive element of the VTODO object that is mandatory in the Interval
<b>Interval</b>	The Interval is a single duration supported by the full information set of the VTODO object as defined in iCalendar (RFC 5545) and refined in xCal. A WS-Calendar interval must include a Duration.

Time Segment	Definition
<b>Sequence</b>	A Sequence is a set of Intervals with defined temporal relationships. Sequences may have gaps between Intervals, or even simultaneous activities. A sequence is re-locatable, i.e., it does not have a specific date and time. A Sequence may consist of a single interval.
<b>Scheduled Sequence</b>	A Scheduled Sequence is a Sequence that is anchored by a specific date and time, that is, it is a Sequence with a start date and time. Specific performance of a Sequence against a service contract always occurs in a Scheduled Sequence.
<b>Partition</b>	A Partition is a set of consecutive intervals. A Partition includes the trivial case of a single Interval. A Partition is used to define a single service or behavior which varies over time. Examples include energy prices over time and or energy usage over time. A Partition is re-locatable, i.e., it does not have a specific date and time.
<b>Scheduled Partition</b>	A Scheduled Partition is a Partition that is anchored by a specific date and time, that is, it is a Partition with a start date and time. The Performance of a Partition against an executed service contract always occurs in a Scheduled Partition.

### 397 3.2.4 Alarms

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

### 401 3.2.5 Related Components

402 WS-Calendar introduces a new iCalendar component, the RelatedComponent. A RelatedComponent is  
403 essentially a VObject with no schedule or interval elements. WS-Calendar uses RelatedComponents to  
404 apply service information to Sequences and Partitions. The use of Related Components is described in  
405 *Section 4: Intervals, Partitions, Sequences, Processes, and Process Synchronization*.

406 *Table 2: RelatedComponent elements in WS-Calendar*

Elements	Use	Discussion
<b>Dtstamp</b>	Mandatory	
<b>Uid</b>	Mandatory	Used to enable unambiguous referencing of each VTODO object
<b>Class</b>	Optional	
<b>Summary</b>	Optional	Text describing the Association
<b>Attach</b>	Mandatory, Multipleoccurs	Contains XML Artifact defining performance or xPointer to artifact defining performance. If repeated, can refer to multiple artifacts
<b>Related</b>	Mandatory	A RelatedComponent must have a relationship with at least one other component. The only relationship defined for the RelatedComponent is the IsParent.

407

408 **3.3 Services and Service Characteristics**

409 While iCalendar expresses time and intervals, WS-Calendar further associates those intervals with  
 410 specific services and service characteristics. WS-Calendar uses the ATTACH element that is part of  
 411 each of the iCalendar components to specify services and performance characteristics.

412 In iCalendar, each component as an ATTACH element to carry unstructured information elaborating the  
 413 event or alarm communication. Attachments in iCalendar can also be in the form of URIs pointing outside  
 414 the iCalendar structure. WS-Calendar uses structured XML to communicate the substance of the request.  
 415 The details of that xml artifact are domain-specific and are outside the scope of this document.

416 **3.3.1 Attachments**

417 The XML artifact in the attachment may be in-line, i.e., contained within the ATTACH element of the  
 418 VTODO or VALARM object, or it may be found in another section of the same XML object, sharing the  
 419 same message as WS-Calendar element, or it may be discovered by external reference. Attachments,  
 420 then, are used to request “perform as described here”, or “perform as described below”, or “perform as  
 421 described elsewhere.”

422 *Table 3: Elements of a WS-Calendar Attachments*

Elements	Use	Discussion
<b>Artifact</b>	Optional. Any in-line XML. Must have at least one of Artifact or Reference	Defined per the business process associated with this interaction. WS-Calendar. This is not an object, it is merely a name for use in documentation
<b>Reference</b>	Optional. XPOINTER. Must have at least one of Artifact or Reference	Points to external XML, or XML located elsewhere in document
<b>Performance</b>	Optional	Specifies time-related performance characteristics.

423 When a WS-Calendar reference uses an external reference to specify a service, that reference is an  
 424 object of the type XPointer (see section 1.2)..XPointer is a general purpose URI and XML traversal  
 425 standard. This XPointer object is in the named data element “Reference.”

