



WS-Calendar Version 1.0

**Committee Specification Draft 0304 /
Public Review Draft 0203**

25 March 17 June 2011

Specification URLs:

This Versionversion:

<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd02/ws-calendar-spec-v1.0-csprd02.pdf> (Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd02/ws-calendar-spec-v1.0-csprd02.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd02/ws-calendar-spec-v1.0-csprd02.doc>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd03/ws-calendar-spec-v1.0-csprd03.pdf> (Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd03/ws-calendar-spec-v1.0-csprd03.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd03/ws-calendar-spec-v1.0-csprd03.doc>

Previous Versionversion:

<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csd02/ws-calendar-spec-v1.0-csd02.pdf> (Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csd02/ws-calendar-spec-v1.0-csd02.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csd02/ws-calendar-spec-v1.0-csd02.doc>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csd03/ws-calendar-spec-v1.0-csd03.pdf> (Authoritative)
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csd03/ws-calendar-spec-v1.0-csd03.html>
<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csd03/ws-calendar-spec-v1.0-csd03.doc>

Latest Versionversion:

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

Technical Committee:

OASIS Web Services Calendar (WS-Calendar) TC

OASIS Web Services Calendar (WS-Calendar) TC

Chair(s):

[Toby Considine](#)

Editor(s):

[Toby Considine](#)

[Mike Douglass](#)

[Toby Considine](#)

Editors:

[Toby Considine](#)

[Mike Douglass](#)

Related work:

[XML Schemas for WS-Calendar Version 1.0](#)

This specification is related to:

- [IETF RFC5545](#), iCalendar
- [IETF RFC5546](#), iCalendar Transport
- [IETF RFC2447](#), iCalendar Message Based Interoperability
- [IETF XCAL](#) [IETF XCAL specification in progress](#)
- [IETF / CalConnect Calendar Resource Schema](#) [IETF / CalConnect Calendar Resource Schema specification in progress](#)

Declared XML Namespace(s):

[urn:ietf:params:xml:ns:calendar-2.0](#)

[XML schemas: ws-calendar-spec/v1.0/csprd03/xsd/](#)

Abstract:

WS-Calendar describes:

- A semantic (or information) model for exchange of calendar information to coordinate activities
- A means of synchronizing and maintaining calendars

The specification includes XML vocabularies for the interoperable and standard exchange of:

- Schedules, including sequences of schedules
- Intervals, including sequences of Intervals
- Other calendar information consistent with the IETF iCalendar standards

These vocabularies describe schedules and Intervals future, present, or past (historical).

~~In this Working Draft the means for synchronizing and maintaining calendars uses REST; in a future version a web services set of services will be defined. The document is divided into three parts; Parts 1 and 2 are in version 1.0; Part 3 will be in a later version.~~

[The specification is divided into three parts.](#)

- 1) The semantic information model and XML vocabularies for exchanging schedule information
- 2) RESTful Services for calendar update and synchronization
- 3) Web services for calendar update and synchronization

[The Technical Committee has decided not to publish Parts 2 and 3 until a later version.](#)

Status:

This document was last revised or approved by the [OASIS Web Services Calendar \(WS-Calendar\) TC](#) on the above date. The level of

approval is also listed above. Check the “Latest [Version](#)[version](#)” location noted above for possible later revisions of this document.

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

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

Citation Format:

When referencing this specification the following citation format should be used:

[ws-calendar-spec] WS-Calendar

WS-Calendar Version 1.0. [25 March](#)[17 June](#) 2011. OASIS Committee Specification [Draft 04](#) / Public Review Draft [02](#). <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd02/ws-calendar-spec-v1.0-csprd02.html>[03](#). <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd03/ws-calendar-spec-v1.0-csprd03.pdf>.

Notices

Copyright © OASIS Open 2011. All Rights Reserved.

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

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

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

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

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

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

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

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

Table of Contents

1	Introduction	12
1.1	Terminology	13
1.2	Normative References	13
1.3	Non-Normative References	15
1.4	Namespace	16
1.5	Naming Conventions	17
1.6	Editing Conventions	17
1.7	Architectural References	17
1.8	Semantics	18
2	Overview of WS-Calendar	23
2.1	Approach taken by the WS-Calendar Technical Committee	23
2.2	Scheduling Service Performance	23
2.2.1	Which Time? UTC vs. Local Time	24
2.3	Overview of This Document	25
3	PART ONE: Semantic Model for WS-Calendar	26
3.1	Intervals, Temporal Relations, and Sequences	26
3.1.1	Core Semantics derived from [XCAL]	26
3.1.2	Intervals	27
3.1.3	Connecting the Intervals	29
3.1.4	Sequences: Combining Intervals	31
3.1.5	State Changes	34
3.2	Attachments and Timely Performance	35
3.2.1	Attachment and the Artifact	35
3.2.2	Specifying Timely Performance	37
3.2.3	Expressing Service and Tolerance	40
3.3	Using Sequences: referencing, modifying, and remote access	40
3.3.1	References and Inheritance	41
3.3.2	Calendar Gluons and Sequences	45
3.3.3	Inheritance rules for Calendar Gluons	49
3.3.4	Optimizing the expression of a Partition	49
3.3.5	Controlling Start Times in Service Advertisements	56
3.3.6	Other Scheduling Scenarios	60
3.4	Time Stamps	63
3.4.1	Time Stamp Realm Discussion	65
4	PART TWO: Calendar Update and Synchronization with RESTful Services	67
4.1	Calendar Services	67
4.1.1	Overview of the protocol	67
4.1.2	Error conditions	69
4.2	Properties and link relations	69
4.2.1	Property and relation-type URLs	69
4.2.2	supported-features property	70
4.2.3	max-attendees-per-instance	70
4.2.4	max-date-time	70

4.2.5 max- <u>instances</u>	70
4.2.6 max- <u>resource-size</u>	70
4.2.7 min- <u>date-time</u>	70
4.2.8 description	70
4.2.9 timezone-service relation	70
4.2.10 principal-home relation	71
4.2.11 current-principal freebusy relation	71
4.2.12 principal freebusy relation	71
4.2.13 child-collection relation	71
4.2.14 created-link property	71
4.2.15 last-modified property	71
4.2.16 displayname property	72
4.2.17 timezone property	72
4.2.18 owner property	72
4.2.19 collection-link property	72
4.2.20 calendar-collection-link property	72
4.2.21 CalWS:privilege-set XML element	72
4.3 Retrieving Collection and Service Properties	73
4.3.1 Request parameters	73
4.3.2 Responses:	73
4.3.3 Example - retrieving server properties:	73
4.4 Creating Calendar Object Resources	74
4.4.1 Request parameters	74
4.4.2 Responses:	74
4.4.3 Preconditions for Calendar Object Creation	74
4.4.4 Example - successful POST:	75
4.4.5 Example - unsuccessful POST:	75
4.5 Retrieving resources	76
4.5.1 Request parameters	76
4.5.2 Responses:	76
4.5.3 Example - successful fetch:	76
4.5.4 Example - unsuccessful fetch:	76
4.6 Updating resources	77
4.6.1 Responses:	77
4.7 Deletion of resources	78
4.7.1 Delete for Collections	78
4.7.2 Responses:	78
4.8 Querying calendar resources	78
4.8.1 Limiting data returned	78
4.8.2 Pre/postconditions for calendar queries	79
4.8.3 Example: time range limited retrieval	79
4.9 Free busy queries	82
4.9.1 ACCEPT header	83
4.9.2 URL Query Parameters	83
4.9.3 URL parameters - notes	84

4.9.4 HTTP Operations	84
4.9.5 Response Codes	84
4.9.6 Examples	84
5 Conformance and Rules for WS-Calendar and Referencing Specifications	87
5.1 Introduction	87
5.2 Semantic Conformance Rules for WS-Calendar	87
5.2.1 Inheritance in WS-Calendar	87
5.2.2 Specific Attribute Inheritance in WS-Calendar	88
5.2.3 Conformance of Intervals in WS-Calendar	89
5.2.4 Conformance of Bound Intervals and Sequences in WS-Calendar	89
5.3 Conformance Rules for RESTful Services	90
5.4 Conformance Rules for Specifications Claiming Conformance to WS-Calendar	90
A Acknowledgements	92
B An Introduction to Internet Calendaring	93
B.1 icalendar	93
B.1.1 History	93
B.1.2 Data model	93
B.1.3 Scheduling	94
B.1.4 Extensibility	94
B.2 Calendar data access and exchange protocols	94
B.2.1 Internet Calendar Subscriptions	94
B.2.2 CalDAV	94
B.2.3 ActiveSync/SyncML	95
B.2.4 CalWS	95
B.2.5 iSchedule	95
B.3 References	95
C Overview of WS-Calendar, its Antecedents and its Use	96
C.1 Scheduling Sequences	97
C.1.1 Academic Scheduling example	97
C.1.2 Market Performance schedule	98
D Revision History	100
1 Introduction	12
1.1 Terminology	13
1.2 Normative References	13
1.3 Non-Normative References	15
1.4 Contributions	16
1.5 Namespace	16
1.6 Naming Conventions	17
1.7 Editing Conventions	17
1.8 Architectural References	17
1.9 Semantics	18
2 Overview of WS-Calendar	23
2.1 Approach taken by the WS-Calendar Technical Committee	23
2.2 Communicating Schedules and Service Performance	23
2.2.1 Which Time? UTC vs. Local Time	24

