



# WS-Calendar Version 1.0

## Working Draft 11

11 September 2010

### Specification URIs:

#### This Version:

- <http://docs.oasis-open.org/WS-Calendar/v1.0/wd11/WS-Calendar-1.0-spec-wd-11.pdf>
- <http://docs.oasis-open.org/WS-Calendar/v1.0/wd11/WS-Calendar-1.0-spec-wd-11.html>
- <http://docs.oasis-open.org/WS-Calendar/v1.0/wd11/WS-Calendar-1.0-spec-wd-11.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 [RFC5545], ICalendar
- IETF RFC5546, ICalendar Transport
- IETF RFC2447, 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

# Table of Contents

109	1	Introduction.....	7
110	1.1	Terminology.....	7
111	1.2	Normative References.....	8
112	1.3	Non-Normative References.....	8
113	1.4	Naming Conventions.....	9
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	Overview of This Document.....	10
118	3	WS-Calendar Semantics.....	11
119	3.1	Scheduling Service Performance.....	11
120	3.2	Core Semantics derived from [XCAL].....	11
121	3.2.1	Time.....	12
122	3.2.2	The iCalendar Components (VComponents).....	12
123	3.2.3	Intervals.....	12
124	3.2.4	Relationships between Intervals.....	13
125	3.2.5	Alarms.....	15
126	3.3	Notification and Synchronization.....	15
127	3.4	Time Stamps.....	15
128	3.4.1	Time Stamp Realm Discussion.....	17
129	4	WS-Calendar Semantics.....	18
130	4.1	Services and Service Characteristics.....	18
131	4.1.1	Attachments.....	18
132	4.1.2	Specifying Timely Performance.....	19
133	4.1.3	Combining Service and Performance.....	20
134	5	WS-Calendar Semantics.....	22
135	5.1.1	Calendar Gluons.....	22
136	5.1.2	Calendar Gluons and Sequences.....	23
137	5.1.3	Optimizing the Sequence for a Partition.....	24
138	5.1.4	Scheduling a Sequence.....	26
139	5.1.5	Mixed Inheritance of Start Time.....	27
140	5.1.6	Other Scheduling Scenarios.....	28
141	6	Calendar Service Interactions: Overview.....	31
142	6.1	Glossary.....	31
143	6.2	Issues not addressed by this specification.....	31
144	7	Service Interactions: Protocol Overview.....	32
145	8	Service Capabilities.....	33
146	9	Creating Calendar Resources.....	34
147	10	Retrieving Calendar Resources.....	35
148	11	Updating Calendar Resources.....	36
149	12	Deletion of Calendar Resources.....	37
150	13	Querying Calendar Resources.....	38
151	14	Conformance.....	39

152 A. Acknowledgements ..... 40  
153 B. Understanding iCalendar, its history, and its use..... 41  
154 C. Overview of WS-Calendar, its Antecedents and its Use ..... 42  
155 C.1 Scheduling Sequences ..... 43  
156 C.1.1 Academic Scheduling example..... 43  
157 C.1.2 Market Performance schedule ..... 44  
158 Revision History ..... 45  
159

---

## 160 Tables

### 161 Index of Tables

162	Table 1: Defining Time Segments for WS-Calendar.....	12
163	Table 2: Temporal Relationships in WS-Calendar.....	13
164	Table 3: Elements of a Temporal Relationship.....	14
165	Table 4: Aspects of Time Stamps.....	15
166	Table 5: Elements of an WS-Calendar Attachment.....	18
167	Table 6: Performance Characteristics.....	19
168	Table 7: Calendar Gluon elements in WS-Calendar.....	22

169  
170

---

### 171 Index of Examples

172	Example 1: Temporal Relationship.....	14
173	Example 2: Temporal Relationship with and without Gap.....	14
174	Example 3: Use of an Attachment with inline XML artifact.....	18
175	Example 4: Use of an Attachment with external reference.....	19
176	Example 5: Performance Component.....	20
177	Example 6: Use of an Attachment with inline XML artifact and optional specified Performance.....	20
178	Example 7: Use of an Attachment with external reference and optional specified performance.....	20
179	Example 8: Sequence with Performance defined in the Calendar Gluon.....	23
180	Example 9: Partition with Duration and Performance defined in the Calendar Gluon.....	24
181	Example 10: Partition in <b>Error! Reference source not found.</b> without annotations.....	25
182	Example 11: A Scheduled Sequence.....	26
183	Example 12: A Scheduled Sequence showing Temporal Relationship Inheritance.....	26
184	Example 13: Partition with Duration and Performance defined in the Calendar Gluon.....	27
185	Example 14: Standard Sequence with Ramp-Up and Ramp Down.....	28

186  
187

---

# 188 1 Introduction

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

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

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

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

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

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

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

## 229 1.1 Terminology

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

## 233 1.2 Normative References

- 234 **RFC2119** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
235 <http://www.ietf.org/RFC/RFC2119.txt>, IETF RFC2119, March 1997.
- 236 **RFC2616** R Fielding, et al. et al, *Hypertext Transfer Protocol -- HTTP/1.1*  
237 <http://tools.ietf.org/html/RFC2616>, IETF RFC2616, June 1999
- 238 **RFC4791** Daboo, et al. *Calendaring Extensions to WebDAV (CalDAV)*.  
239 <http://www.ietf.org/RFC/RFC4791.txt>. IETF RFC 2119, March 2007
- 240 **RFC5545** B. Desruisseaux *Internet Calendaring and Scheduling Core Object*  
241 *Specification (iCalendar)*, <http://www.ietf.org/RFC/RFC5545.txt>, IETF  
242 RFC5545, September 2009.
- 243 **RFC5546** C. Daboo *iCalendar Transport-Independent Interoperability Protocol*  
244 *(iTIP)*, <http://www.ietf.org/RFC/RFC5546.txt>, IETF RFC5546, January  
245 1999.
- 246 **RFC2447** F. Dawson, S. Mansour, S. Silverberg, *iCalendar Message-Based*  
247 *Interoperability Protocol (iMIP)*, <http://www.ietf.org/RFC/RFC2247.txt>,  
248 IETF RFC2447, December 2009.
- 249 **Calendar Resource Schema** C. Joy, C. Daboo, M Douglas, *Schema for representing*  
250 *resources for calendaring and scheduling services*,  
251 <http://tools.ietf.org/html/draft-cal-resource-schema-00>, (Internet-Draft),  
252 April 2010.
- 253 **FreeBusy Read URL** E York. *Freebusy read URL*,  
254 [http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)  
255 [URL%20V1.0.pdf](http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf)
- 256 **SOA-RM** OASIS Standard, *Reference Model for Service Oriented Architecture 1.0*,  
257 October 2006.  
258 <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>
- 259 **Web-Linking** M. Nottingham, *Web linking*. [http://tools.ietf.org/html/draft-nottingham-](http://tools.ietf.org/html/draft-nottingham-http-link-header)  
260 [http-link-header](http://tools.ietf.org/html/draft-nottingham-http-link-header) May 2010
- 261 **draft xCal** C. Daboo, M Douglas, S Lees *xCal: The XML format for iCalendar*,  
262 <http://tools.ietf.org/html/draft-daboo-et-al-icalendar-in-xml-03>, Internet-  
263 Draft, April 2010.
- 264 **XPATH** A Berglund, S Boag, D Chamberlin, MF Fernández, M Kay, J Robie, J  
265 Siméon *XML Path Language (XPath) 2.0*, <http://www.w3.org/TR/xpath20/>  
266 January 2007.
- 267 **XLINK** S DeRose, E Maler, D Orchard, N Walsh *XML Linking Language (XLink)*  
268 *Version 1.1.*, <http://www.w3.org/TR/xlink11/> May 2010.
- 269 **XPOINTER** S DeRose, E Maler, R Daniel Jr. *XPointer xpointer Scheme*,  
270 <http://www.w3.org/TR/xptr-xpointer/> December 2002.
- 271 **XML SCHEMA** PV Biron, A Malhotra, *XML Schema Part 2: Datatypes Second Edition*,  
272 <http://www.w3.org/TR/xmlschema-2/> October 2004.
- 273 **XRD** OASIS XRI Committee Draft 01, *Extensible Resource Descriptor (XRD)*  
274 *Version 1.0*, <http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-cd01.pdf>  
275 October 2009.