426 *Example 1: Use of an Attachment with inline XML artifact*

```

427 <VTODO>
428   <dtstamp></dtstamp>
429   <uid>aaaaaaaa1</uid>
430   <description>first contract</description>
431   <summary>defines contract to invoke Hello World Service</summary>
432   <duration>T00:15</duration>
433   <attach>
434     <process name="pns:HelloWorld>
435       <active>TRUE</active>
436       <service name="wns:HelloWorldService" port="HelloWorldPort"/>
437     </process>
438   </attach>
439 </VTODO>
  
```

440 *Example 2: Use of an Attachment with external reference*

```

441 <VTODO>
442   <dtstamp></dtstamp>
443   <uid>aaaaaaaa1</uid>
444   <description>first contract</description>
445   <summary>defines contract to described at reference</summary>
446   <duration>T00:15</duration>
  
```

```

447 <attach>
448 <reference>http://scheduled.ws-calendar-service.com/contract1</reference>
449 </attach>
450 </VTODO>

```

451

### 452 3.3.2 Specifying Timely Performance

453 Service coordination between systems requires precise communication about expectation for the  
454 timeliness of performance. These expectations can be set for each interval or for an entire sequence.  
455 This communication is through the performance component of the Attachment.

456 The Performance component refines the meaning of time-related service communication. All elements of  
457 the Performance object use the Duration element as defined in RFC 5545.

458 *Table 4: Performance Characteristics*

Performance Characteristic	Definition	Discussion
<b>StartBeforeTolerance</b>	A Duration enumerating how far before the requested start time the requested service may commence.	Indicates if a service that begins at 1:57 is compliant with a request to start at 2:00
<b>StartAfterTolerance</b>	A Duration enumerating how far after the requested start time the requested service may commence.	Indicates if a service that begins at 2:01 is compliant with a request to start at 2:00
<b>EndBeforeTolerance</b>	A Duration enumerating how far before scheduled end time may end.	Indicates if a service that ends at 1:57 is compliant with a request to end at 2:00
<b>EndAfterTolerance</b>	A Duration enumerating how far after the scheduled end time the requested service may commence.	Indicates if a service that ends at 2:01 is compliant with a request to end at 2:00
<b>DurationLongTolerance</b>	A Duration indicating by how much the performance duration may exceed the duration specified in the Interval . It may be 0.	Used when run time is more important than start and stop time. DurationLongTolerance SHALL NOT be used when Start and End Tolerances are both specified.
<b>DurationShortTolerance</b>	A Duration indicating by how much the performance duration may fall short of duration specified in the Interval . It may be 0.	Used when run time is more important than start and stop time. DurationShortTolerance SHALL NOT be used when Start and End Tolerances are both specified.
<b>Granularity</b>	A Duration enumerating the smallest unit of time measured or tracked	Whatever the time tolerance above, there is some minimum time that is considered insignificant. A Granularity of 1 second defines the tracking and reporting requirements for a service.

459 Performance is part of the core WS-Calendar service definition. Similar products or services, identical  
460 except for different Performance characteristics may appear in different markets. Performance  
461 characteristics influence the price offered and the service selected.

462 Note that Performance object does not indicate time, but only duration. A performance object associated  
463 with an unscheduled Interval does not change when that Interval is scheduled.

464 The Performance object is an optional component of each WS-Calendar attachment.

465 *Example 3: Performance Component*

```
466 <performance>  
467   <startbefore>T00:10</startbefore>  
468   <startafter>T00:00</startafter>  
469   <durationlong>T00:00</durationlong>  
470   <durationshort>T00:00</durationshort>  
471 </performance>
```

472 In the example, the service can start as much as 10 minutes earlier than the scheduled time, and must  
473 start no later than the scheduled time. Whenever the service starts, it must be performed for exactly the  
474 duration indicated.

475 Generally, the implementer should refrain from expressing unnecessary or redundant performance  
476 characteristics.

### 477 3.3.3 Combining Service and Performance

478 Services, references and performance each appear in the ATTACH element of the iCalendar  
479 components.

480 *Example 4: Use of an Attachment with inline XML artifact and optional specified Performance*

```
481 <VTODO>  
482   <dtstamp></dtstamp>  
483   <uid>aaaaaaaa1</uid>  
484   <description>first contract</description>  
485   <summary> defines contract to invoke Hello World Service as early as 10  
486   minutes before scheduled time, and no later than scheduled time</summary>  
487   <duration>T00:15</duration>  
488   <attach>  
489     <process name="pns:HelloWorld">  
490       <active>TRUE</active>  
491       <service name="wns:HelloWorldService" port="HelloWorldPort"/>  
492     </process>  
493     <performance>  
494       <startbefore>T00:10</startbefore>  
495       <startafter>T00:00</startafter>  
496       <durationlong>T00:00</durationlong>  
497       <durationshort>T00:00</durationshort>  
498     </performance>  
499   </attach>  
500 </VTODO>
```