2.3 Overview of This Document	25
2.4 Security Considerations	25
3 PART ONE: Information model for WS-Calendar	26
3.1 Intervals, Temporal Relations, and Sequences	26
3.1.1 Core Semantics derived from [XCAL]	26
3.1.2 Intervals	27
3.1.3 Connecting the Intervals	29
3.1.4 Sequences: Combining Intervals	31
3.1.5 State Changes	34
3.2 Attachments and Tolerance	35
3.2.1 Attachment and the Artifact	35
3.2.2 Tolerance: What is Timely Performance	37
3.3 Using Sequences: referencing, modifying, and remote access	40
3.3.1 References and Inheritance	41
3.3.2 Gluons and Sequences	45
3.3.3 Inheritance rules for Gluons	49
3.3.4 Optimizing the expression of a Partition	49
3.3.5 Notifying Partners of Process Availability	56
3.3.6 Other Scheduling Scenarios	60
3.4 Time Stamps	63
3.4.1 Time Stamp Realm Discussion	65
4 Conformance and Rules for WS-Calendar and Referencing Specifications	67
4.1 Introduction	87
4.2 Conformance Rules for WS-Calendar	87
4.2.1 Inheritance in WS-Calendar	87
4.2.2 Specific Attribute Inheritance	88
4.2.3 General Conformance Issues	88
4.2.4 Covarying Elements	89
4.2.5 Conformance of Intervals	89
4.2.6 Conformance of Bound Intervals and Sequences	89
4.3 Conformance Rules for Specifications Claiming Conformance to WS-Calendar	90
4.4 Security Considerations	90
A Acknowledgements	92
B An Introduction to Internet Calendaring	93
B.1 icalendar	93
B.1.1 History	93
B.1.2 Data model	93
B.1.3 Scheduling	94
B.1.4 Extensibility	94
B.2 Calendar data access and exchange protocols	94
B.2.1 Internet Calendar Subscriptions	94
B.2.2 CalDAV	94
B.2.3 ActiveSync/SyncML	95
B.2.4 CalWS	95
B.2.5 iSchedule	95

<u>B.3 References</u>	95
<u>C. Overview of WS-Calendar, its Antecedents and its Use</u>	96
<u>C.1 Scheduling Sequences</u>	97
<u>C.1.1 Academic Scheduling example</u>	97
<u>C.1.2 Market Performance schedule</u>	98
<u>D. Revision History</u>	100

Tables

Index of Tables

<u>Table 1-1: Namespaces used in this specification</u>	16
<u>Table 1-2: Schemas and Extensions Used in this Specification</u>	16
<u>Table 1-3: Semantics: Foundational Elements</u>	18
<u>Table 1-4: Semantics: Relations, Limits, and Constraints</u>	19
<u>Table 1-5: Semantics: Inheritance</u>	19
<u>Table 1-6: Semantics: Describing Intervals</u>	20
<u>Table 3-1: Properties of Intervals</u>	28
<u>Table 3-2: Temporal Relationships</u>	29
<u>Table 3-3: Introducing the Sequence</u>	31
<u>Table 3-4: Elements of a WS-Calendar Attachment</u>	36
<u>Table 3-5: Performance Tolerance</u>	37
<u>Table 3-6: Calendar Gluon Elements</u>	42
<u>Table 3-7: Availability elements with specified use in WS-Calendar</u>	43
<u>Table 3-8: Availability elements contained within Availability</u>	44
<u>Table 3-9: Gluon Inheritance rules</u>	49
<u>Table 3-10: Aspects of Time Stamps</u>	63
<u>Table 1-1: Namespaces used in this specification</u>	16
<u>Table 1-2: Schemas and Extensions Used in this Specification</u>	16
<u>Table 1-3: Semantics: Foundational Elements</u>	18
<u>Table 1-4: Semantics: Relations, Limits, and Constraints</u>	19
<u>Table 1-5: Semantics: Inheritance</u>	19
<u>Table 1-6: Semantics: Describing Intervals</u>	20
<u>Table 3-1: Elements of Intervals</u>	28
<u>Table 3-2: Temporal Relationships</u>	29
<u>Table 3-3: Elements of a WS-Calendar Attachment</u>	35
<u>Table 3-4: Tolerance Elements</u>	37
<u>Table 3-5: Gluon Elements</u>	42
<u>Table 3-6: Gluon Inheritance rules</u>	49
<u>Table 3-7: Elements of Time Stamps</u>	63

Index of Examples

<u>Example 3-1: An Unanchored Interval</u>	28
<u>Example 3-2: Temporal Relationship</u>	30
<u>Example 3-3: Temporal Relationship with Gap</u>	30
<u>Example 3-4: Temporal Relationship without Gap</u>	30
<u>Example 3-5: An Anchored Sequence</u>	32
<u>Example 3-6: State Change communication</u>	34

<u>Example 3-7: Use of an Attachment with inline XML artifact</u>	36
<u>Example 3-8: Use of an Attachment with external reference</u>	36
<u>Example 3-9: Performance Component</u>	39
<u>Example 3-10: Interval with inline XML artifact and optional specified Performance</u>	40
<u>Example 3-11: Sequence with Performance defined in the Calendar Gluon</u>	45
<u>Example 3-12: Partition with Duration and Relationship defined in the Calendar Gluon</u>	49
<u>Example 3-13: Partition with Duration, Relationship, and Gap defined in the Calendar Gluon</u>	52
<u>Example 3-14: Vavailability</u>	56
<u>Example 3-15 Gluon publishing availability of pre-existing sequence</u>	58
<u>Example 3-16: Standard Sequence with Ramp-Up and Ramp Down</u>	60
<u>Example 4-1: Successful Update</u>	77
<u>Example 4-2: Unsuccessful Update</u>	77
<u>Example 3-1: An Unanchored Interval</u>	28
<u>Example 3-2: Temporal Relationship</u>	30
<u>Example 3-3: Temporal Relationship with Gap</u>	30
<u>Example 3-4: Temporal Relationship without Gap</u>	30
<u>Example 3-5: Introducing the Sequence</u>	31
<u>Example 3-6: An Anchored Sequence</u>	32
<u>Example 3-7 State Change communication using Zero Duration Interval</u>	34
<u>Example 3-8 State Change communication using Event without Duration or End</u>	35
<u>Example 3-9: Use of an Attachment with inline XML artifact</u>	36
<u>Example 3-10: Use of an Attachment with external reference</u>	37
<u>Example 3-11: Interval with inline XML artifact and optional specified Performance</u>	40
<u>Example 3-12: Sequence with Performance defined in the Gluon</u>	45
<u>Example 3-13: Partition with Duration and Relationship defined in the Gluon</u>	49
<u>Example 3-14: Vavailability</u>	56
<u>Example 3-15 Gluon publishing availability of pre-existing sequence</u>	58
<u>Example 3-16: Standard Sequence with Ramp-Up and Ramp Down</u>	60

1 Introduction

2 The semantic information model of WS-Calendar is intended to be used by to define information payloads
3 for Web services and other serviceService-style interactions. [SOA-RM]. Placing these requirements in
4 context requires a brief overview of service requirements.

5 One of the most Agreement on when something should or did occur is fundamental components of to
6 negotiating service use. Negotiated services is agreeing when something should occur, and in auditing
7 when they did occur must be audited to understand timely performance. Short running services
8 traditionally have been handled as if they were instantaneous, and have handled scheduling through just-
9 in-time requests. Longer running processes, including physical processes, may require significant lead-
10 times. When multiple long-running services participate in the same business process, it may be more
11 important to negotiate a common completion time than a common start time. Pre-existing approaches that
12 rely on direct control of such services by a central system increases integration costs and reduce
13 interoperability as they require the controlling agent to know and manage multiple lead times.

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

20 Web services already coordinate a number of physical processes. Web services for building-based
21 systems include the standards [oBIX], BACnet/WS¹ LON-WS², OPC UA³, as well as a number of
22 proprietary systems. LON-WS⁴. The European research and advanced development project SIRENA
23 (Service Infrastructure for Real time Embedded Networked Applications) explored SOA for buildings,
24 factories and devices, including SODA (Service Oriented Device Architecture). SOA4D⁵ (Service-Oriented
25 Architecture for Devices) offers a collaborative open source development web platform, including
26 implementations ([SOAP] messaging, [WS-Management], [WS-Security], [DP-WSDPWS]) adapted to
27 the specific constraints of embedded devices. There is a growing interest in coordinating the activities of
28 things, building systems, industrial processes, homes, with human enterprise activities. In particular, if
29 building systems coordinate with the schedules of the building's occupants, they can reduce energy use
30 while improving performance.

31 An increasing number of specifications envision synchronization of processes through mechanisms
32 including broadcast scheduling. Efforts to build an intelligent power grid (or smart grid) rely on
33 coordinating processes in homes, offices, and industry with projected and actual power availability;

¹ BACnet® is a registered trademark of American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE).

² LON is a registered trademark of Echelon Corporation.

³ OPC UA is owned by the OPC Foundation

⁴ LON is a registered trademark of Echelon Corporation.

⁵ <http://cms.soa4d.org/>

mechanisms proposed include communicating different prices at different times. Several active OASIS Technical Committees require a common means to specify schedule and interval: Energy Interoperation [EITC] and Energy Market Information Exchange [EMIX]. Emergency management coordinators wish to inform geographic regions of future events, such as a projected tornado touchdown, using [EDXL]. The open Building Information Exchange specification [eBIX] lacks a common schedule communications for interaction with enterprise activities. These and other efforts would benefit from a common cross-domain, cross specification standard for communicating schedule and interval.

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

The intent of the WS-Calendar technical committee was to adapt the existing specifications for calendaring Technical Committee (TC) based its work upon the iCalendar specification as updated in 2009 (IETF [RFC5545]) and apply them to develop its the XML serialization [XCAL], currently (2011-05) on a standard for how schedule and event information is passed between and within services standards track in the IETF. The standards specification adopts the semantics and vocabulary of iCalendar for application to the completion of web-service contracts. WS-Calendar builds on work done and ongoing in The and inter-process interactions. Members of the Calendaring and Scheduling Consortium (CalConnect), which works to increase interoperability between calendaring systems.org developed both updates to IETF specifications and provided advice to this TC.