## 276 1.3 Non-Normative References

- 277 **NIST Framework and Roadmap for Smart Grid Interoperability Standards**, Office of the  
278 National Coordinator for Smart Grid Interoperability, Release 1.0, NIST  
279 Special Publication 1108,



280 [http://www.nist.gov/public\\_affairs/releases/upload/smartgrid\\_interoperability\\_final.pdf](http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf) January 2010.  
281  
282 **NAESB Smart Grid Requirements** (*dunno what reference I need here*)  
283  
284 **REST** T Fielding, *Architectural Styles and the Design of Network-based*  
285 *Software Architectures*,  
286 <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.  
287

## 288 1.4 Naming Conventions

289 This specification follows some naming conventions for artifacts defined by the specification, as follows:  
290 For the names of elements and the names of attributes within XSD files, the names follow the CamelCase  
291 convention, with all names starting with a lower case letter, eg  
292 

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

  
293 For the names of types within XSD files, the names follow the CamelCase convention with all names  
294 starting with an upper case letter, e.g.,  
295 

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

  
296 For the names of intents, the names follow the CamelCase convention, with all names starting with a  
297 lower case letter, EXCEPT for cases where the intent is to represent an established acronym, in which  
298 case the entire name follows the usage of the established acronym.  
299 An example of an intent which references an acronym is the "SOAP" intent.

## 300 1.5 Architectural References

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

---

## 304 2 Overview of WS-Calendar

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

### 311 2.1 Approach taken by the WS-Calendar Technical Committee

312 The Technical Committee (TC) based its work upon the iCalendar specification as updated in 2009 (IETF  
313 RFC5545) and its the XML serialization [XCAL], currently (2010-07) on a standards track in the IETF.  
314 Members of the Calendaring and Scheduling Consortium (CalConnect.org) developed both updates to  
315 IETF specifications and provided advice to this TC. This work provides the vocabulary for use in this  
316 specification.

317 The committee solicited requirements from a range of interests, notably the NIST Smart Grid Roadmap  
318 and the requirements if the Smart Grid Interoperability Panel (SGIP) as developed by the North American  
319 Energy Standards Board (NAESB). Others submitting requirements included members of the oBIX  
320 technical committee and representative of the FIX Protocol Association. These requirements are reflected  
321 in the semantic elements described in Chapters 3 and 4.

322 In a parallel effort, the CalConnect TC-XML committee developed a number of schedule and calendar-  
323 related services. CalConnect drew on its experience in interoperability between enterprise calendaring  
324 systems as well as interactions with web-based calendars and personal digital assistants (PDAs). These  
325 services were developed as RESTful services by CalConnect and contributed to the WS-Calendar TC.

### 326 2.2 Overview of This Document

327 The specification consists of a standard schema and semantics for schedule and interval information.  
328 These semantic elements are defined and discussed in Section 3.

329 Often the most important service schedule communications involve series of related services over time,  
330 which WS-Calendar defines as a Series. Section 4 discusses the construction of series, and the  
331 association of service attributes to an entire series.

332 Within an iCalendar message, there is a larger document envelope containing transaction and  
333 synchronization information. This information is used for interactions between schedules, calendars, and  
334 calendar collections. The specification defines services for calendar inquiries, event scheduling, event  
335 updating, and event cancellation. RESTful service interactions for scheduling and interactions with  
336 calendars are described in sections 5-nn.

337

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

---

<sup>1</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.

---

## 339 **3 WS-Calendar Semantics**

340 WS-Calendar Elements are semantic elements derived from the [XCAL] specification. These elements  
341 are smaller than a full schedule interaction, and describe the intervals, durations, and time-related events  
342 that are relevant to service interactions. The Elements are used to build a precise vocabulary of time,  
343 duration, sequence, and schedule.

344 WS-Calendar elements elaborate the objects defined in iCalendar, to make interaction requirements  
345 explicit. For example, in human schedule interactions, different organizations have their own  
346 expectations. Meetings may start on the hour or within 5 minutes of the hour. As agents scheduled in  
347 those organizations, people learn the expected precision. In WS-Calendar, that precision must be explicit  
348 to prevent interoperability problems. WS-Calendar defines a performance element elaborate the simple  
349 specification of [XCAL] to make explicit the performance expectations within a scheduled event.

350 WS-Calendar defines common semantics for recording and exchanging event information.

### 351 **3.1 Scheduling Service Performance**

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

355 As defined in the OASIS Reference Model for Service Oriented Architecture 1.0<sup>2</sup>, service requests access  
356 the capability of a remote system.

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

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

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

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

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

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

### 378 **3.2 Core Semantics derived from [XCAL]**

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

---

<sup>2</sup> See normative references in section 1.2

381 information. WS-Calendar relies on [XCAL] standards and data representation to develop its semantic  
382 components.

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

### 384 3.2.1 Time

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

```
387 Year:  
388     YYYY (eg 1997)  
389 Year and month:  
390     YYYY-MM (eg 1997-07)  
391 Complete date:  
392     YYYY-MM-DD (eg 1997-07-16)  
393 Complete date plus hours and minutes:  
394     YYYY-MM-DDThh:mmTZD (eg 1997-07-16T19:20+01:00)  
395 Complete date plus hours, minutes and seconds:  
396     YYYY-MM-DDThh:mm:ssTZD (eg 1997-07-16T19:20:30+01:00)  
397 Complete date plus hours, minutes, seconds and a decimal fraction of a  
398 second  
399     YYYY-MM-DDThh:mm:ss.sTZD (eg 1997-07-16T19:20:30.45+01:00)
```

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

401 In WS-Calendar, unless otherwise noted, all times are un Greenwich Mean Time (UTC).

### 402 3.2.2 The iCalendar Components (VComponents)

403 iCalendar and [XCAL] have a number of long defined component objects that comprise the payload inside  
404 of an iCalendar message. These include the VTOD, the VALARM, the VEVENT. These element names  
405 begin with "V" for historic reasons. The definitions and use of each of the vObjects is described in  
406 [RFC5545].

407 Because of its flexibility, the VTOD object is the basis for WS-Calendar objects for service performance.  
408 Because WS-Calendar services support all traditional iCalendar-based interactions (CalDAV, et al.) all  
409 VComponents SHALL be supported.

### 410 3.2.3 Intervals

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

415 The base data type for time segments is the Interval. The Interval is a time segment defined by the  
416 Duration element as defined in [XCAL]. The [XCAL] duration is a data type based upon the string  
417 representation in the iCalendar duration. The Committee listened to arguments that we should redefine  
418 the use and meaning of Duration. Whatever their merit, the iCalendar Duration has a pre-existing  
419 meaning of the length of time of scheduled within an event. In this section, the Duration is enumerated as  
420 one of several time segments.

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

Time Segment	Definition
Duration	Well-known element from iCalendar and [XCAL], Duration is the length of an event 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 VTOD object that is mandatory in the Interval.