501 *Example 5: Use of an Attachment with external reference and optional specified performance*

```
502 <VTODO>  
503   <dtstamp></dtstamp>  
504   <uid>aaaaaaaa1</uid>  
505   <description>first contract</description>  
506   <summary>defines first behavior to perform in contract with a precisions  
507   required of 1 second</summary>  
508   <duration>T00:15</duration>  
509   <attach>  
510     <reference>http://scheduled.ws-calendar-service.com/contract1</reference>  
511     <performance>  
512       <startbefore>T00:10</startbefore>
```

```

513     <startafter>T00:00</startafter>
514     <durationlong>T00:00</durationlong>
515     <durationshort>T00:00</durationshort>
516     </performance>
517 </attach>
518 </VTODO>

```

519

### 520 3.4 Time Stamps

521 Time stamps are used everywhere in inter-domain service performance analysis and have particular use  
522 in smart grids to support event forensics. Time stamps are often assembled and collated from events  
523 across multiple time zones.

524 Different systems may track time and therefore record events with different levels of Tolerance. It is not  
525 unusual for a time stamp from a domain with a low Tolerance to appear to have occurred after events  
526 from a domain with high-Tolerance time-stamps that it caused. A fully qualified time-stamp includes the  
527 granularity measure.

528 *Table 5: Aspects of Time Stamps*

WS-Calendar Time Stamp Element	Definition (Normative)	Note (Non-Normative)
<b>TimeStamp</b>	WS-Calendar:time A fully qualified date and time of event	May include two objects as defined above.
<b>Precision</b>	A Duration defining the accuracy of the time stamp.	Identifies whether one hour interval is indeed one hour or plus or minus some number of seconds and minutes
<b>TimeStampRealm</b>	The realm identifies where the time stamp originated  <b>Is this a GUID or some other sort of data type?</b>	The term Realm originates in ISO 61850. A realm is a set of points which are reasonably synchronized. 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.
<b>LeapSecondsKnown</b>	Xs:boolean	<b>Need a definition here</b>
<b>ClockFailure</b>	xs:boolean	Indicates that the time source of the sending device is unreliable
<b>ClockNotSynchronized</b>	xs:boolean	indicates that the time source of the sending device is not synchronized with the external UTC time
<b>Accuracy</b>	A Duration defining the accuracy of the clock used in the TimeStampRealm.	represents the time accuracy class of the time source of the sending device relative to the external UTC time.
<b>Attachment</b>	As defined in section <b>Error! Reference source not found.</b>	Contains either local description of service or reference to xml document describing service

---

## 529 4 Intervals, Partitions, Sequences, Processes, and 530 Process Synchronization

531 WS-Calendar derives objects for communicating intervals and for synchronizing time from the  
532 corresponding iCalendar objects. Within an iCalendar message, there is a larger document envelope  
533 containing transaction and synchronization information. The use of those fields is discussed below in  
534 section 5 under Calendar Service Interactions.

535 In iCalendar (and therefore xCalendar), one of the top-level objects is the Components section which can  
536 contain one or many iCalendar components, the so-called VObjects. Traditional calendar sharing has  
537 tended to use only one or two components, say a single meeting (VEVENT) or perhaps a task (VTODO)  
538 and a request to warn the recipient of the impending due date in advance (VALARM).

539 Within WS-Calendar, these components can be strung together to create packages of service interactions  
540 and market operations. As services are advertised, they may not yet have specific performance time  
541 scheduled. For this reason, only the Duration is required.

542 A Start time plus a duration fully implies an end time; it is not necessary to specify a start, duration, and  
543 end. Specifying all three could allow to a message that is internally inconsistent. Allowing options leads to  
544 complexity. As the duration is the required element, one of the times is redundant. WS-Calendar specifies  
545 that only the Start Time and Duration are considered. While an end time is a legal component of a  
546 VTODO object, WS-Calendar services ignore it.

### 547 4.1 Use of VTODO elements

548 The simplest segment of time is a single interval. Intervals are derived from a single VTODO object. For  
549 ease of reference, the VTODO object is described below. In all cases, implementers SHALL refer to RFC  
550 5545 and the xCal specifications for the normative description and definitions.

#### 551 4.1.1 Use of VTODO elements

552 All elements of the VTODO component are legal in WS-Calendar, certain elements are more critical when  
553 invoking services. These elements and their definitions within WS-Calendar are listed in *Table 6: VTODO  
554 elements in WS-Calendar*. Elements marked mandatory SHALL be in every use of VTODO in WS-  
555 Calendar.