While this specification (WS-Calendar) defines the use of core semantic elements from iCalendar, no part of this document prevents is intended to prevent the use of other semantic elements from iCalendar from being used. WS-Calendar describes the minimal use of that standard, not the maximal.

Everything with the exception of all examples, all appendices, and the introduction is normative unless otherwise specifically noted.

1.1 Terminology

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

1.2 Normative References

64	Calendar Resource Schema	CalendarResource C. Joy, C. Daboo, M Douglas Douglass, Schema for representing resources for calendaring and scheduling services, http://tools.ietf.org/html/draft-cal-resource-schema-00 , (Internet-Draft), April http://tools.ietf.org/html/draft-cal-resource-schema-03 , (Internet-Draft), November 2010.
65	CalDAV	C. Daboo, B. Desruisseaux, L. Dusseault, <i>Calendaring Extensions to WebDAV (CalDAV)</i> , http://www.ietf.org/RFC/RFC4791.txt , IETF RFC4791, March 1997.
66	DPWS	T Nixon, A Regnier, D Driscoll, Antoine Mensch, <i>Devices Profile for Web Services 1.1</i> , http://docs.easis-open.org/ws-dd/ns/dpws/2009/01 , July 2009
67	FreeBusy-Read URL	E York. <i>Freebusy read URL</i> , http://www.calconnect.org/pubdocs/CD0903%20Freebusy%20Read%20URL%20V1.0.pdf , April 2009
68	REST	T Fielding, <i>Architectural Styles and the Design of Network-based Software Architectures</i> , http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm , Dissertation, 2000
69	ISO8601	ISO (International Organization for Standardization). <i>Representations of dates and times, third edition</i> , December 2004, (ISO 8601:2004)

83	RFC2119	S. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , http://www.ietf.org/RFC/RFC2119.txt , IETF RFC2119, March 1997.
84		
85		
86	RFC2616	R Fielding, et al. et al., <i>Hypertext Transfer Protocol -- HTTP/1.1</i> RFC2447 . Dawson, S. Mansour, S. Silverberg, <i>iCalendar Message-Based Interoperability Protocol (iMIP)</i> , http://www.ietf.org/RFC/RFC2247.txt , IETF RFC2447, December 2009.
87		
88		
89		
90	RFC2616	R Fielding, et al. et al., <i>Hypertext Transfer Protocol -- HTTP/1.1</i> http://tools.ietf.org/html/RFC2616 , IETF RFC2616, November 1998
91		
92		http://tools.ietf.org/html/RFC2616 , IETF RFC2616, draft standard, June, 1999
93	RFC3339	G Klyne, C Newman, <i>Date and Time on the Internet: Timestamps</i> http://tools.ietf.org/html/rfc3339 http://tools.ietf.org/html/rfc3339 , IETF RFC3339 proposed standard, July 2002
94		
95		
96	RFC4791	Daboo, et al. <i>Calendaring Extensions to WebDAV (CalDAV)</i> , http://www.ietf.org/rfc/rfc4791.txt , http://www.ietf.org/rfc/rfc4791.txt . IETF RFC 21194791, proposed standard, March 2007
97		
98		
99	RFC4918	L. Dusseault, <i>HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)</i> , http://tools.ietf.org/html/rfc4918
100		
101	RFC5545	B. Desruisseaux <i>Internet Calendaring and Scheduling Core Object Specification (iCalendar)</i> , http://www.ietf.org/rfc/rfc5545.txt http://www.ietf.org/rfc/rfc5545.txt , IETF RFC5545, proposed standard, September 2009.
102		
103		
104	RFC5546	C. Daboo <i>iCalendar Transport-Independent Interoperability Protocol (iTIP)</i> , http://www.ietf.org/rfc/rfc5546.txt http://www.ietf.org/rfc/rfc5546.txt , IETF RFC5546, proposed standard, December 2009.
105		
106		
107	RFC5598 RFC5988	M. Nottingham, <i>Web Linking</i> , http://www.ietf.org/rfc/rfc5598.txt , IETF RFC5598 http://tools.ietf.org/html/rfc5988 , IETF RFC5988, proposed standard, October 2010.
108		
109		
110	RFC6047	A. Melnikov, <i>iCalendar Message-Based Interoperability Protocol (iMIP)</i> , http://tools.ietf.org/html/rfc6047 , IETF RFC6047, December 2010.
111		
112	SOA-RM	OASIS Standard, <i>Reference Model for Service Oriented Architecture 1.0</i> , http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf , October 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf
113		
114		
115	SOAP	M Gudgin, M Hadley, N Mendelsohn, J Moreau, H Nielsen, A Karmarkar, SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation http://www.w3.org/TR/soap12-part1/ , http://www.w3.org/TR/soap12-part1/ , W3C Recommendation, April 2007
116		
117		
118		
119	Vailability	C. Daboo, B. Desruisseaux, <i>Calendar Availability</i> , http://tools.ietf.org/html/draft-daboo-calendar-availability-01 , IETF draft, November 2008 http://tools.ietf.org/html/draft-daboo-calendar-availability-02 , IETF Internet Draft, April 2011
120		
121		
122		
123	WSDL	E Christensen, F Curbera, G Meredith, S Weerawarana, <i>Web Services Description Language (WSDL) 1.1</i> , http://www.w3.org/TR/wsdl , W3C Note March 2001
124		
125		
126	WS-Management	DMTF Standard, <i>Web Services for Management</i> , http://www.dmtf.org/sites/default/files/standards/documents/DSP0226_1.0.0.pdf April 2010
127		
128		
129	WS-Security	K Lawrence, C Kaler, A Nadalin, R Monzillo, P Hallam-Baker, <i>Web Services Security: 4 SOAP Message Security 1.1</i> , http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf , February 2006
130		
131		
132		
133	draft-xCal	C. Daboo, M Douglas-Douglas, S Lees <i>xCal: The XML format for iCalendar</i> , http://tools.ietf.org/html/draft-daboo-et-al-icalendar-in-xml-07 , Internet Draft,
134		

135		October 2010 http://tools.ietf.org/html/draft-daboo-et-al-icalendar-in-xml-08 , IETF Internet-Draft, April 2011.
136		
137	XPATH	A Berglund, -S Boag, D Chamberlin, MFM Fernández, M Kay, J Robie, J Siméon, <i>XML Path Language (XPath) 2.0, (Second Edition)</i> , http://www.w3.org/TR/xpath20/ January 2007 http://www.w3.org/TR/xpath20/ , W3C Recommendation, December 2010.
138		
139	XLINK	S DeRose, E Maler, D Orchard, N Walsh, <i>XML Linking Language (XLink) Version 1.1</i> , http://www.w3.org/TR/xlink11/ http://www.w3.org/TR/xlink11/ May 2010.
140		
141	XPOINTER	S DeRose, E Maler, R Daniel Jr., <i>XPointer xpointer Scheme</i> , http://www.w3.org/TR/xptr-xpointer/ December 2002.
142		
143	XPTR	P Grosso, E Maler, J Marsh, N Walsh, <i>XPointer Framework</i> , http://www.w3.org/TR/xptr-framework/ W3C Recommendation, March 2003
144		
145	XML NAMES	P Grosso, E Maler, J Marsh, N Walsh, <i>XPointer element() Scheme</i> , http://www.w3.org/TR/xptr-element/ , W3C Recommendation March 2003
146		
147		
148		
149	XML SCHEMA	T Bray, D Hollander, A Layman, R Tobin, HS Thompson “Namespaces in XML 1.0 (Third Edition)“ http://www.w3.org/TR/xml-names/ W3C Recommendation, December 2009
150		
151		
152		
153		
154		
155	XRD	OASIS XRI Committee Draft 01, Extensible Resource Descriptor (XRD) Version 1.0, http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-ed01.pdf http://docs.oasis-open.org/xri/xrd/v1.0/cd01/xrd-1.0-ed01.pdf October 2009.
156		
157		
158		

1.3 Non-Normative References

159	DPWS	T Nixon, A Regnier, D Driscoll, Antoine Mensch, Devices Profile for Web Services 1.1 , http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01 , July 2009
160		
161		
162	NIST Framework	NIST Framework and Roadmap for Smart Grid Interoperability Standards , Office of the National Coordinator for Smart Grid Interoperability, Release 1.0, NIST Special Publication 1108, http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf
163		
164		
165		
166		
167		
168	NAESB Requirements	NAESB Smart Grid Requirements (awaiting publication) (draft contributed) http://lists.oasis-open.org/archives/ws-calendar-comment/201005/doc00000.doe http://lists.oasis-open.org/archives/ws-calendar-comment/201005/doc00000.doc , May 2010
169		
170		
171		
172	REST	T Fielding, Architectural Styles and the Design of Network-based Software Architectures, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm .
173		
174	RFC4918	L. Dusseault, HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV) , http://tools.ietf.org/html/rfc4918 IETF RFC 4918, proposed standard, June 2007
175		
176		
177	TZDB	P Eggert, A.D. Olson, "Sources for Time Zone and Daylight Saving Time Data", http://www.twinsun.com/tz/tz-link.htm http://www.twinsun.com/tz/tz-link.htm
178		
179	Time Zone Recommendations	CalConnect, CalConnect EDST (Extended Daylight Savings Time) Reflections and Recommendations, Version: 1.1, http://www.calconnect.org/pubdocs/CD0707%20CalConnect%20EDST%20Reflections%20and%20Recommendations%20V1.1.pdf http://www.calconnect.org/pubdocs/CD0707%20CalConnect%20EDST%20Reflections%20and%20Recommendations%20V1.1.pdf
180		
181		
182		
183		
184		
185		October 2010