Time Segment	Definition
<b>Interval</b>	The Interval is a single duration supported by the full information set of the VTODO object as defined in iCalendar ([RFC5545]) and refined in [XCAL]. A WS-Calendar interval must include a Duration.
<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.

### 422 3.2.4 Relationships between Intervals

423 Many iCalendar communications involve more than one vComponent. In iCalendar interactions there are  
424 few components they have stereotypical interactions. For example, a vAlarm may be associated with a  
425 vEvent. The registered relationships for iCalendar components are PARENT and Child. In [XCAL], these  
426 are usually expressed as:

```
427 <relationship>
428   <uid>aaaaaaaa1</uid>
429   <reltype>PARENT</reltype>
430 </relationship>
```

431 WS-Calendar instead uses the reltype as an attribute of a relationship to support more expressive XSD  
432 annotation, like this:

```
433 <relationship reltype= "PARENT">
434   <uid>aaaaaaaa1</uid>
435 </relationship>
```

436 This format more easily supports the more expressive relationships used in Sequences.

437 WS-Calendar defines additional relationships to support temporal relationships between intervals. The  
438 relationships express the order of performance and to declare the spacing between those intervals.  
439 These relationships are referred to as the temporal relationships between components.

440 *Table 2: Temporal Relationships in WS-Calendar*

Temporal Relationship	Short Form	Definition
<b>FINISHSTART</b>	FS	As soon as the related Component finishes, this interval begins.
<b>FINISHFINISH</b>	FF	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>STARTFINISH</b>	SF	This component must Finish before the related component starts.

<b>STARTSTART</b>	SS	These Components must start at the same time
<b>Gap</b>		Attribute to indicate the separation, if any, between the state of the first Interval and the state of the second. Expressed as a duration.

441 WS-Calendar specifies more elements in the Relationship to accommodate the needs of Temporal  
442 Relationships. WS-Calendar also extends iCalendar relationship to allow references to external  
443 Components as well as to those internal to the iCalendar object.

444 *Table 3: Elements of a Temporal Relationship*

Relationship Element		Definition
<b>reltype</b>	String, Mandatory	Enumerated list from union of iCalendar and WS-Calendar Temporal Relationships.
<b>guid</b>	string Optional	Identifier of Component in Components collection. Mandatory if uri not present.
<b>reference</b>	[XPOINTER]	Reference to component external to components collection. Mandatory if uri not present.
<b>gap</b>	Duration <i>Optional</i>	Attribute to indicate the separation, if any, between the state of the first Interval and the state of the second. Expressed as a duration. Only used with Temporal Relationships

445 The relationship below indicates that this Interval is to start ten minutes following the finish of interval  
446 aaaaaa1.

447 *Example 1: Temporal Relationship*

```
448 <relationship type="FINISHSTART">
449 <uid>aaaaaaa1</uid>
450 <gap>T00:10</gap>
451 </relationship>
```

452 If there is no temporal separation between Intervals, the gap element is optional. The following examples  
453 are equivalent expressions to express a relationship wherein both intervals must start at the same  
454 moment.

455 *Example 2: Temporal Relationship with and without Gap*

```
456 <relationship type="STARTSTART">
457 <uid>aaaaaaa1</uid>
458 <gap>T00:00</gap>
459 </relationship>
```

460 Leaving out the optional Gap element, we have:

```
461 <relationship type="STARTSTART">
462 <uid>aaaaaaa1</uid>
463 </relationship>
```

464 These two expressions of a Temporal Relationship above are equivalent.

465 Intervals with Temporal Relationships enable the message to express a Sequences as consecutive  
466 Intervals, as in a Partition, or they may express more complex temporal relations.

467 As the rules for parsing XML do not mandate preservation of order within a sub-set, we cannot assume  
468 that order is preserved when parsing a set of Components. For Sequences, mere order is not enough—  
469 each Interval must either refer to or be referred by at least one interval.

### 470 3.2.5 Alarms

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

### 474 3.3 Notification and Synchronization

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

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

482 *A "VALARM" calendar component is a grouping of component properties that is a reminder or alarm for*  
483 *an event or a to-do. For example, it may be used to define a reminder for a pending event or an overdue*  
484 *to-do. The "VALARM" calendar component MUST include the "ACTION" and "TRIGGER" properties. .*  
485 *.The "ACTION" property is used within the "VALARM" calendar component to specify the type of action*  
486 *invoked when the alarm is triggered. The "VALARM" properties provide enough information for a specific*  
487 *action to be invoked<sup>3</sup>.*

488 In WS-Calendar, an alarm is a VALARM object within a VTODDO object, Its actions are [XPOINTER]  
489 references to the service or event that is triggered,

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

494

### 495 3.4 Time Stamps

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

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

503

Table 4: Aspects of Time Stamps

Time Stamp Element	Definition (Normative)	Note (Non-Normative)
<b>timeStamp</b>	WS-Calendar:time A fully qualified date and time of event. Mandatory.	May include two objects as defined above.

<sup>3</sup> From the [RFC5545] – see normative references

Time Stamp Element	Definition (Normative)	Note (Non-Normative)
<b>precision</b>	A Duration defining the accuracy of the TimeStamp value. Mandatory.	Identifies whether one hour interval is indeed one hour or plus or minus some number of milliseconds, seconds and minutes.
<b>timeStampRealm</b>	Of type Uri, shall identify the system where the TimeStamp value originated. The value of this element shall be set by: <ul style="list-style-type: none"> <li>• The component at the realm border in a particular inter-domain interaction or,</li> <li>• By any component able to accurately set it within a system or sub-system.</li> </ul> In the latter case, nothing prevents the component at the realm border to overwrite it without any notice. Optional.	A set of points originating from the same realm 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. A system border is crossed in an interaction when the 2 communication partners are not synchronized based on the same time source. See the example below for more information.
<b>leapSecondsKnown</b>	Xs:boolean If True, shall indicate that the TimeStamp value takes into account all leap seconds occurred. Otherwise False. Optional.	Indicates that the time source of the sending device support leap seconds adjustments.
<b>clockFailure</b>	xs:boolean If True, shall indicate a failure on the time source preventing the TimeStamp value issuer from setting accurate timestamps. Otherwise False. Mandatory.	Indicates that the time source of the sending device is unreliable. The timestamp should be ignored.
<b>clockNotSynchronized</b>	xs:boolean If True, shall indicate the time source of the TimeStamp value issuer is not synchronized correctly, putting in doubt the accuracy of the timestamp. Mandatory.	Indicates that the time source of the sending device is not synchronized with the external UTC time source.
<b>timeSourceAccuracy</b>	A Duration defining the accuracy of the time source used in the TimeStampRealm system. Optional.	Represents the time accuracy class of the time source of the sending device relative to the external UTC time source.



### 504 3.4.1 Time Stamp Realm Discussion

505 Within a single system, or synchronized system of systems, one can sort the temporal order of event by  
506 sorting them by TimeStamp. Determining the order of events is the first step of event forensics. This  
507 assumption does not apply when events are gathered across systems.

508 Different systems may not have synchronized time, or may synchronize time against different sources.  
509 This means different system clocks may drift apart. It may be that a later timestamp from one system  
510 occurred before an earlier timestamp in another. As this drift is unknown, it cannot be automatically  
511 corrected for without additional information.

512 The TimeStampRealm element identifies which system created an event time-stamp. The  
513 TimeStampRealm identifies a source system in inter-domain interactions (a system of systems). For  
514 example: <http://SystemA.com> and <http://SystemB.com> identify 2 systems. This example assumes  
515 SystemA and SystemB do not have a common time source.