556 *Table 6: VTODO elements in WS-Calendar*

Elements	Use	Discussion
<b>Dtstamp</b>	Mandatory	
<b>Uid</b>	Mandatory	Used to enable unambiguous referencing of each VTODO object
<b>Class</b>	Optional	
<b>Duration</b>	Mandatory	
<b>dtStart</b>	Optional	Scheduled start date and time for interval
<b>dtEnd</b>	Ignored	Legal only when Duration is not specified. As WS-Calendar required Duration, dtEnd is ignored.
<b>Attach</b>	Mandatory, Multipleoccurs	Contains XML Artifact defining performance or xPointer to artifact defining performance. If repeated, can refer to multiple artifacts

<b>Related</b>	Optional compound element	Defines relations to other components. May be mandatory in derived specifications, esp. if component order is important.
----------------	---------------------------	--

## 557 4.1.2 Relationships between VTODO Objects

558 Many service communications involve more than one time segment. These segments may be  
559 consecutive, as in an Interval, or they may have a more complex temporal relation, as in Sequences. The  
560 rules for parsing XML do not mandate preservation of order within a sub-set. This means that we cannot  
561 assume that order is preserved when parsing a set of iCalendar Components. For Sequences, mere  
562 order is not enough—this leads to the relationships.

563 In iCalendar, each Component (a VObject in the Components Section) may have an array of relationships  
564 to the other Components. In WS-Calendar, each relationship may also have an optional Gap expressed  
565 as an iCalendar duration.

566 *Example 6: Vobject Relationship*

```
567 <relationship>
568   <uid>aaaaaaaa1</uid><reltype>FS</reltype><gap>T00:10</gap>
569 </relationship>
```

570 The Gap refines the relationship, in this cases, adding 10 minutes to the FS relationship. In the absence  
571 of a Gap, the Gap is assumed to be 0.

572 *Table 7: Use of Inter-component Relationships in WS-Calendar*

Relationship	Memnonic	Definition
<b>FS</b>	Finish-Start	As soon as the related Component finishes, this interval begins. This form is normally used in Intervals, i.e., consecutive Components. If there is a gap, than it indicates the period between the finish of the referenced Component and the start of the referring Component.
<b>FF</b>	Finish-Finish	Used without gap when to components must finish at the same time. If there is a gap, it indicates that the referring component will finish execution a duration after the referred-to component.
<b>SF</b>	Start-Finish	This component must Finish before the related component starts. This relationship would be used by the preceding Component to refer to the next Component in a sequential Interval.
<b>SS</b>	Start-Start	These Components must start at the same time. If there is a gap, it refers to how long the referring Component has to start after the Component it references.

573 In an Interval, each component would have a FS relationship to the prior Component with no gap. In a  
574 Sequence, the relationships can be more complex.

## 575 4.2 Intervals and Sequences

576 An Interval specifies a single segment of time specified using a VTODO object. Sequences consist of one  
577 or more intervals. A Partition is a special case of a Sequence in which the Durations are identical and  
578 Intervals occur consecutively with no time in between.

579 Each VTODO in a Sequence may have a relationship with the other objects. These relationships detmine  
580 the temporal relationship between the Intervals. In Partitions, these relationships are limited to ordering  
581 the Intervals.

582 XML does not specify that sequence is maintained during XML processing. For this reason, even in the  
583 simple case of a Partition, each VTODO has a relationship its precedent and succedent. While we have

584 included examples below, implementers should refer to RFC 5545 and the xCal specifications for the  
 585 normative descriptions and definitions.

## 586 4.2.1 Intervals: the Basic Time Segment

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

589 *Table 8: Interval Data Elements*

Elements	Use	Discussion
<b>Dtstamp</b>	Mandatory	
<b>Uid</b>	Mandatory	Used to enable unambiguous referencing of each VTODO object
<b>Class</b>	Optional	
<b>Duration</b>	Mandatory	
<b>Performance</b>	Optional	Defines performance characteristics of Interval. Ignored by traditional iCalendar processors.
<b>Attach</b>	Optional, Multipleoccurs	Contains XML Artifact defining service as specified in the Attachment section. Required unless a relationship is defined to a RelatedComponent that defines the service.
<b>Related</b>	Optional compound element	Defines relations to other components. Used to define temporal relationships. Can be used as external reference to service definition.

590 The example below shows the components section of a WS-Calendar event containing two consecutive  
 591 15 minute time segments. They are listed in order. As XML parsing is not guaranteed to result in the  
 592 same order, each has a UID and a relationship. defining the order that each will occur. No start date is  
 593 specified in these time segments. As they are linked to each other, they describe a service of 30 minutes  
 594 total made up of two consecutive segments.