186	Time Zone Service , M Douglas, C Daboo, Timezone Service Protocol, Draft RFC, IETF, http://datatracker.ietf.org/doc/draft-douglass-timezone-service/
187	
188	
189	WS-Management DMTF Standard, Web Services for Management, http://www.dmtf.org/sites/default/files/standards/documents/DSP0226_1.0.0.pdf April 2010
190	
191	
192	WS-Security K Lawrence, C Kaler, A Nadalin, R Monzillo, P Hallam-Baker, Web Services Security: 4 SOAP Message Security 1.1, http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf, February 2006
193	
194	
195	
196	WSDL R Chinnici, J Moreau, A Ryman, S Weerawarana, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language http://www.w3.org/TR/wsdl20/ , W3C Recommendation, June 2007. R Chinnici, H Haas, A Lewis, J Moreau, D Orchard, S Weerawarana, Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts, http://www.w3.org/TR/wsdl20-adjuncts/ , W3C Recommendation, June 2007.
197	
198	
199	
200	
201	

202 1.4 Contributions

203 The NIST Roadmap for Smart Grid Interoperability Standards [[NIST Framework](#)] requested that many
 204 standards development organizations (SDOs) and trade associations work together closely in
 205 unprecedented ways. An extraordinary number of groups came together and contributed effort, and time,
 206 requirements, and documents. The North American Energy Standards Board (NAESB) oversaw meetings
 207 with many representatives from every energy sector to contribute requirements to the TC. These
 208 meetings were presided over by Jonathan Booé to support the Roadmap's Priority Action Plan 04
 209 (PAP04), a common specification of time and schedule.

210 **[NAESB Smart Grid Standards Development Subcommittee:](#)**

211 The following documents are password protected. For information about obtaining access to these
 212 documents, please visit www.naesb.org or contact the NAESB office at (713) 356 0060.

213 **Wholesale** http://www.naesb.org/member_login_check.asp?doc=fa_2010_weq_api_6_b_ii.doc

214 **Retail** http://www.naesb.org/member_login_check.asp?doc=fa_2010_retail_api_9_b_ii.doc

215 1.4.1.5 Namespace

216 The XML namespace [[XML-ns](#)][\[XMLNAMES\]](#) URI that MUST be used by implementations of this
 217 specification is:

218 urn:ietf:params:xml:ns:icalendar-2.0
--

219 Table 1-1 lists the XML schemas that are used in this specification. The choice of any namespace prefix
 220 is arbitrary and not semantically significant.

221 *Table 1-1: Namespaces used in this specification*

Prefix	Namespace
xs	http://www.w3.org/2001/XMLSchema http://www.w3.org/2001/XMLSchema
xcal	urn:ietf:params:xml:ns:icalendar-2.0
ts	http://docs.oasis-open.org/ns/ws-calendar/timestamp/201103

222 The Resource Directory Description Language [[RDDL 2.0](#)[RDDL 2.0](#)] document that describes this
 223 namespace can be found at <http://docs.oasis-open.org/ns/ws-calendar>.<http://docs.oasis-open.org/ns/ws-calendar>. The normative schemas for WS-Calendar can be found linked from this namespace document.
 224 The schemas are listed in Table 1-2.

226 *Table 1-2: Schemas and Extensions Used in this Specification*

Schema	Description
iCalendar.xsd	Base Schema expressing core iCalendar information
iCalendar-params.xsd	Parameters used in iCalendar objects
iCalendar-props.xsd	Properties of iCalendar objects
iCalendar-valtypes.xsd	Values used by iCalendar
iCalendar-link-extension.xsd	Link extensions based on [web linking RFC5998] to define relationships between <code>components.Components</code> .
iCalendar-wscal-extensions.xsd	Extensions to iCalendar to support service functionality
iCalendar-bw-extensions.xsd	Extensions to support integration with Bedeworks server.
iCalendar-ms-extensions.xsd	Extensions to support integration with MS Exchange Server
TimeStamp.xsd	An ancillary information model describing the elements needed to support event forensics

227 Reviewers can find the schemas at <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csprd02/xsd/>-<http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/csd03/xsd/>.

229 **1.51.6 Naming Conventions**

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

231 For the names of elements and the names of attributes within XSD files, the names follow the lower camelCase convention, with all names starting with a lower case letter. For example,

```
233      <element name="componentType" type="energyinterop:ComponentType"/>
```

234 For the names of types within XSD files, the names follow the lower CamelCase convention with all names starting with a lower case letter prefixed by "type-". For example,

```
236      <complexType name="type-componentService">
```

237 For the names of intents, the names follow the lower camelCase convention, with all names starting with a lower case letter, EXCEPT for cases where the intent represents an established acronym, in which case the entire name is in upper case.

240 An example of an intent that is an acronym is the "SOAP" intent.

241 **1.61.7 Editing Conventions**

242 For readability, element names in tables appear as separate words. The actual names are lowerCamelCase, as specified above, and as they appear in the XML schemas.

244 All elements in the tables not marked as "optional" are mandatory.

245 Information in the "Specification" column of the tables is normative. Information appearing in the note column is explanatory and non-normative.

247 All sections explicitly noted as examples are informational and are not to be considered normative.

248 **1.71.8 Architectural References**

249 WS-Calendar assumes incorporation into services. Accordingly it assumes a certain amount of definitions of roles, names, and interaction patterns. This document relies heavily on roles and interactions as defined in the OASIS Standard Reference Model for Service Oriented Architecture [[SOA-RM](#)].

252 **1.81.9 Semantics**

253 Certain terms appear throughout this document, some with extensive definitions. [The table](#)Table 1-3
 254 provides [summary](#)definitions for the convenience of the reader and reviewer. [When full definitions of the](#)
 255 [Many terms below appear require fuller discussion than is in this section, and are discussed in greater](#)
 256 [depth in later sections of this document, with the exception of in the appendices, then that later.](#) In all
 257 [cases, the normative actual definition is normative the one in this section.](#)

258 WS-Calendar terminology begins with a specialized terminology for the segments of time, and for groups
 259 of related segments of time. These terms are defined in Table 1-3 through Table 1-6 below.

260 *Table 1-3: Semantics: Foundational Elements*

Time Segment	Definition
Component	In iCalendar, the primary information structure is a Component. Intervals and Gluons are new Components defined in this specification.
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 VTTODO object that is mandatory in the Interval.
Interval	The Interval is a single duration Duration derived from the common calendar components Components as defined in iCalendar ([RFC5545]) and refined in [XCAL]. In Calendar systems, it]. An Interval is processed as part of a vtodo, but the constraints and conformanceSequence. An entire Sequence can be scheduled by scheduling a single Interval in that sequence. For this reason, Intervals are different defined through Duration rather than through dtStart or dtEnd.
Sequence	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. A Sequence may optionally include a Lineage. A Sequence can be scheduled multiple times through repeated reference by different Gluons. Intervals are defined through their Duration, and the schedule, dtEnd or dtStart, is applied to the Sequence as a whole.
Partition	A Partition is a set of consecutive Intervals. The Partition includes the trivial case of a single Interval. Partitions are used to define a single service or behavior that varies over time. Examples include energy prices over time and energy usage over time.
Gluon	A gluon influences the serialization of Intervals in a Sequence, through inheritance and through schedule setting. The Gluon is similar to the Interval, but has no service or schedule effects until applied to an Interval or Sequence.
Artifact	An Artifact is the thing that occurs during an Interval. WS-Calendar extends the [XCAL] attach object to contain this uses the Artifact as a placeholder. The contents of the Artifact are not specified in WS-Calendar; rather the Artifact provides an extension base for the use of WS-Calendar in other specifications. Artifacts may inherit elements as do Intervals within a Sequence.

261 WS-Calendar works with groups of Intervals that have relationships between them. These relations
 262 constrain the final instantiation of a schedule-based service. Relations can control the ordering of

263 Intervals in a Sequence. They can describe when a service can be, or is prevented from, being invoked.
264 They establish the parameters for how information will be shared between elements using Inheritance.
265 The terminology for these relationships is defined in Table 1-4.

266 *Table 1-4: Semantics: Relations, Limits, and Constraints*

Term	Definition
Link	The Link is used by one WS-Calendar object to reference another. A link can reference either an internal object, within the same calendar, or an external object in a remote system.
Relationship	Relationships link between components Components for Binding. ICalendar defines several relationships, but WS-Calendar uses only the CHILD relationship, and that only to bind Gluons to each other and to Intervals.
Temporal Relationship	Temporal Relationships extend the [RFC5545] Relationships to define how Intervals become a Sequence by creating an order between Intervals. The Predecessor Interval includes a Temporal Relation, which references the Successor Interval. When the start time and duration Duration of one Interval is known, the start time of the others can be computed through applying Temporal Relations.
Availability	Availability expresses the range of times in which an Interval or Sequence can be Scheduled. Availability is often overlays or is overlaid by Busy. Availability can be Inherited.
Busy	Busy expresses the range of times in which an Interval or Sequence cannot be Scheduled. Busy is often used to overlay or be overlays is overlaid by Availability. Busy can be Inherited.
Child, Children	The CHILD relationship type (rel_type) defines a logical link (via URI or UID) from parent object to a child object. A Child object is the target of one or more CHILD relationships and may have zero one to many Parent objects.
Parent [Gluon]	A Gluon (in a Sequence) that includes a CHILD relationship parameter type (rel_type) defines a logical link (via URI or UID) from parent object to a child object. A Parent Component contains one or more CHILD Relationships

267 WS-Calendar describes how to modify and complete the specification of Sequences. WS-Calendar calls
268 this process Inheritance and specifies a number of rules that govern inheritance. Table 1-5 defines the
269 terms used to describe inheritance.