516 The TimeStampRealm can also be used to identify sub-systems in intra-domain interactions (sub-systems  
517 of a system). For For example: <http://SystemA.com/SubSystem1> and <http://SystemA.com/SubSystem2>  
518 identify 2 subsystems of the same higher level system. In case the upper level SystemA does not have a  
519 global time source for synchronizing all of its sub-system, it can be useful to identify sub-systems in such  
520 a way.

521

## 4 WS-Calendar Semantics

522

### 4.1 Services and Service Characteristics

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

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

530

#### 4.1.1 Attachments

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

536

Table 5: Elements of an WS-Calendar Attachment

Attachment Element	Use	Discussion
<b>artifact</b>	Any in-line XML. <i>Optional.</i> An attachment 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 An attachment must have at least one of
<b>reference</b>	[XPOINTER] <i>Optional</i> An attachment must have at least one of artifact or reference	Points to external XML, or XML located elsewhere in document
<b>performance</b>	WsCalendar:Performance <i>Optional</i>	Specifies time-related performance characteristics.

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

540

Example 3: Use of an Attachment with inline XML artifact

541

542

543

544

545

546

547

548

549

550

551

552

```

<VTODO>
  <dtstamp></dtstamp>
  <uid>aaaaaaaa</uid>
  <description>first contract</description>
  <summary>defines contract to invoke Hello World Service</summary>
  <duration>T00:15</duration>
  <attach>
    <process name="pns:HelloWorld">
      <active>TRUE</active>
      <service name="wns:HelloWorldService" port="HelloWorldPort"/>
    </process>
  </attach>

```

553

```
</VTODO>
```

554

*Example 4: Use of an Attachment with external reference*

555

```
<VTODO>
556   <dtstamp></dtstamp>
557   <uid>aaaaaaaa1</uid>
558   <description>first contract</description>
559   <summary>defines contract to described at reference</summary>
560   <duration>T00:15</duration>
561   <attach>
562     <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
563   </attach>
564 </VTODO>
```

565

### 566 4.1.2 Specifying Timely Performance

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

570 The Performance component refines the meaning of time-related service communication. All elements of  
571 the Performance object use the Duration element as defined in [RFC5545].

572

*Table 6: 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.

Performance Characteristic	Definition	Discussion
<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.

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

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

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

579 *Example 5: Performance Component*

```
580 <performance>
581   <startbefore>T00:10</startbefore>
582   <startafter>T00:00</startafter>
583   <durationlong>T00:00</durationlong>
584   <durationshort>T00:00</durationshort>
585 </performance>
```

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

589 Generally, the implementer should refrain from expressing unnecessary or redundant performance  
590 characteristics.

### 591 4.1.3 Combining Service and Performance

592 Services, references and performance each appear in the ATTACH element of the iCalendar  
593 components.

594 *Example 6: Use of an Attachment with inline XML artifact and optional specified Performance*

```
595 <VTODO>
596   <dtstamp></dtstamp>
597   <uid>aaaaaaa1</uid>
598   <description>first contract</description>
599   <summary> defines contract to invoke Hello World Service as early as 10
600 minutes before scheduled time, and no later than scheduled time</summary>
601   <duration>T00:15</duration>
602   <attach>
603     <process name="pns:HelloWorld>
604       <active>TRUE</active>
605       <service name="wns:HelloWorldService" port="HelloWorldPort"/>
606     </process>
607     <performance>
608       <startbefore>T00:10</startbefore>
609       <startafter>T00:00</startafter>
610       <durationlong>T00:00</durationlong>
611       <durationshort>T00:00</durationshort>
612     </performance>
613   </attach>
614 </VTODO>
```

615 *Example 7: Use of an Attachment with external reference and optional specified performance*

```
616 <VTODO>
```

```
617 <dtstamp></dtstamp>
618 <uid>aaaaaaaa1</uid>
619 <description>first contract</description>
620 <summary>defines first behavior to perform in contract with a precisions
621 required of 1 second</summary>
622 <duration>T00:15</duration>
623 <attach>
624 <reference>http://scheduled.ws-calendar-service.com/contract1</reference>
625 <performance>
626 <startbefore>T00:10</startbefore>
627 <startafter>T00:00</startafter>
628 <durationlong>T00:00</durationlong>
629 <durationshort>T00:00</durationshort>
630 </performance>
631 </attach>
632 </VTODO>
```

633

## 5 WS-Calendar Semantics

634 Calendar Gluons and Inheritance

### 5.1.1 Calendar Gluons

636 WS-Calendar introduces a new iCalendar component, the Calendar Gluon. An Calendar Gluon is  
 637 essentially a placeholder vComponent (see Appendix *Overview of WS-Calendar, its Antecedents and its*  
 638 *Use*) used to assign attributes to an entire Sequence. Calendar Gluons use the RelatedComponent  
 639 attribute to apply service information to Sequences and Partitions. The use of Calendar Gluons is  
 640 described in Section **Error! Reference source not found.: Error! Reference source not found.**

641

Table 7: Calendar Gluon elements in WS-Calendar

Calendar Gluon Element	Use	Discussion
<b>dtStamp</b>	[XCAL]:dtstamp <i>Mandatory</i>	Time and date that Calendar Gluon object was created
<b>Uid</b>	<i>Mandatory</i>	Used to enable unambiguous referencing of each VTODO object
<b>summary</b>	Text' <i>Optional</i>	Text describing the Calendar Gluon
<b>Related</b>	WsCalendar:Relationship <i>Mandatory</i>	An Calendar Gluon must have a relationship with at least one other component. The only relationship defined for the Calendar Gluon is the IsParent.
<b>dtStart</b>	[XCAL]:Time. Start time for the related interval of the sequence. <i>Optional</i>	An Calendar Gluon may either have a dtStart or a dtEnd, but may not have both.
<b>dtEnd</b>	[XCAL]:Time. Scheduled completion time for the related interval of the sequence. <i>Optional</i>	An Calendar Gluon may either have a dtStart or a dtEnd, but may not have both.
<b>duration</b>	[XCAL]:Duration <i>Optional</i>	If specified, a duration is inherited by all intervals in the referred-to sequence,
<b>Attach</b>	WSCalendar:Attachment <i>Mandatory</i> <i>Multipleoccurs</i>	Contains WS-Calendar:attachment attribute defining service and performance. Can be inherited by all intervals in sequence.

642 Because Calendar Gluon properties are inherited by the associated Sequence, they can serve as the  
 643 elements in any Interval in the Sequence. An inherited element can even serve as a substitute for an  
 644 Interval mandatory element. For example, Duration is mandatory for all Intervals. A Duration expressed in

645 an Calendar Gluon is inherited by each Interval in the associated Sequence. This makes Intervals without  
646 internal Duration compliant, because the Interval inherits the Duration from the Calendar Gluon. If a  
647 Interval in the associated Sequence does include a Duration, that value overrides the value from the  
648 Calendar Gluon.

649 Inheritance is discussed in greater detail in Chapter 4.

## 650 5.1.2 Calendar Gluons and Sequences

651 The Calendar Gluon is used to define common service requirements for an entire sequence. If a  
652 RelatedComponent has a parent relationship with the an Interval in a sequence, then the  
653 RelatedComponent's Attachment defines service attributes by all Intervals in the Sequence.