595 *Example 7: An Interval*

```

596 <VTODO>
597   <dtstamp></dtstamp>
598   <uid>aaaaaaa1</uid>
599   <description>first contract</description>
600   <summary>defines first behavior to perform in contract with a precisions
601   required of 1 second</summary>
602   <duration>T00:15</duration>
603   <attach>
604     <reference>http://scheduled.ws-calendar-service.com/contract1</reference>
605     <performance>
606       <endbefore>T00:00</endbefore>
607       <endafter>T00:00</endafter>
608       <durationlong>T00:00</durationlong>
609       <durationshort>T00:00</durationshort>
610     </performance>
611   </attach>
612 </VTODO>
  
```

613 Note that no start time is specified, and no relationship. Relationships are mandatory when an interval is  
 614 incorporated into a Sequence.

## 615 4.2.2 Sequences: Putting things together

616 Sequences are collections of related Intervals. The relationships define the time relationships between  
617 the Intervals. Sequences become Scheduled Sequences when the first Interval is assigned a starting  
618 time.

619 *Example 8: Simple sequence with three intervals*

```
620 <components>
621 <VTODO>
622   <dtstamp></dtstamp>
623   <uid>aaaaaaaa1</uid>
624   <description>first contract</description>
625   <priority>high</priority>
626   <summary>defines first behavior to perform in contract with a precisions
627   required of 1 second</summary>
628   <duration>T00:15</duration>
629   <attach>
630     <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
631     <performance>
632       <endbefore>T00:00</endbefore>
633       <endafter>T00:00</endafter>
634       <durationlong>T00:00</durationlong>
635       <durationshort>T00:00</durationshort>
636     </performance>
637   </attach>
638   <related-to>
639     <relationship>
640       <uid>aaaaaaaa2</uid><reltype>SF</reltype>
641     </relationship>
642   </related-to>
643 </VTODO>
644 <VTODO>
645   <dtstamp></dtstamp>
646   <uid>aaaaaaaa2</uid>
647   <description>second interval</description>
648   <priority>high</priority>
649   <summary>defines second behavior to perform in contract with a precision
650   required of 1 second</summary>
651   <duration>T00:15</duration>
652   <attach>
653     <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
654     <performance>
655       <endbefore>T00:00</endbefore>
656       <endafter>T00:00</endafter>
657       <durationlong>T00:00</durationlong>
658       <durationshort>T00:00</durationshort>
659     </performance>
660   </attach>
661   <related-to>
662     <relationship>
663       <uid>aaaaaaaa1</uid><reltype>FS</reltype>
664       <uid>aaaaaaaa3</uid><reltype>SF+10m</reltype>
665     </relationship>
666   </related-to>
667 </VTODO>
668 <VTODO>
669   <dtstamp></dtstamp>
670   <uid>aaaaaaaa3</uid>
671   <description>second interval</description>
672   <priority>high</priority>
673   <summary>defines second behavior to perform in contract with a precision
674   required of 1 second</summary>
675   <duration>T00.30</duration>
```

```

676 <attach>
677 <reference>http://scheduled.ws-calendar-service.com/contract3</reference>
678 <performance>
679 <endbefore>T00:00</endbefore>
680 <endafter>T00:00</endafter>
681 <durationlong>T00:00</durationlong>
682 <durationshort>T00:00</durationshort>
683 </performance>
684 </attach>
685 <related-to>
686 <relationship>
687 <uid>aaaaaaa2</uid><reltype>FS+T00.10</reltype>
688 </relationship>
689 </related-to>
690 </VTODO>
691 <components>

```

692 The first interval of 15 minutes is followed immediately by the second interval of 15 minutes. There is a 10  
693 minute interval between the completion of the second interval and the beginning of the third.

694 In the example above, each Interval has its own performance characteristics.

### 695 4.2.3 Related Components and Sequences

696 The RelatedComponent can be used to define common service requirements for an entire sequence. If a  
697 RelatedComponent has a parent relationship with the first Interval in a sequence, then the  
698 RelatedComponent's Attachment defines service attributes by all Intervals in the Sequence.

699 This performance component defines that service may start up to 10 minutes early, may not start late,  
700 and an accuracy of two seconds is expected in all timing and reporting.