270 *Table 1-5: Semantics: Inheritance*

Term	Definition
Lineage	The ordered set of Parents that results in a given inheritance or execution context for a Sequence.
Inheritance	Parents bequeath information to Children that inherit them. If a child does not already possess that information, then it accepts the inheritance. WS-Calendar specifies rules whereby information specified in one informational object is considered present in another that is itself lacking expression of that information. This information is termed the Inheritance of that object.
Bequeath	A Parent Bequeaths attributes (Inheritance) to its Children.
Inherit	A Child Inherits attributes (Inheritance) from its Parent.

Term	Definition
Covarying Attributes	Some attributes are inherited as a group. If any member of that group is expressed in a Child, all members of that group are deemed expressed in that Child, albeit some may be default values. These characteristics are called covarying or covariant. A parent bequeaths covarying characteristics as a group and a child accepts or refuses them as a group.
Decouplable Attributes	Antonym for Covarying Attributes. Decouplable Attributes can be inherited separately.

271 As Intervals are processed, as Intervals are assembled, and as inheritance is processed, the information
 272 conveyed about each element changes. When WS-Calendar is used to describe a business process or
 273 service, it may pass through several stages in which the information is not yet complete or actionable, but
 274 is still a conforming expression of time and Sequence. Table 1-6 defines the terms used when discussing
 275 the processing or processability of Intervals and Sequences.

276 During the life-cycle of communications concerning Intervals, different information may be available or
 277 required. For service performance, Start Duration and the Attachment Payload must be complete. These
 278 may not be available or required during service advertisement or other pre-execution processes. Table
 279 1-6 defines the language used to discuss how the information in an Interval is completed.

280 *Table 1-6: Semantics: Describing Intervals*

Term	Definition
<u>Anchored</u> <u>Designated</u> <u>Interval</u>	An Interval is Anchored [in time] if it is Bound to a full date and time. A Sequence or Partition is Anchored if it contains an Anchored Interval, and when Fully Bound, the specific date, time, and duration of all Intervals can be determined unambiguously. Specific performance of a Service Contract always occurs in an Anchored Sequence. An Interval that is referenced by a Gluon is the Designated Interval for a Series. An Interval can be Designated and still not Anchored.
<u>Partially</u> <u>Anchored</u>	An Interval is Partially Anchored if EITHER its Date OR its Time is Bound when it includes a Start or End, either directly or through Binding. A Sequence or Partition is Partially Anchored if when its Designated Interval is Partially Anchored.
<u>Unanchored</u>	An Interval is Unanchored if NEITHER its Begin Date nor its Begin Time are known. An Interval is Unanchored when it includes neither a Start or an End, either internally, or through Binding. A Sequence is Unanchored if its Designated Interval Unanchored. Note: a Sequence that is re-used may be Unanchored in one context even while it is Anchored in another.
<u>Binding</u>	Binding is the application of information to an Interval or Gluon, information derived through Inheritance or through Temporal Assignment.
<u>Bound Element</u>	A Bound Element refers to an Element and its Value after Binding, e.g., a Bound Duration.
<u>Bound</u> <u>Interval</u>	As in mathematical logic where a metasyntactic variable is called "bound", an Interval, Sequence, or Partition is said to be Bound when the values necessary to execute it (as a service) are completely filled in. A Bound Interval refers to an Interval and the values of its Elements after Binding.

Term	Definition
Partially Bound Sequence	A Partially Bound Interval is one that is still not Bound after receiving its Inheritance. A Sequences or Partitions is Partially Bound if it contains at least one Interval that is Partially Bound. A Bound Sequence refers to a Sequence and the values of its Intervals after Binding.
Partially Bound Unbound	An Unbound Partially Bound refers to an Interval or a Sequence which is not itself yet complete, but must still receive inheritance to following Binding, i.e., the processes cannot yet be fully specified. A Sequences or Partitions is Unbound if it contains at least one Interval that is Unbound.
Fully Bound	Fully Bound refers to an Interval or Sequence that is complete after Binding, i.e., the process can be unambiguously executed when Anchored.
Unbound Fully Bound	A synonym for Bound An Unbound Interval or Sequence is not itself complete, but must still receive inheritance to be fully specified. A Sequence or Partition is Unbound if it contains at least one Interval that is Unbound.
Constrained	An Interval is Constrained if it is not Anchored and it is bound to one or more Availability or Free/Busy elements
Temporal Assignment	Temporal Assignment determines the start times of Intervals in a Sequence through processing of their Durations and Temporal Relations.
Scheduled	A Sequence or Partition is said to be Scheduled when it is Anchored, Fully Bound, and service performance has been requested.
Unscheduled	An Interval is Unscheduled if its neither its begin date and time it is not Anchored, nor its end date and time have been set. A Sequence or Partition is Unscheduled if none of its Intervals, after when Fully Bound, is Scheduled.
Designated Interval	In a Sequence the Designated Interval is either (a) (if there are no Gluons related to the Sequence) one of the Earliest Interval(s), or (b) (if there is at least one Gluon related to the Sequence) the single Interval referenced by a Gluon as CHILD.
Predecessor Interval	A Predecessor Interval includes a Temporal Relation which references a Successor Interval.
Successor Interval	A Successor Interval is one referred to by a Temporal Relationship in a Predecessor Interval.
Antecedent Interval(s)	An Antecedents are an Interval or set of Intervals that precede a given Interval within the same Sequence
Earliest Interval	The set of Intervals at the earliest time in a given Sequence
Composed Interval	A Composed Interval is the virtual Interval specified by applying inheritance through the entire lineage and into the Sequence in accord with the inheritance rules. A Composed Interval may be Bound, Partially Bound, or Unbound.
Composed Sequence	A Composed Sequence is the virtual Sequence specified by applying inheritance through the entire lineage and into the Sequence in accord with the inheritance rules. A Composed Sequence may be Bound, Partially Bound, or Unbound.

Term	Definition
Comparable Sequences	Two Sequences are Comparable if and only if there exists a ^a the Composed version of each that defines the same schedule.

281

2 Overview of WS-Calendar

282 A calendar communication without a real world effect⁶ is of little interest. That real world effect is the result
 283 of a service execution context within a policy context. Practitioners can use WS-Calendar to add
 284 communication of schedule and Interval to the execution context of a service. Use of WS-Calendar will
 285 align the performance expectations between execution contexts in different domains. The Technical
 286 Committee intends for other specifications and standards to normatively reference and claim
 287 conformance to WS-Calendar, bringing a common scheduling context to diverse interactions in different
 288 domains

289 2.1 Approach taken by the WS-Calendar Technical Committee

290 The Technical Committee (TC) based its work upon the iCalendar specification as updated in 2009 (IETF
 291 [[RFC5545](#)] and its the XML serialization [[XCAL](#)], currently ([2010-07-2011-05](#)) on a standards track in the
 292 IETF. Members of the Calendaring and Scheduling Consortium (CalConnect.org) developed both updates
 293 to IETF specifications and provided advice to this TC. [This work](#)[[RFC5545](#)] provides the [normative](#)
 294 vocabulary for use in this specification.

295 This committee developed the normative schema (XSD) for iCalendar. This schema, including the
 296 schema extensions necessary for the services defined herein, is part of the WS-Calendar specification.

297 The committee solicited requirements from a range of interests, notably the NIST Smart Grid Roadmap
 298 [[NIST Framework](#)] and the requirements of the Smart Grid Interoperability Panel (SGIP) as developed by
 299 the North American Energy Standards Board (NAESB). [[NAESB Requirements](#)]. Others submitting
 300 requirements included members of the oBIX technical committee and [representative](#)[representatives](#) of
 301 the FIX Protocol Association. These requirements are reflected in the semantic elements described in
 302 Chapters 3 and 4.

303 In a parallel effort, the CalConnect TC-XML committee developed a number of schedule and calendar-
 304 related services. CalConnect drew on its experience in interoperability between enterprise calendaring
 305 systems as well as interactions with web-based calendars and personal digital assistants (PDAs). These
 306 services were developed as RESTfull (using [[REST](#)]) services by CalConnect and contributed to the WS-
 307 Calendar TC. CalConnect also developed and contributed [[SOAP](#)] and [[WSDL](#)] definitions to this TC.

308 2.2 Scheduling Communicating Schedules and Service Performance

309 Time semantics are critical to [WS-Calendar](#).[process interactions](#). Services requested differently can have
 310 different effects on performance even though they appear to request the same time interval. This is
 311 inherent in the concept of a service-oriented architecture.

⁶ [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.](#)

312 | As defined in the OASIS Reference Model for Service Oriented Architecture 1.0⁷, [SOA-RM], service
313 requests access the capability of a remote system.

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

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

322 | A request for remote service performance is a request for specific real world effects. For WS-
323 Calendarprocess interaction, these effects are expected to occur during a given period. Consider two
324 service providers that offer the same service. One must start planning an hour or more in advance. The
325 second may be able to achieve the service in five minutes. The service start time is the time when that
326 service becomes fully available; that is the time specified in service interactions. Because this service
327 start time and service period are all that matters, the same service can be offered by different providers
328 using quite different technologies.

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

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

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

337 | 2.2.1 Which Time? UTC vs. Local Time

338 | Coordinated Universal Time (abbreviated UTC) is a time standard based on International Atomic Time
339 (TAI) with leap seconds added at irregular intervals to compensate for the Earth's slowing rotation. Time
340 zones around the world can be expressed as positive or negative offsets from UTC.