654 In this example, the Sequence in the previous example is expressed using an Calendar Gluon.

655 *Example 8: Sequence with Performance defined in the Calendar Gluon*

```
656 <components>
657 <Calendar Gluon>
658   <dtstamp></dtstamp>
659   <uid>aaaaaaa0</uid>
660   <description>Calendar Gluon with sequence</description>
661   <summary>creates common performance expectations (+/- 1 second) for
662     entire sequence. Also sets common duration for all members of the
663     sequence.
664   </summary>
665   <duration>T00:15</duration>
666   <attach>
667     <performance>
668       <endbefore>T00:00</endbefore>
669       <endafter>T00:00</endafter>
670       <durationlong>T00:00</durationlong>
671       <durationshort>T00:00</durationshort>
672     </performance>
673   </attach>
674   <related-to>
675     <relationship type="PARENT">
676       <uid>aaaaaaa1</uid>
677     </relationship>
678   </related-to>
679 </Calendar Gluon>
680 <vtodo>
681   <dtstamp></dtstamp>
682   <uid>aaaaaaa1</uid>
683   <description>first contract</description>
684   <summary>inherits performance expectations & duration</summary>
685   <attach>
686     <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
687   </attach>
688 </vtodo>
689 <vtodo>
690   <dtstamp></dtstamp>
691   <uid>aaaaaaa2</uid>
692   <description>second interval</description>
693   <summary>inherits performance expectations & duration</summary>
694   <attach>
695     <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
696   </attach>
697   <related-to>
698     <relationship type="FINISHSTART">
699       <uid>aaaaaaa1</uid>
700     </relationship>
701   </related-to>
702 </vtodo>
```

```

704 <vtodo>
705   <dtstamp></dtstamp>
706   <uid>aaaaaaa3</uid>
707   <description>third interval</description>
708   <summary>inherits performance expectations, overrides duration</summary>
709   <duration>T00.30</duration>
710   <attach>
711     <reference>http://scheduled.ws-calendar-service.com/contract3</reference>
712   </attach>
713   <related-to>
714     <relationship type="FINISHSTART">
715       <uid>aaaaaaa2</uid>
716       <gap>T00:10</gap>
717     </relationship>
718   </related-to>
719 </vtodo>
720 <components>

```

721 This sequence is functionally identical to the one before. Note that the performance expectations,  
722 identical for each interval, have moved into the Calendar Gluon.

723 The Calendar Gluon happens to be related to the first Interval in the sequence; there are specific use cases  
724 (discussed below) which require it to be linked to other Intervals. As a Sequence creates single temporal  
725 relationships, assigning a start time (dtstart) to any Interval allows the starting time to be computed for any  
726 of them.

### 727 5.1.3 Optimizing the Sequence for a Partition

728 Partitions are sequences with consecutive Intervals. Partition communication can be further optimized by  
729 bringing the relationship into the Calendar Gluon. Notice that while the type of the relationship is defined  
730 in the Calendar Gluon, the guid for each interval must still be expressed within the interval.

731 *Example 9: Partition with Duration and Performance defined in the Calendar Gluon*

```

732 <components>
733 <Calendar Gluon>
734   <dtstamp></dtstamp>
735   <uid>aaaaaaa0</uid>
736   <description>Calendar Gluon with sequence</description>
737   <summary>creates common performance expectations (+/- 1 second) for
738     entire sequence. Also sets common duration for all members of the
739     sequence.
740   </summary>
741   <duration>T00:50</duration>
742   <attach>
743     <performance>
744       <startbefore>T00:00</endbefore>
745       <startafter>T00:05</endafter>
746       <durationlong>T00:00</durationlong>
747       <durationshort>T00:05</durationshort>
748     </performance>
749   </attach>
750   <related-to>
751     <relationship type="PARENT">
752       <uid>aaaaaaa1</uid>
753     </relationship>
754     <relationship type="FINISHSTART">
755       <gap>T00:10</gap>
756     </relationship>
757   </related-to>
758 </Calendar Gluon>
759 <vtodo>
760   <dtstamp></dtstamp>
761   <uid>aaaaaaa1</uid>

```



```

762 <description>first contract</description>
763 <attach>
764 <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
765 </attach>
766 </vtodo>
767 <vtodo>
768 <dtstamp></dtstamp>
769 <uid>aaaaaaa2</uid>
770 <description>second interval</description>
771 <attach>
772 <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
773 </attach>
774 <related-to>
775 <relationship><uid>aaaaaaa1</uid></relationship>
776 </related-to>
777 </vtodo>
778 <vtodo>
779 <dtstamp></dtstamp>
780 <uid>aaaaaaa3</uid>
781 <description>third interval</description>
782 <attach>
783 <reference>http://scheduled.ws-calendar-service.com/contract3<reference>
784 </attach>
785 <related-to>
786 <relationship><uid>aaaaaaa2</uid></relationship>
787 </related-to>
788 </vtodo>
789 <components>

```

790 This Partition shows a school schedule in which classes start one hour apart. Each service is performed  
791 for 50 minutes, and there is a 10 minute gap between each as students move between classes. Classes  
792 may not begin before the schedule, but they may start up to five minutes late.

793 Stripped of all annotations, this can be expressed as follows:

794 *Example 10: Partition in **Error! Reference source not found.** without annotations*

```

795 <components>
796 <Calendar Gluon>
797 <dtstamp></dtstamp>
798 <uid>aaaaaaa0</uid>
799 <duration>T00:50</duration>
800 <attach>
801 <performance>
802 <startbefore>T00:00</endbefore>
803 <startafter>T00:05</endafter>
804 <durationlong>T00:00</durationlong>
805 <durationshort>T00:05</durationshort>
806 </performance>
807 </attach>
808 <related-to>
809 <relationship type="PARENT">
810 <uid>aaaaaaa1</uid>
811 </relationship>
812 <relationship type="FINISHSTART">
813 <gap>T00:10</gap>
814 </relationship>
815 </related-to>
816 </Calendar Gluon>
817 <vtodo>
818 <uid>aaaaaaa1</uid>
819 <attach>
820 <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
821 </attach>
822 </vtodo>

```

```

823 <vtodo>
824   <uid>aaaaaaa2</uid>
825   <attach>
826     <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
827   </attach>
828   <related-to>
829     <relationship><uid>aaaaaaa1</uid></relationship>
830   </related-to>
831 </vtodo>
832 <vtodo>
833   <uid>aaaaaaa3</uid>
834   <attach>
835     <reference>http://scheduled.ws-calendar-service.com/contract3<reference>
836   </attach>
837   <related-to>
838     <relationship><uid>aaaaaaa2</uid></relationship>
839   </related-to>
840 </vtodo>
841 </components>

```

842 Notice that the dtstamp for all Intervals in this Partition is inherited from the Calendar Gluon.

### 843 5.1.4 Scheduling a Sequence

844 A Sequence becomes a Scheduled Sequence whenever single interval within the sequence is scheduled.  
845 An interval is scheduled when it has a specific starting time (dtstart).