701 *Example 9: Scheduled Sequence with shared performance definition*

```

702 <components>
703 <RelatedComponent>
704 <dtstamp></dtstamp>
705 <uid>aaaaaaa0</uid>
706 <summary>Defines performance requirements for entire sequence</summary>
707 <attach>
708 <performance>
709 <startbefore>T00:10</startbefore>
710 <startafter>T00:00</startafter>
711 <durationlong>T00:00</durationlong>
712 <durationshort>T00:00</durationshort>
713 <granularity>T00:00:02</dgranularity>
714 </performance>
715 </attach>
716 <relationship>
717 <uid>aaaaaaa1</uid><reltype>Parent</reltype>
718 </relationship>
719 </RelatedComponent>
720 <VTODO>
721 <dtstamp></dtstamp>
722 <uid>aaaaaaa1</uid>
723 <description>first contract</description>
724 <priority>high</priority>
725 <summary>defines first behavior to perform in contract with a precisions
726 required of 1 second</summary>
727 <dtstart>20100524T220000Z</dtstart>
728 <duration>T00:15</duration>
729 <attach>http://scheduled.ws-calendar-service.com/contract1</attach>
730 <related-to>
731 <relationship>
732 <uid>aaaaaaa2</uid><reltype>SF</reltype>
733 </relationship>

```

```

734     </related-to>
735 </VTODO>
736 <VTODO>
737     <dtstamp></dtstamp>
738     <uid>aaaaaaa2</uid>
739     <description>second interval</description>
740     <priority>high</priority>
741     <summary>defines second behavior to perform in contract with a precision
742     required of 1 second</summary>
743     <duration>T00:15</duration>
744     <attach>http://scheduled.ws-calendar-service.com/contract2</attach>
745     <related-to>
746         <relationship>
747             <uid>aaaaaaa1</uid><reltype>FS</reltype>
748             <uid>aaaaaaa3</uid><reltype>SF+10m</reltype>
749         </relationship>
750     </related-to>
751 </VTODO>
752 <VTODO>
753     <dtstamp></dtstamp>
754     <uid>aaaaaaa3</uid>
755     <description>second interval</description>
756     <priority>high</priority>
757     <summary>defines second behavior to perform in contract with a precision
758     required of 1 second</summary>
759     <duration>30m</duration>
760     <attach>http://scheduled.ws-calendar-service.com/contract2</attach>
761     <related-to>
762         <relationship>
763             <uid>aaaaaaa2</uid><reltype>FS+T00:10</reltype>
764         </relationship>
765     </related-to>
766 </VTODO>
767 <components>

```

768 The shared performance component simplifies processing by establishing a common standard for all  
769 elements. Individual TODO elements need not be evaluated for performance.

770 A shared performance component is outside of the Sequence structure but within the components  
771 structure. If there is more than one Sequence in the components section, and if a Performance  
772 component is specified for the entire sequence, it specifies the performance characteristics for all  
773 Intervals in the components collection.

774 There SHALL be no more than one Performance component in the components collection. If there is a  
775 Performance component in the components collection, then any performance components within the  
776 individual intervals SHALL be ignored.

#### 777 4.2.4 Partitions: Regular repeating sets

778 Perhaps the most common Sequence is one in which all Intervals have an identical duration, and each  
779 Interval follows immediately upon the completion of its predecessor. This partitioned set of intervals  
780 defines a Partition.

781 A partition is also degenerate in that the intervals are assumed to be identical in most aspects. For this  
782 reason, Description, Summary, and Priority are ignored when processing Partitions.

783 *Example 10: Scheduled Partition with common contract*

```

784 <components>
785 <RelatedComponent>
786     <dtstamp></dtstamp>
787     <uid>aaaaaaa0</uid>
788     <summary>Defines performance requirements for entire sequence</summary>
789     <attach>
790         <reference>http://scheduled.ws-calendar-service.com/contract1</reference>

```

```

791     <performance>
792         <startbefore>T00:10</startbefore>
793         <startafter>T00:00</startafter>
794         <durationlong>T00:00</durationlong>
795         <durationshort>T00:00</durationshort>
796         <granularity>T00:00:02</dgranularity>
797     </performance>
798 </attach>
799     <relationship>
800         <uid>aaaaaaaa1</uid><reltype>Parent</reltype>
801     </relationship>
802 </RelatedComponent>
803 <VTODO>
804     <dtstamp></dtstamp>
805     <uid>aaaaaaaa1</uid>
806     <description>first contract</description>
807     <dtstart>20100524T220000Z</dtstart>
808     <duration>T00:15</duration>
809     <attach>http://scheduled.ws-calendar-service.com/contract1</attach>
810     <related-to>
811         <relationship>
812             <uid>aaaaaaaa2</uid><reltype>SF</reltype>
813         </relationship>
814     </related-to>
815 </VTODO>
816 <VTODO>
817     <dtstamp></dtstamp>
818     <uid>aaaaaaaa2</uid>
819     <duration>T00:15</duration>
820     <attach>http://scheduled.ws-calendar-service.com/contract2</attach>
821     <related-to>
822         <relationship>
823             <uid>aaaaaaaa1</uid><reltype>FS</reltype>
824             <uid>aaaaaaaa3</uid><reltype>SF+T00:10</reltype>
825         </relationship>
826     </related-to>
827 </VTODO>
828 <VTODO>
829     <dtstamp></dtstamp>
830     <uid>aaaaaaaa3</uid>
831     <duration>15m</duration>
832     <attach>http://scheduled.ws-calendar-service.com/contract2</attach>
833     <related-to>
834         <relationship>
835             <uid>aaaaaaaa2</uid><reltype>FS+T00:10</reltype>
836         </relationship>
837     </related-to>
838 </VTODO>
839 </components>