341 | When 2 or more parties attempt to agree on a time, e.g., for a meeting, or when to provide a service, they
342 agree to start at a particular instant of time UTC. They agree on that instant in time by converting from
343 local time, e.g., they want a meeting to start at 13:00 Eastern, 18:00 UK. Our lives and the use of services
344 | are bound by local time not by UTC. To humans Experientially, local time is the invariant and UTC is
345 mapped on to it. If a government modifies the rules we adjust the mappings and we shift the UTC time.
346 We still want to meet at 13:00 local or have the heating start at 07:00.

347 | As long as the rules never change this causes no confusion—but they do. Recent experience has
348 included considerable efforts when the rules for the start of Daylight Savings Saving Time (DST) have
349 changed. If all information is in UTC, and no record of the event's basis in the local time and time

⁷ See normative references in section 1.2

350 zone remains, there is no way to re-compute existing contracts. ~~We don't~~ It is often necessary to know if
351 ~~that~~ UTC was calculated based on an old or a new rule.

352 A triplet of Local time + timezoneid + (UTC or offset) always allows ~~you to determine~~ the determination if
353 ~~the~~ time is valid. If a recalculation of UTC for that local time + tzid results in a different value from that
354 stored then presumably the DST rules have changed since the data was stored. If you can detect that
355 the scheduled time is no longer valid you, one can take corrective action.

356 ~~For simplicity, all examples and discussion in this document are based on Greenwich Mean Time also
357 known as Coordinated Universal Time (UTC).~~ The Technical Committee makes no representation as
358 whether UTC or local time are more appropriate for a given interaction. Because WS-Calendar is based
359 on [iCalendar], business practices built upon WS-Calendar can support either. Specifications that claim
360 conformance with this specification may require choices to support their particular business processes.

361 ~~Practitioners should~~ For a fuller discussion of time zones, consult [**Time Service Recommendations**] and
362 [**Time Zone Service**] in the non-normative references.

363 2.3 Overview of This Document

364 The specification consists of a standard schema and semantics for schedule and interval information.
365 Often the most important service schedule communications involve series of related services over time,
366 which WS-Calendar defines as a Sequence. These semantic elements are defined and discussed in
367 Section 3. While this specification describes only the use of core semantic elements from iCalendar, no
368 part of this document prevents other semantic elements from iCalendar from also being used.

369 Section 3.2 introduces notions of tolerance, i.e. what does it mean to be "on time". This section also
370 describes the different ways to associate a service request with each Interval in a Sequence.

371 Managing information exchanges about a Sequence of events can easily become cumbersome, or prone
372 to error. WS-Calendar defines the Calendar Gluon, a mechanism for making assertions about all or most
373 of the Intervals in a Sequence. Intervals can inherit from a Calendar Gluon, or they can override locally
374 assertions inherited from the Calendar Gluon. Section 3.3 discusses inheritance and parsimony of
375 communication and introduces contract scheduling.

376 ~~In Sections 4-4.9, this document describes [REST]-based, (RESTfull) web services for interacting with
377 remote calendars. These interactions are derived from the well-known interactions defined in [CalDAV],
378 although they do not specify any interaction with [CalDAV] servers. This specification defines services for
379 calendar inquiries, event scheduling, event updating, and event cancellation.~~

380 ~~In Sections n-n, this document describes [SOAP]-based interactions for Calendar services. As with
381 REST, the specification defines services for calendar inquiries, event scheduling, event updating, and
382 event cancellation using the iCalendar schema.~~

383 ~~With incompatible communications defined (REST, SOAP), the specification is not prescriptive of the
384 communications used. The practitioner must decide whether to use eitherone or the other of these
385 communication protocols, or whether WS-Calendar artifacts are better used when embedded within other
386 messages. These decisions, along with decisions about the specific security needed by the
387 communication must be based upon the specific application and message content. Specifications that
388 claim conformance to this specification may wish to provide guidance appropriate for the business
389 purposes of that specification.~~

390 2.4 Security Considerations

391 Part 1 describes an information model. The information models can be expressed in any interaction,
392 using any protocol. There are no security aspects of the information model.

393 Specifications which claim conformance with WS-Calendar may wish to specify security approaches or
394 techniques. Security choices must be based on the business requirements and operational risks of the
395 interaction that those specifications define. As this specification defines a general information model, for
396 use in many interactions, it specifies no security approach.

397 **3 PART ONE: Semantic Model**Information model for
398 WS-Calendar

399 **3.1 Intervals, Temporal Relations, and Sequences**

400 WS-Calendar Elements are semantic elements derived from the [XCAL] specification. ~~This set of~~
401 elements ~~are~~ smaller than ~~those needed for~~ full schedule interaction, and describe the intervals,
402 durations~~Intervals, Durations~~, and time-related events that are relevant to service interactions. ~~The WS-~~
403 ~~Calendar uses the~~ elements ~~are used~~ to build a precise vocabulary of time, ~~duration Duration~~, Sequence,
404 and ~~schedule Schedule~~.

405 ~~WS-Calendar elements adapt the iCalendar objects to make interaction requirements explicit. For~~
406 ~~example, in human schedule interactions, different organizations have their own expectations. Meetings~~
407 ~~may start on the hour or within 5 minutes of the hour. As agents scheduled in those organizations, people~~
408 ~~learn the expected precision. WS-Calendar elements adapt the iCalendar objects to make interaction~~
409 ~~requirements explicit. For example, in This human schedule interactions, different organizations have their~~
410 ~~own expectations. Meetings may start on the hour or within 5 minutes of the hour. As agents scheduled in~~
411 ~~these organizations, people learn the expected precision. In WS-Calendar, that precision expectation~~
412 ~~must be explicit to prevent interoperation problems. WS-CalendarThis specification~~ defines a
413 performance element to elaborate the simple specification of [XCAL] to make explicit the performance
414 expectations within a scheduled event.

415 ~~WS-Calendar~~This specification defines common semantics for recording and exchanging event
416 information.(Time Stamps).

417 **3.1.1 Core Semantics derived from [XCAL]**

418 **3.1.1.1 The iCalendar data format [RFC5545] is a widely deployed**
419 **interchange format for calendaring and Core Semantics derived from**
420 **[XCAL]**

421 ~~The iCalendar data format [RFC5545] is a widely deployed interchange format for calendaring and~~
422 ~~scheduling~~schedule data. The [XCAL] specification ~~(in process)~~ standardizes the XML representation of
423 iCalendar information. WS-Calendar relies on [XCAL] standards and data representation to develop its
424 semantic ~~components~~Components.

425 **3.1.1.1 Time**

426 **3.1.1.1 Time is an ISO 8601 compliant time**

427 ~~[ISO8601] defines string formats for the optional accompaniment expression of a date, time, and~~
428 ~~duration interval. [ISO8601] also defines string formats to define times to express the passage of less than 1~~
429 ~~second time, herein a Duration. This specification relies extensively on [ISO8601]. Examples of date and~~
430 ~~time representations the from the ISO 8601 standard include:~~

431 Year:
432 YYYY (eg 1997)
433 Year and month:
434 YYYY-MM (eg 1997-07)
435 Complete date:
436 YYYY-MM-DD (eg 1997-07-16)
437 Complete date plus hours and minutes:
438 YYYY-MM-DDThh:mmTZD (eg 1997-07-16T19:20+01:00)
439 Complete date plus hours, minutes and seconds:

440 YYYY-MM-DDThh:mm:ssTZD (eg 1997-07-16T19:20:30+01:00)
441 Complete date plus hours, minutes, seconds and a decimal fraction of a second
442 YYYY-MM-DDThh:mm:ss.sTZD (eg 1997-07-16T19:20:30.45+01:00)

443 ~~Normative information on [ISO 8601] is found in section 1.2.~~

444 ~~This specification is general purpose. Standards that claim conformance to this specification may need to~~
445 ~~restrict the variability above to improve interoperation within their own interactions.~~

446 3.1.1.2 The iCalendar Components (VComponents)

447 iCalendar and [XCAL] have a number of long defined ~~component~~Component objects that comprise the
448 payload inside of an iCalendar message. These include the VTODO, the VALARM, the VEVENT. (The "v"
449 that begins each element name is there for historic ~~purposes~~reasons.) The definitions and use of each of
450 the ~~vObject~~vComponents can be found in [RFC5545].

451 The ~~vObject~~vComponents share the same parameters and properties. The distinctions between these
452 informational ~~objects~~ is in ~~which~~types are ~~permitted~~ones of purpose and ~~which~~ are required. ~~Because of~~
453 ~~its flexibility, conformance. The Interval and Gluon are new vComponents; each is derived from the~~
454 ~~VTODO object is same base type as~~ the basis other vComponents.

455 This specification in no way deprecates the pre-existing vComponents. The new components are
456 introduced to support stored sequences of operations and remote invocation. The existing vComponents
457 are extended to support informational payloads for WS-Calendar objects for service performance.
458 Because WS-Calendar services process interaction. A conforming specification can use both old and new
459 vComponents where each makes sense.

460 The RESTful and SOAP in Parts Two and Three of a future version of this specification support all
461 traditional vComponents as well as the new ones defined here. Conforming information elements MAY be
462 processed using traditional iCalendar-based interactions (CalDAV, et al.), all VComponents SHALL be
463 supported.) and managed in traditional iCalendar stores.

464 The Interval and Gluon are new vObjects, and each is derived from vtodo.

465 3.1.1.3 Duration and the granularity~~Granularity~~ of Time

466 This specification uses Duration as defined in [ISO 8601] as a data-type throughout. iCalendar makes a
467 number of assumptions about the meaning of time when expressed as duration, based on guidance in
468 [ISO 8601]. These become, i.e., a duration is over when the same common metric is reached in the next
469 such unit. For example, a duration of one day starting at 6:00 AM lasts until 6:00 AM the next day. This
470 becomes important during times periods when the meaning of a duration changes. The passage of a
471 month that begins on January 5 is complete on February 5. Another month comes to March 5. Each is
472 expressed using the format (1M)."P1M". These durations are, respectively, 31, 28 or 29, and 31 days. In
473 a similar way, Years (1Y)"P1Y" may be 365 or 366 days long, days (1D)"P1D" may be 23, 24, or 25 hours
474 long. A duration is over, when the same common metric is reached in the next such unit

475 The meaning of a communication is based upon the granularity of the communication. If the intention is to
476 express 30 days, then one should use (30D)"P30D" and not ("P1M"). Similarly, if the intent is to express
477 from now until the same time tomorrow, use (1D)"P1D" rather than 24 hours (24H)."PT24H".