846 *Example 11: A Scheduled Sequence*

```

847 <components>
848 <vtodo>
849   <dtstamp></dtstamp>
850   <uid>aaaaaaa1</uid>
851   <description>first contract</description>
852   <dtstart>2010-09-11T13:00</dtstart>
853   <duration>T00:15</duration>
854   <attach>
855     <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
856   </attach>
857 </vtodo>
858 <vtodo>
859   <dtstamp></dtstamp>
860   <uid>aaaaaaa2</uid>
861   <description>second interval</description>
862   <duration>T00:15</duration>
863   <attach>
864     <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
865   </attach>
866   <related-to>
867     <relationship type="FINISHSTART">
868       <uid>aaaaaaa1</uid>
869     </relationship>
870     <relationship>
871   </related-to>
872 </vtodo>
873 </components>

```

874 A sequence can also be scheduled in the Calendar Gluon.

875 *Example 12: A Scheduled Sequence showing Temporal Relationship Inheritance*

```

876 <components>
877 <Calendar Gluon>
878   <dtstamp></dtstamp>
879   <uid>aaaaaaa0</uid>

```

```

880 <dtstart>2010-09-11 T00:15</dtstart>
881 <related-to>
882 <relationship type="PARENT">
883 <uid>aaaaaaa1</uid>
884 </relationship>
885 <relationship type="FINISHSTART">
886 <gap>T00:10</gap>
887 </relationship>
888 </related-to>
889 </Calendar Gluon>
890 <vtodo>
891 <dtstamp></dtstamp>
892 <uid>aaaaaaa1</uid>
893 <description>first contract</description>
894 <duration>T00:15</duration>
895 <attach>
896 <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
897 </attach>
898 </vtodo>
899 <vtodo>
900 <dtstamp></dtstamp>
901 <uid>aaaaaaa2</uid>
902 <description>second interval</description>
903 <duration>T00:15</duration>
904 <attach>
905 <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
906 </attach>
907 <related-to>
908 <relationship type="FINISHSTART">
909 <uid>aaaaaaa1</uid>
910 </relationship>
911 <relationship>
912 </related-to>
913 </vtodo>
914 </components>

```

## 915 5.1.5 Mixed Inheritance of Start Time

916 A Sequence is not schedule until it has both a start time and a start date. Start time and date SHALL be  
917 expressed together when all components are in a single communication. Time and Date MAY be  
918 separated when the full sequence and schedule are created by reference.

919 To illustrate this, here is the classroom scheduling Partition from **Error! Reference source not found.**,  
920 updated to include each day's school opening.

921 *Example 13: Partition with Duration and Performance defined in the Calendar Gluon*

922 <http://scheduled.ws-calendar-service.com/classSchedule>

```

923 <components>
924 <Calendar Gluon>
925 <dtstamp></dtstamp>
926 <uid>aaaaaaa0</uid>
927 <dtstart>T19:00</dtstart>
928 <description>Classroom Schedule</description>
929 <duration>T00:50</duration>
930 <related-to>
931 <relationship type="PARENT">
932 <uid>aaaaaaa1</uid>
933 </relationship>
934 <relationship type="FINISHSTART">
935 <gap>T00:10</gap>
936 </relationship>
937 </related-to>

```

```

938 </Calendar Gluon>
939 <vtodo>
940 <dtstamp></dtstamp>
941 <uid>aaaaaaa1</uid>
942 <description>first interval</description>
943 <attach>
944 <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
945 </attach>
946 </vtodo>
947 <vtodo>
948 <dtstamp></dtstamp>
949 <uid>aaaaaaa2</uid>
950 <description>second interval</description>
951 <attach>
952 <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
953 </attach>
954 <related-to>
955 <relationship><uid>aaaaaaa1</uid></relationship>
956 </related-to>
957 </vtodo>
958 <vtodo>
959 <dtstamp></dtstamp>
960 <uid>aaaaaaa3</uid>
961 <description>third interval</description>
962 <attach>
963 <reference>http://scheduled.ws-calendar-service.com/contract3<reference>
964 </attach>
965 <related-to>
966 <relationship><uid>aaaaaaa2</uid></relationship>
967 </related-to>
968 </vtodo>
969 <components>

```

970 and the invoking Calendar Gluon for a given day:

```

971 <components>
972 <Calendar Gluon>
973 <dtstamp></dtstamp>
974 <uid>aaaaaaaac</uid>
975 <dtstart>2010-09-11</dtstart>
976 <related-to>
977 <relationship type="PARENT">
978 <reference>http://scheduled.ws-calendar-service.com/classSchedule
979 </reference>
980 </relationship>
981 </related-to>
982 </Calendar Gluon>
983 <components>

```

984 In this case, the Sequence is offered at 13:00. The Sequence is not yet scheduled because a schedule  
985 requires a full start date and time. The Sequence has an external reference. The Calendar Gluon  
986 schedules a particular performance of this sequence on 2010-09-11. The date from the invocation and  
987 the time from the offering are combined to produce 2010-09-11T13:00 and the result is a Scheduled  
988 Sequence.

## 989 5.1.6 Other Scheduling Scenarios

990 Sometime the invoker of a service is interested only in single Interval of the Sequence, but the entire  
991 Sequence is required. In this case, it is valuable to invoke the Sequence by a particular interval.

992 *Example 14: Standard Sequence with Ramp-Up and Ramp Down*

993 <http://scheduled.ws-calendar-service.com/anotherSchedule>

```

994 <components>

```

```

995 <Calendar Gluon>
996 <dtstamp></dtstamp>
997 <uid>aaaaaaa0</uid>
998 <dtstart>T19:00</dtstart>
999 <description>Note reference to second interval</description>
1000 <duration>T00:50</duration>
1001 <related-to>
1002 <relationship type="PARENT">
1003 <uid>aaaaaaa2</uid>
1004 </relationship>
1005 <relationship type="FINISHSTART">
1006 <gap>T00:10</gap>
1007 </relationship>
1008 </related-to>
1009 </Calendar Gluon>
1010 <vtodo>
1011 <dtstamp></dtstamp>
1012 <uid>aaaaaaa1</uid>
1013 <description>first interval</description>
1014 <summary>set up process required before second interval can be performed.
1015 <summary>setup takes a fixed time
1016 </summary>
1017 <duration>T00:50</duration>
1018 <attach>
1019 <reference>http://scheduled.ws-calendar-service.com/contract1<reference>
1020 </attach>
1021 </vtodo>
1022 <vtodo>
1023 <dtstamp></dtstamp>
1024 <uid>aaaaaaa2</uid>
1025 <description>second interval</description>
1026 <summary>This is the one that the invoker is interested in. The invoker
1027 <summary>may invoke this event for whatever period desired. Note that the
1028 <summary>external Calendar Gluon is parent to this Interval
1029 </summary>
1030 <attach>
1031 <reference>http://scheduled.ws-calendar-service.com/contract2<reference>
1032 </attach>
1033 <related-to>
1034 <relationship><uid>aaaaaaa1</uid></relationship>
1035 </related-to>
1036 </vtodo>
1037 <vtodo>
1038 <dtstamp></dtstamp>
1039 <uid>aaaaaaa3</uid>
1040 <description>third interval</description>
1041 <summary>however long the second interval takes, this interval must run
1042 <summary>take 15 minutes for completion.
1043 </summary>
1044 <duration>T00:15</duration>
1045 <attach>
1046 <reference>http://scheduled.ws-calendar-service.com/contract3<reference>
1047 </attach>
1048 <related-to>
1049 <relationship><uid>aaaaaaa2</uid></relationship>
1050 </related-to>
1051 </vtodo>
1052 </component>

```

1053 When the service is scheduled, the time and duration are specified. The duration only applies to the  
1054 Second Interval as all others have their duration explicitly specified.

```

1055 <components>
1056 <Calendar Gluon>

```

```
1057 <dtstamp></dtstamp>
1058 <uid>aaaaaaac</uid>
1059 <dtstart>2010-09-11T19:00</dtstart>
1060 <duration>T01:15</duration>
1061 <description>Classroom Schedule</description>
1062 <duration>T00:50</duration>
1063 <related-to>
1064   <relationship type="PARENT">
1065     <reference>http://scheduled.ws-calendar-service.com/classSchedule
1066     <uid>aaaaaaa1</uid>
1067   </relationship>
1068 </related-to>
1069 </Calendar Gluon>
```

1070 In this case, the specific interval is scheduled and a run time of 75 minutes is specified.

1071

---

1072 **6 Calendar Service Interactions: Overview**

1073 This OASIS Committee has worked closely with the CalConnect TC-XML committee, which publishes its  
1074 work through the IETF<sup>4</sup>. CalConnect is defining the core scheduling service interactions, i.e., scheduling  
1075 an event, determining availability, etc., and publishing them as Cal-WS.