```

840 Notice that the common contract is part of the Performance object. In this case, the Partition may start up  
841 to one minute early, and may not begin late. For each interval, the performance component that the  
842 duration be absolute.

843 Note also that Priority, Description, and Summary do not appear in any Interval in the Partition.

844 Note as well that this is a Scheduled Partition, and only the first Interval has a start time.

845 The same Scheduled Partition could be created with identical Durations for each Interval by specifying  
846 the Duration in the RelatedComponent.

847 *Example 11: Scheduled Sequence with common contract, common duration*

```

848 <components>
849 <RelatedComponent>
850     <dtstamp></dtstamp>

```

```

851 <uid>aaaaaaa0</uid>
852 <summary>Defines performance requirements and duration for entire
853 sequence</summary>
854 <duration>T00:15</duration>
855 <attach>
856 <reference>http://scheduled.ws-calendar-service.com/contract1</reference>
857 <performance>
858 <startbefore>T00:10</startbefore>
859 <startafter>T00:00</startafter>
860 <durationlong>T00:00</durationlong>
861 <durationshort>T00:00</durationshort>
862 <granularity>T00:00:02</dgranularity>
863 </performance>
864 </attach>
865 <relationship>
866 <uid>aaaaaaa1</uid><reltype>Parent</reltype>
867 </relationship>
868 </RelatedComponent>
869 <VTODO>
870 <dtstamp></dtstamp>
871 <uid>aaaaaaa1</uid>
872 <description>first contract</description>
873 <dtstart>20100524T220000Z</dtstart>
874 <attach>http://scheduled.ws-calendar-service.com/contract1</attach>
875 <related-to>
876 <relationship>
877 <uid>aaaaaaa2</uid><reltype>SF</reltype>
878 </relationship>
879 </related-to>
880 </VTODO>
881 <VTODO>
882 <dtstamp></dtstamp>
883 <uid>aaaaaaa2</uid>
884 <attach>http://scheduled.ws-calendar-service.com/contract2</attach>
885 <related-to>
886 <relationship>
887 <uid>aaaaaaa1</uid><reltype>FS</reltype>
888 <uid>aaaaaaa3</uid><reltype>SF+T00:10</reltype>
889 </relationship>
890 </related-to>
891 </VTODO>
892 <VTODO>
893 <dtstamp></dtstamp>
894 <uid>aaaaaaa3</uid>
895 <attach>http://scheduled.ws-calendar-service.com/contract2</attach>
896 <related-to>
897 <relationship>
898 <uid>aaaaaaa2</uid><reltype>FS+T00:10</reltype>
899 </relationship>
900 </related-to>
901 </VTODO>
902 <components>

```

903 Each Interval in this Sequence shares the 15 minute duration defined in the RelatedComponent as well  
904 as common Performance characteristics

### 905 4.3 Notification and Synchronization

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

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

913 *A "VALARM" calendar component is a grouping of component properties that is a reminder or alarm for*  
914 *an event or a to-do. For example, it may be used to define a reminder for a pending event or an overdue*  
915 *to-do. The "VALARM" calendar component MUST include the "ACTION" and "TRIGGER" properties. .*  
916 *.The "ACTION" property is used within the "VALARM" calendar component to specify the type of action*  
917 *invoked when the alarm is triggered. The "VALARM" properties provide enough information for a specific*  
918 *action to be invoked<sup>4</sup>.*

919 In WS-Calendar, an alarm is a VALARM object within a VTODO object, Its actions are XPOINTER  
920 references to the service or event that is triggered,

921 Valarm also supports recurring activities. A long-running VTODO service could be started alongside a  
922 recurring call-out to a 3<sup>rd</sup> service providing observation of the service's effects. For example, a Demand  
923 Response VTODO could be launched accompanied by a recurring 5 minute request to read the meter  
924 from another service.