478 3.1.2 Intervals

479 Time Segments, i.e., increments of Clear communication of the continuous passage of time, are a is
480 critical component of to defining service alignment using WS-Calendar. There are many overloaded uses
481 of terms about time, and within a particular time segment, there may be many of them. coordination.

482 The building block for the WS-Calendar this information model is the Interval. The Interval is a time
483 segment whose length is specified by the a Duration. The Duration is represented by a string as defined
484 in the iCalendar specification [RFC5545]. The Committee listened to arguments that we should redefine
485 the use and meaning of Duration. Whatever their merit, the iCalendar Duration has a pre-existing
486 meaning of the length of time of scheduled within an event.

517 An Interval is a unit of time, and can be bound to service delivery, and can be bound to
 518 time. An Unscheduled Interval is not linked to a specific date and time. A Scheduled
 519 Interval has a known specified start date and time. Intervals can legally contain all elements of the
 520 VTOD properties as defined in [RFC5545]. For convenience, the elements essential to coordinating
 521 service operations using Intervals are listed in Table 3-1.

522 An Interval is part of a Sequence. An entire Sequence can be scheduled by scheduling a single Interval in
 523 a Sequence. A single Sequence can be scheduled multiple times through repeated reference by different
 524 Gluons. It may be useful to consider the Unanchored Sequence as a process subroutine and that a Gluon
 525 can be used to invoke that subroutine. For this reason, of the three primary temporal elements (dtStart,
 526 dtEnd, and Duration) in a Component, the Duration has primacy in Intervals. Within a Sequence, a
 527 maximum of a single Interval MAY have a dtStart or a dtEnd.

528 Nothing in this section supersedes [RFC5545]. Implementers SHALL refer to those respective
 529 specifications [RFC5545] and the [XCAL] specifications for the normative description of each element.
 530 with the exception of Duration, which is as defined as in [ISO8601].

531 Table 3-1: Properties Elements of Intervals

Elements	Description	Use	Use in WS-Calendar
Dtstamp	xcal:dtstamp Mandatory	Identifies when Interval object was created	Deleted
Uid	Mandatory	Used to enable unambiguous referencing by other components	Deleted
Duration	xcal:Duration Optional	Identifies length of time for Interval. Duration must be known before an Interval can be transacted, but the Duration may only come through Binding.	
DtStart	Xcal:dateTime Optional	Scheduled start date and time for Interval. The Start must be known before an Interval can be transacted, but the Duration may only come through Binding.	
Attach	Mandatory	In [XCAL], any attachment. In WS-Calendar, the Attach contains the informational payload used by incorporating conforming specifications. Defined in See section 3.2.	
Relations	As defined in [RFC5998] Optional	Relations contain the temporal relations between Intervals that create Sequences. Section 0. describes Temporal Relations and their use.	

532 An Interval specifies how long an activity lasts. An Unanchored Interval is not linked to a specific date and
 533 time. The example below (Example 3-1) shows the components section of a WS-Calendar-
 534 based message containing a single Unanchored Interval, i.e., it contains neither a dtStart nor a dtEnd.
 535 Note that there is no Relationship; there is no need for Relationships until an Interval is incorporated into
 536 a Sequence.

537 Example 3-1: An Unanchored Interval

```

538 <xcal:interval xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
539   <xcal:properties>
540     <xcal:uid>
541       <xcal:text>6fa8b9e5-e9b1-4ba1-bf9e-
542       5e5da03eb943@examples.easis-open.org</xcal:text>
543       6a2f12639c7e</xcal:text>
544     </xcal:uid>
545     <xcal:duration>
546       <xcal:duration>T10HPT10H</xcal:duration>

```

```

547           </xcal:duration>
548       </xcal:properties>
549   <xcal:components/>
550 </xcal:interval>

```

551 Note that no start time is specified, and no relationship. Relationships are not needed until an Interval is
 552 incorporated into a Sequence.

553 3.1.3 Connecting the Intervals

554 Many iCalendar communications involve more than one Interval. Classic iCalendar [RFC5545] defines
 555 relationships internally. WS-Calendar instead [xCAL] uses the extensible expression pattern of Web Link
 556 [RFC5998], both for links (as described in [RFC5588]) to express the traditional Relationships (parent,
 557 child, sibling) iCalendar relationships PARENT, CHILD, and for the SIBLING. This specification extends
 558 these relationships by adding Temporal Relationships. Relationships include Relations. Temporal
 559 Relations consist of a reference, a relation, and optional Tolerance parameters a Gap that specifies any
 560 Duration between Predecessor and Successor.

561 Temporal Relationships, new in WS-Calendar, use Web Linking [RFC5998] in an Interval (the
 562 Predecessor) to reference another Interval (the Successor). Temporal Relationships optionally include a
 563 Gap that specifies any lag between Predecessor and Successor.

564 Unlike most semantic elements in this specification, Temporal Relations are defined in this specification,
 565 rather than defined elsewhere and used herein.

566 Table 3-2: Temporal Relationships

Temporal Relationship	Short Form	Definition	Example
<u>finishToStartGap</u>	FS	As soon as Duration indicating the time between the predecessor Interval finishes, and the successor Interval starts. Optional, where missing, Gap is treated as a zero duration	When sanding is complete, painting begins. Gap may be positive or negative. In the examples below, the Gap, when present, is 20 minutes.
<u>finishToFinishFinish To Start</u>	FFFS	The successor Interval continues as long As soon as the predecessor Interval finishes, the successor Interval starts.	The concession stand stops serving 20 minutes after the end of the game. When sanding is complete, painting begins.
<u>startToFinishFinish To Finish</u>	SFFF	The start of successor Interval continues as long as the predecessor controls the finish of the successor Interval.	The start of Attendee Check-in controls concession stand stops serving 20 minutes after the end of the Interval "Set up registration booth." game.
<u>startToStartStart To Finish</u>	SSSF	The Predecessor Interval triggers the start of the second task. The Gap indicates predecessor controls the finish of the lag times successor.	The start of Attendee Check-in controls the end of the Interval "Set up registration booth." 20 minutes after the caterer begins work, the dining lines are open.

GapStart To Start	SS	Duration indicating the time between the predecessor and the successor. Optional, where missing, Gap is treated as a zero duration. The Predecessor Interval triggers the start of the second task.	20 minutes after the caterer begins work, the dining lines are open. Gap may be positive or negative.
--------------------------	----	---	---

567 While simple relationships may be ordered based on which task occurs first (finishToStart), if a later
 568 Interval is controlling, other choices may make more useful. For example, if ramp-up time must be
 569 completed before run-time, and run-time start is indicated in a contract, it may be useful to specify that the
 570 Ramp Interval (Successor) must complete before (startToFinish) the Designated Interval's (Predecessor)
 571 scheduled start time. [Referencing specifications](#) [Specifications claiming conformance](#) should consider
 572 [statements of](#) conformance around Temporal Relationships.

573 The relationship below indicates that this Interval is to start ten minutes following the finish of the Interval
 574 specified.

575 *Example 3-2: Temporal Relationship*

```
576 <xcal:related-to xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
577   <xcal:parameters>
578     <xcal:reltype>
579       <xcal:text>FS</xcal:text>
580     </xcal:reltype>
581     <xcal:gap>
582       <xcal:duration xs:type="xcal:DurationPropType">
583         <xcal:parameters/>
584         <xcal:duration>T10M</xcal:duration>
585       <!-->PT10M</xcal:duration>
586     </xcal:gap>
587   </xcal:parameters>
588   <xcal:uid>05782926-1d71-4a55-ac3b-cba5ebf419d3@examples.oasis-
589 open.org<07fb177d-54ea-44ea-8ef5-5b763dc9f0c6</xcal:uid>
590 </xcal:related-to>
```

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

594 *Example 3-3: Temporal Relationship with Gap*

```
595 <xcal:related-to xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
596   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
597   <xcal:related-to>
598     <xcal:parameters>
599       <xcal:reltype>
600         <xcal:text>FS</xcal:text>
601       </xcal:reltype>
602       <xcal:gap>
603         <xcal:duration xs:type="xcal:DurationPropType">
604           <xcal:parameters/>
605           <xcal:duration>T0M</xcal:duration>
606         <!-->PT10M</xcal:duration>
607       </xcal:gap>
608     </xcal:parameters>
609     <xcal:uid>5deedb30-7278-4e96-9f81-c20c81f283c3@examples.oasis-
610 open.org<07fb177d-54ea-44ea-8ef5-5b763dc9f0c6</xcal:uid>
611 </xcal:related-to>
```

612 Leaving out the optional Gap element, we have:

613 *Example 3-4: Temporal Relationship without Gap*

```

614 <xcal:related-to xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
615   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
616   <xcal:related-to>
617     <xcal:parameters>
618       <xcal:reltype>
619         <xcal:text>FS</xcal:text>
620       </xcal:reltype>
621     </xcal:parameters>
622     <xcal:uid>5deedb30-7278-4e96-9f81-c20c81f283c3@examples.oasis-
623 open.org07fb177d-54ea-44ea-8ef5-5b763dc9f0c6</xcal:uid>
624   </xcal:related-to>

```

625 The two expressions of a Temporal Relationship abovein Example 3-3 and Example 3-4 are equivalent.
626 Intervals with Temporal Relationships enable the message to express complex temporal relations to form
627 a Sequence, as well as express the simple. A Sequence consisting of identical consecutive Intervals is
628 named a Partition
629 A Sequence describes a coherent set of Intervals that can be assembled from a collection of Intervals.
630 As the rules for parsing XML do not mandate preservation of order within a sub-set, we cannot assume
631 that order is preserved when parsing a set of Components. Intervals. For Sequences in WS-Calendar,
632 then, mere order is not enough—a Sequence is a collection of Intervals each of which Interval either
633 refers to or is referred by at least one Interval. Using the references, expressed as Temporal Relations,
634 WS-Calendar describes a single coherent Sequence that is assembled from a set of Intervals in a
635 collection.