1076 **6.1 Glossary**

1077 3.1 Hrefs ..... 8  
1078 3.2 Calendar Object Resource 8  
1079 3.3 Calendar Collection .....8  
1080 3.4 Scheduling Calendar Collection .....

1081  
1082 **6.2 Issues not addressed by this specification**

1083 2.1 Access Control ... 7  
1084 2.2 Creating Collections .....7  
1085 2.3 Provisioning ..... 7  
1086 2.4 Discovery ..... 7  
1087 2.5 Retrieving collections .....7  
1088

---

<sup>4</sup> <http://datatracker.ietf.org/wg/calsify/charter/>

---

1089 **7 Service Interactions: Protocol Overview**

1090 4 Overview of the protocol .....9  
1091 4.1 Error conditions .. 9  
1092 4.2 HTTP Methods ... 9  
1093 4.3 Operations ..... 9  
1094 4.4 Calendar Object Resources .....9  
1095 4.5 Timezone information .....10



1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103

---

## 8 Service Capabilities

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

5 Capabilities ..... 11

5.1 Request parameters .....11

5.2 Responses: .....11

5.3 Example: ..... 11

---

1104 **9 Creating Calendar Resources**

1105 6 Creating Calendar Object Resources ...12  
1106 6.1 Request parameters .....12  
1107 6.2 Responses: .....12  
1108 6.3 Preconditions for PUT, COPY, and MOVE ..12  
1109 6.4 Example: ..... 13  
1110

---

1111 **10 Retrieving Calendar Resources**

1112 7 Retrieving resources 14  
1113 7.1 Request parameters .....14  
1114 7.2 Responses: .....14  
1115

---

1116 **11 Updating Calendar Resources**

1117 8 Updating resources .. 15

1118 8.1 Responses: .....15

1119

---

1120 **12 Deletion of Calendar Resources**

1121 9 Deletion of resources ..... 16

1122 9.1 Responses: .....16

1123

---

1124 **13 Querying Calendar Resources**

1125 10 Querying calendar resources 1

1126

---

1127 **14 Conformance**

1128 WS-Calendar Intervals SHALL have a Duration. Intervals MAY have a StartTime. Intervals SHALL NOT  
1129 include an END time. If a non-compliant Interval is received with an END time, it may be ignored.

1130 A performance component SHALL not include Start, Stop, and Duration elements. Two out of the three  
1131 elements is acceptable, but not three.

1132 In Partitions, the Description, Summary and Priority of each Interval SHALL be excluded.

1133 An Calendar Gluon may either have a dtStart or a dtEnd, but may not have both.

1134 *All OASIS specifications require conformance*

1135

---

## A. Acknowledgements

1136 The following individuals have participated in the creation of this specification and are gratefully  
1137 acknowledged:

1138 **Participants:**

1139 Brad Benson, Trane  
1140 Edward Cazalet, Individual  
1141 Toby Considine, University of North Carolina at Chapel Hill  
1142 William Cox, Individual  
1143 Craig Gemmill, Tridium, Inc.  
1144 Girish Ghatikar, Lawrence Berkeley National Laboratory  
1145 Gale Horst, Electric Power Research Institute (EPRI)  
1146 Gershon Janssen, Individual  
1147 Ed Koch, Akuacom Inc.  
1148 Benoit Lepeuple, LonMark International\*  
1149 Carl Mattocks, CheckMi\*  
1150 Robert Old, Siemens AG  
1151 Alexander Papaspyrou, Technische Universitat Dortmund  
1152 Jeremy Roberts, LonMark International\*  
1153 David Thewlis, CalConnect

1154

1155 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-  
1156 Calendar Technical Committee, bridging to developing IETF standards and contributing the Services  
1157 definitions that make up Section 6, Calendar Service Interactions. The Technical Committee gratefully  
1158 acknowledges their assistance and cooperation as well.

1159



---

1160 **B. Understanding iCalendar, its history, and its use**

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

1163 Non normative stuff coming from CalConnect

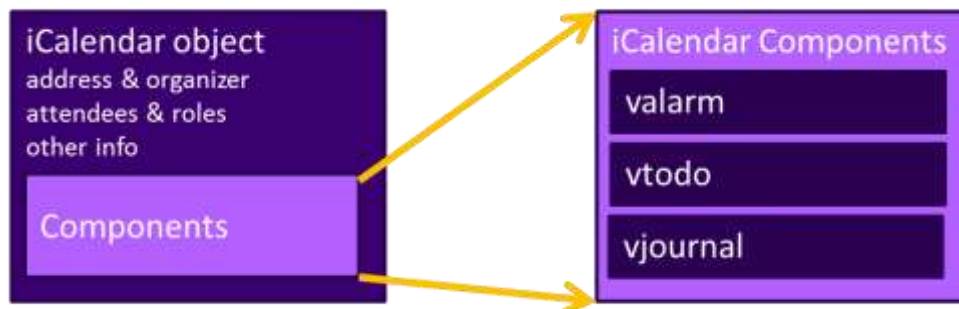
1164

## C. Overview of WS-Calendar, its Antecedents and its Use

iCalendar has long been the predominant message format for an Internet user to send meeting requests and tasks to other Internet users by email. The recipient can respond to the sender easily or counter propose another meeting date/time. iCalendar support is built into all major email systems and email clients. While SMTP is the predominant means to transport iCalendar messages, protocols including WebDAV and SyncML are used to transport collections of iCalendar information. No similar standard for service interactions has achieved similar widespread use.

The Calendar and Scheduling Consortium (CalConnect), working within the IETF, updated the iCalendar standard in the summer of 2009 to support extension ([RFC5545]). In 2010, the same group defined [XCAL], a canonical XML serialization for iCalendar, currently (08/21/2008) on the recommended standards track within the IETF. This specification supports extensions, including handling non-standard, i.e., non-iCalendar, data during message storage and retrieval.

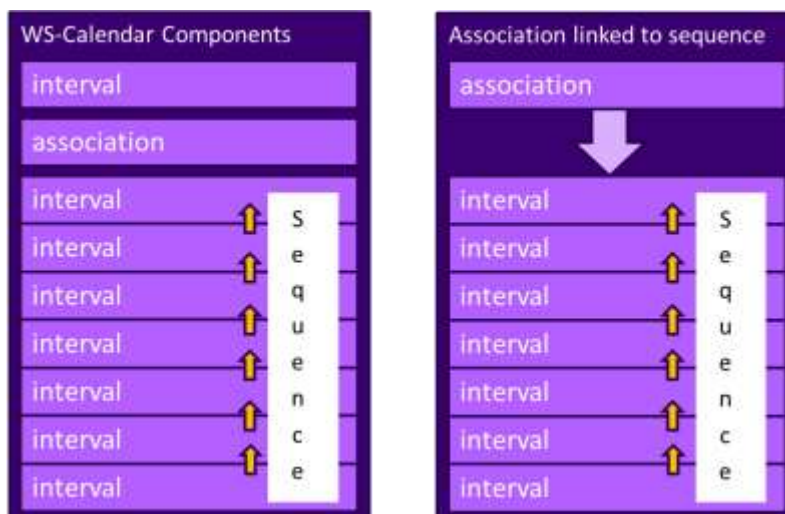
WS-Calendar builds on this work, and consists of extensions to the vocabulary of iCalendar, along with standard services to extend calendaring and scheduling into service interactions. iCalendar consists of a number of fields that support the delivery, update, and synchronization of if calendar messages and a list of components. The components can specify defined relationships between each other.