---

<sup>4</sup> From the RFC 5545 – see normative references

---

## 925 **5 Calendar Service Interactions: Overview**

926 This OASIS Committee has worked closely with the CalConnect TC-XML committee, which publishes its  
927 work through the IETF<sup>5</sup>. CalConnect is defining the core scheduling service interactions, i.e., scheduling  
928 an event, determining availability, etc., and publishing them as Cal-WS.

### 929 **5.1 Glossary**

930	3.1 Hrefs	.....	8
931	3.2 Calendar Object Resource		8
932	3.3 Calendar Collection	.....	8
933	3.4 Scheduling Calendar Collection	.....	
934			

### 935 **5.2 Issues not addressed by this specification**

936	2.1 Access Control	...	7
937	2.2 Creating Collections	.....	7
938	2.3 Provisioning	.....	7
939	2.4 Discovery	.....	7
940	2.5 Retrieving collections	.....	7
941			

---

<sup>5</sup> <http://datatracker.ietf.org/wg/calsify/charter/>

---

## 942 **6 Service Interactions: Protocol Overview**

943	4 Overview of the protocol	.....9
944	4.1 Error conditions	.. 9
945	4.2 HTTP Methods	... 9
946	4.3 Operations	..... 9
947	4.4 Calendar Object Resources	.....9
948	4.5 Timezone information	.....10

---

949 **7 Service Capabilities**

950 Different Calendars and schedule systems have different capabilities. The more sophisticated system  
951 may have to simplify interactions to interact with the less capable system.

952 5 Capabilities ..... 11

953 5.1 Request parameters .....11

954 5.2 Responses: .....11

955 5.3 Example: ..... 11

956

---

957 **8 Creating Calendar Resources**

958 6 Creating Calendar Object Resources ...12  
959 6.1 Request parameters .....12  
960 6.2 Responses: .....12  
961 6.3 Preconditions for PUT, COPY, and MOVE ..12  
962 6.4 Example: ..... 13  
963

---

964 **9 Retrieving Calendar Resources**

965 7 Retrieving resources 14  
966 7.1 Request parameters .....14  
967 7.2 Responses: .....14  
968

---

969 **10 Updating Calendar Resources**

970 8 Updating resources .. 15

971 8.1 Responses: .....15

972

---

973 **11 Deletion of Calendar Resources**

974 9 Deletion of resources ..... 16

975 9.1 Responses: .....16

976

---

977 **12 Querying Calendar Resources**

978 10 Querying calendar resources 1

979

---

980 **13 Conformance**

981 WS-Calendar Intervals SHALL have a Duration. Intervals MAY have a StartTime. Intervals SHALL NOT  
982 include an END time. If a non-compliant Interval is received with an END time, it may be ignored.

983 A performance component SHALL not include Start, Stop, and Duration elements. Two out of the three  
984 elements is acceptable, but not four.

985 In Partitions, the Description, Summary and Priority of each Interval SHALL be excluded.

986

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

989 *All OASIS specifications require conformance*

---

## 990 Acknowledgements

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

993 **Participants:**

994 Brad Benson, Trane

995 Edward Cazalet, Individual

996 Toby Considine, University of North Carolina at Chapel Hill

997 William Cox, Individual

998 Craig Gemmill, Tridium, Inc.

999 Girish Ghatikar, Lawrence Berkeley National Laboratory

1000 Gale Horst, Electric Power Research Institute (EPRI)

1001 Gershon Janssen, Individual

1002 Ed Koch, Akuacom Inc.

1003 Benoit Lepeuple, LonMark International\*

1004 Carl Mattocks, CheckMi\*

1005 Robert Old, Siemens AG

1006 Alexander Papaspyrou, Technische Universitat Dortmund

1007 Jeremy Roberts, LonMark International\*

1008 David Thewlis, CalConnect

1009

1010 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-  
1011 Calendar Technical Committee, bridging to developing IETF standards and contributing the Services  
1012 definitions that make up Section 5, Calendar Service Interactions. The Technical Committee gratefully  
1013 acknowledges their assistance and cooperation as well.

1014

1015

## Revision History

1016

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	2010-07-28	Toby Considine	Incorporated input from informal public review, esp. SGIP PAP04. Firmed up relationships between scheduled objects
1.0 WD 08	2010-08-07	Toby Considine	Aligned with Interval / Partition / Sequence language. Reduced performance characteristics to before / after durations.
1.0 WD 09	2010-08-15	Toby Considine	Formalized Attachment section and rolled Performance into the Attachment. Created RelatedComponent object. Added CalWS Outline to specification. Removed SOOP section

1017

1018