1182  
1183

Figure 1: iCalendar overview

WS-Calendar defines the Interval, a profile of the vtodo component requiring only a duration and an artifact to define service delivery and performance. WS-Calendar also defines the Calendar Gluon component, a container for holding only a service delivery and performance artifact, to associate with a component or group of components.



1188

1189

Figure 2: WS-Calendar and EMIX

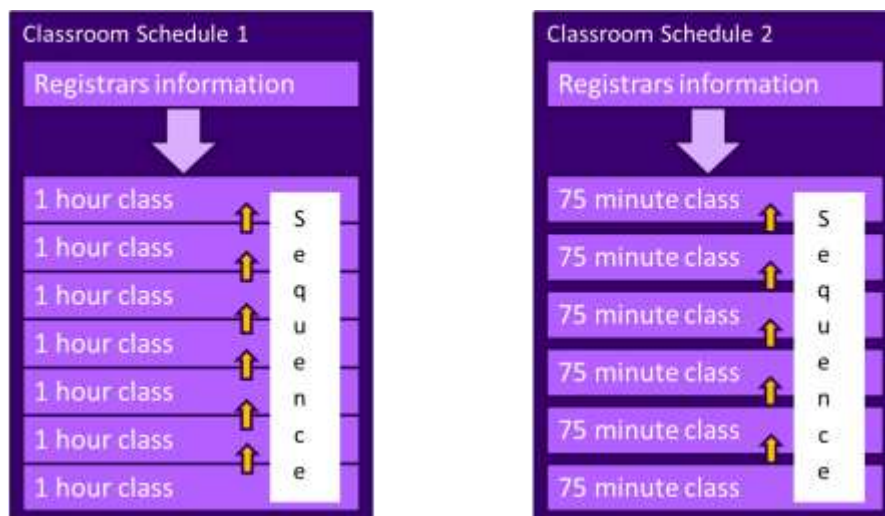
1190 A set of intervals that have defined temporal relationships is a Sequence. Temporal relationships express  
1191 how the occurrence of one interval is related to another. For example, Interval B may begin 10 minutes  
1192 after Interval A completes, or Interval D may start 5 minutes after Interval C starts. An Calendar Gluon  
1193 linked to a Sequence defines service performance for all Intervals in the Sequence. Because each  
1194 interval has its own service performance contract, specifications built on WS-Calendar can define rules for  
1195 inheritance and over-rides with a sequence.

1196 The Partition is a sub-class of a Sequence in which all Intervals follow consecutively with no lag time.  
1197 Intervals in a Partition normally have the same Duration, but WS-Calendar does support overriding the  
1198 duration on an individual basis.

## 1199 C.1 Scheduling Sequences

1200 A Sequence is a general pattern of behaviors and results that does not require a specific schedule. A  
1201 publishing service may advertise a Sequence with no schedule, i.e., no specific time for performance.  
1202 When the Sequence is invoked or contracted, a specific performance time is added. In the original  
1203 iCalendar components, this would add the starting date and time (dtStart) to the component. In WS-  
1204 Calendar, we add the starting date and time only to the first Interval of a Sequence; the performance  
1205 times for all other Intervals in the Sequence are derived from that one start time.

### 1206 C.1.1 Academic Scheduling example



1207

1208

Figure 3: Classroom Scheduling Example

1209 A college campus uses two schedules to schedule its buildings. In Schedule 1, classes start on the hour,  
1210 and follow one after another; each class starts on the hour. In the second schedule, each class lasts an  
1211 hour and a quarter, and there is a fifteen minute gap between classes; classes start on the half hour. On  
1212 many campuses, the sequence in Schedule 1 may describe classes taught on Monday, Wednesday, and  
1213 Friday. Schedule 2 may describe classes taught on Tuesday and Thursday.

1214 The registrar's office knows some key facts about each classroom, including whether it hosts a class  
1215 during a particular period, and the number of students that will be in that class. The college wishes to  
1216 optimize the provision of building services for each class. Such services may include adequate ventilation  
1217 and comfortable temperatures to assure alert students. Other services may ensure that the classroom  
1218 projection systems and A/V support services are warmed up in advance of a class, or powered off when a  
1219 classroom is vacant.

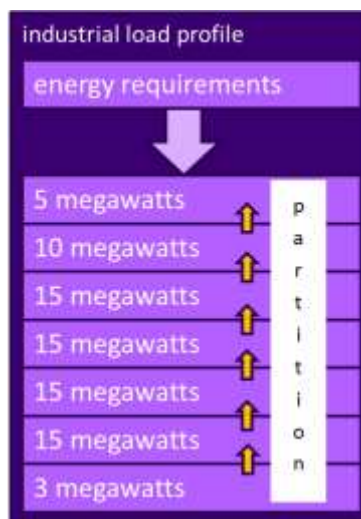
1220 Although most classes meet over typical schedule for the week (M-W-F or Tu-Th), some classes may not  
1221 meet on Friday, or may have a tutorial section one day a week. The registrar's system, ever mindful of  
1222 student privacy, shares only minimal information with the building systems such as how many students  
1223 will be supported.

1224 The Registrar's system schedule building systems using the Calendar Gluon (registrar's information) and  
1225 the student counts for each interval, and schedules the Sequence in classroom schedule 1 three days a  
1226 week for the next 10 weeks. The Registrar's system also schedules the sequence in classroom schedule  
1227 2 two days a week, also for 10 weeks.

1228 This example demonstrates a system (A) that offers services using either of two sequences. Another  
1229 business system (B) with minimal knowledge of how (A) works determines the performance requirements  
1230 for (A). The business system (B) communicates these expectations are by scheduling the Sequences  
1231 offered by (A).

### 1232 C.1.2 Market Performance schedule

1233 A factory relies on an energy-intensive process which is performs twice a year for eight weeks. The  
1234 factory has some flexibility about scheduling the process; it can perform the work in either the early  
1235 morning or the early evening; it avoids the afternoon when energy costs are highest. The factory works up  
1236 a detailed profile of when it will need energy to support this process.



1237  
1238 *Figure 4: Daily Load Profile for Market Operations Example*

1239 Factory management has decided that they want to use only renewable energy products for this process.  
1240 They approach two regional wind farms with the intent of making committed purchases of wind energy.  
1241 The wind farms consider their proposals taking into account the seasonal weather forecasts they use to  
1242 project their weather capacity, and considering the costs that may be required to buy additional wind  
1243 energy on the spot market to make up any shortfalls.

1244 Each energy supplier submits of the same sequence, a schedule, i.e. a daily starting time, and a price for  
1245 the season's production. After considering the bids, and other internal costs of each proposal, the factory  
1246 opts to accept a contract for the purchase of a fixed load profile (Partition), using the evening wind  
1247 generation from one of the suppliers. This contract specifies Schedules of load purchases (starting data  
1248 and time for the sequence) for each day.

1249

1250

## Revision History

1251

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
1.0 WD 10	2010-08-28	Toby Considine, Benoit Lepeuple	Updated Time Stamp section Added background Appendices Incorporated Association language to replace RelatedComponent Recast examples to show inheritance, remove inconsistencies
1.0 WD 11	2010-09-11	Toby Considine	Traceability Release in support of a re-shuffling of the document. Sections 3, 4 were re-shuffled to create: 3: Interval / Relationships / Time Stamps 4: Performance / Attachments 5: Associations & Inheritance Also, changed all associations to Gluons. No paragraphs have been changed, just shuffled, changes accepted, to create clean base for editing

1252

1253