636 3.1.4 Sequences: Combining Intervals

637 A Sequence is a collection of Intervals with a coherent set of Temporal Relationships. (Table 1-3).
638 Temporal Relationships are transitive, so that if Interval A is related to Interval B, and Interval B is related
639 to Interval C, then Interval A is related to Interval C. Sequences can also include Gluons (see section
640 3.3.1, References and Inheritance. 3.3.1, References and Inheritance., but for this section, we will discuss
641 Sequences only as a set of Intervals.

642 *TableExample 3-35: Introducing the Sequence*

```

643 <xcal:vcalendar xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
644   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0" xs:type="xcal:VcalendarType">
645   <xcal:vcalendar>
646     <xcal:components>
647       <xcal:interval>
648         <xcal:properties>
649           <xcal:uid>
650             <xcal:text>6bf5b06f-0418-4fd7-b861-
651 d3a2b9b0292a@examples.oasis-open.org69343fc9-c1da-4cd0-abbd-889716a401d2</xcal:text>
652           </xcal:uid>
653           <xcal:duration>
654             <xcal:parameters/>
655             <xcal:duration>T1HPT1H</xcal:duration>
656           </xcal:duration>
657         </xcal:duration>
658       </xcal:properties>
659     </xcal:interval>
660     <xcal:interval>
661       <xcal:properties>
662         <xcal:uid>
663         </xcal:properties>
664       </xcal:interval>
665       <xcal:interval>
666         <xcal:properties>
667           <xcal:uid>
668             <xcal:text>a40a85bb-3052-4e52-ad20-
669 4d19eb76d9e7@examples.oasis-open.org0ba5a8c0-4eb2-49db-8514-5da18f53caaa</xcal:text>
670           </xcal:uid>
671           <xcal:duration>
672             <xcal:duration>T2H</xcal:duration>

```

```

673 |                                     </xcal:duration>
674 |                                     <xcal:related-to>
675 |                                         <xcal:parameters>
676 |                                             <xcal:reltype>
677 |                                                 <xcal:text>FS</xcal:text>
678 |                                             </xcal:reltype>
679 |                                         </xcal:parameters>
680 |                                         <xcal:uid>6bf5b06f-0418-4fa7-b861-
681 |                                         d3a2b9b0292a@examples.oasis-open.org</xcal:uid>
682 |                                     </xcal:related-to>
683 |                                         </>
684 |                                         <xcal:related-to>
685 |                                             <xcal:propertiesduration>
686 |                                                 <xcal:interval>
687 |                                                     <duration>PT2H</xcal:intervalduration>
688 |                                                 </xcal:duration>
689 |                                         </xcal:properties>
690 |                                         <xcal:interval>
691 |                                         <xcal:interval>
692 |                                             <xcal:properties>
693 |                                                 <xcal:uid>
694 |                                                 <xcal:properties>
695 |                                                 <xcal:uid>
696 |                                                 <xcal:text>9aa279d1-78d0-4ef7-a737-
697 |                                                 9519bec09007@examples.oasis-open.org</xcal:text>
698 |                                         </xcal:uid>
699 |                                         <xcal:duration>
700 |                                             <xcal:duration>T3H</xcal:duration>
701 |                                         </xcal:duration>
702 |                                         <xcal:related-to>
703 |                                             <xcal:parameters>
704 |                                                 <xcal:reltype>
705 |                                                 <xcal:text>FS</xcal:text>
706 |                                             </xcal:reltype>
707 |                                             <xcal:gap>
708 |                                                 <xcal:duration
709 |                                         xs:type="xcal:DurationPropType">
710 |                                         <xcal:duration>T10M</xcal:duration>
711 |                                         </xcal:duration>PT10M</xcal:duration>
712 |                                         </xcal:gap>
713 |                                         </xcal:parameters>
714 |                                         <xcal:uid>a40a85bb-3052-4e52-ad20-
715 |                                         4d19eb76d9e7@examples.oasis-open.org</xcal:uid>
716 |                                         </xcal:related-to>
717 |                                         </xcal:related-to>
718 |                                         <xcal:duration>
719 |                                             <xcal:duration>PT3H</xcal:duration>
720 |                                         </xcal:duration>
721 |                                         </xcal:properties>
722 |                                         </xcal:interval>
723 |                                         </xcal:components>
724 |                                     </xcal:vcalendar>

```

In [this](#) [the](#) example [above](#), the Intervals are [one hour](#), 1 hour, 2 hours, and [three](#)[3](#) hours long. There is a ten minute period between the second and third periods.

3.1.4.1 Anchoring a Sequence

A Sequence becomes an Anchored Sequence whenever [a single](#)[the Designated](#) Interval within the Sequence [is](#)[becomes](#) Anchored. An Interval is Anchored when it has a specific starting date and time (dtstart). [A Sequence may become Anchored when a Designated Interval becomes Anchored through Binding.](#) A Gluon may reference a Designated Interval through an external reference, i.e., through referring to a resolvable Uid. A given Sequence may remain Unanchored while being incorporated into many Anchored Sequences through multiple Gluon references each creating a different Bound dtStart.

[Example 3-6: An Anchored Sequence](#)

```

735 | <xcal:vcalendar xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
736 |   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0"
737 |   xs:type="xcal:VcalendarType">
738 |     <xcal:vcalendar>
739 |       <xcal:components>
740 |         <xcal:interval>
741 |           <xcal:properties>
742 |             <xcal:uid>
743 |               <xcal:text>12b59df6-eae2-41e7-a4e8-
744 |               41a9d347e54c@examples.oasis-open.org86879372-6d90-4de3-8267-
745 |               eae50c774f82</xcal:text>
746 |             </xcal:uid>
747 |             <xcal:duration>
748 |               <xcal:duration>PT15MPT15M</xcal:duration>
749 |             </xcal:duration>
750 |           </xcal:properties>
751 |           <xcal:components/>
752 |         </xcal:interval>
753 |         <xcal:interval>
754 |           <xcal:properties>
755 |             <xcal:uid>
756 |               <xcal:text>5dce9e77-8afa-4371-9437-
757 |               11d673f7f901@examples.oasis-open.orgc212aa90-6fc4-41e1-b8fb-
758 |               a23308610247</xcal:text>
759 |             </xcal:uid>
760 |             <xcal:duration>
761 |               <xcal:duration>T2H</xcal:duration>
762 |             </xcal:duration>
763 |             <xcal:related-to>
764 |               <xcal:parameters>
765 |                 <xcal:reltype>
766 |                   <xcal:text>FS</xcal:text>
767 |                 </xcal:reltype>
768 |               </xcal:parameters>
769 |               <xcal:uid>12b59df6-eae2-41e7-a4e8-
770 |               41a9d347e54c@examples.oasis-open.org86879372-6d90-4de3-8267-
771 |               eae50c774f82</xcal:uid>
772 |             </xcal:related-to>
773 |             </xcal:related_to>
774 |             <xcal:duration>
775 |               <xcal:duration>PT2H</xcal:duration>
776 |             </xcal:duration>
777 |             <xcal:dtstart>
778 |               <xcal:parameters>
779 |                 <xcal:tzid>
780 |
781 |               <xcal:text>America/New_York</xcal:text>
782 |                 </xcal:tzid>
783 |               </xcal:parameters>
784 |               <xcal:date-time>2011-03-15T09:00:00Z</xcal:date-time>
785 |               2011-05-28T09:00:00</xcal:date-time>
786 |                 </xcal:dtstart>
787 |               </xcal:properties>
788 |               <xcal:components/>
789 |             </xcal:interval>
790 |             <xcal:interval>
791 |               <xcal:properties>
792 |                 <xcal:uid>
793 |                   <xcal:text>ec72e7df-c837-4cba-afbb-
794 |                   aa54b9043158@examples.oasis-open.org50970789-1a4b-4fe5-a7e3-
795 |                   2da2c6db43ef</xcal:text>
796 |                 </xcal:uid>
797 |               <xcal:duration>

```

```

798     <xcal:duration>T30M</xcal:duration>
799     </xcal:duration>
800     <xcal:related-to>
801         <xcal:parameters>
802             <xcal:reltype>
803                 <xcal:text>FS</xcal:text>
804             </xcal:reltype>
805             <xcal:gap>
806                 <xcal:duration
807                 xs:type="xcal:DurationPropType">
808                     >PT10M</xcal:duration>
809                     <xcal:duration>T10M</xcal:duration>
810                     </xcal:duration>
811             </xcal:gap>
812             </xcal:parameters>
813             <xcal:uid>5dee9e77-8afa-4371-9437-
814             11d673f7f9010examples.oasis-open.orgc212aa90-6fc4-41e1-b8fb-
815             a23308610247</xcal:uid>
816             </xcal:related-to>
817             <!--<xcal:related-to duration="PT30M">
818                 <xcal:duration>PT30M</xcal:duration>
819             </xcal:duration>
820             </xcal:properties>
821             </xcal:interval>
822             </xcal:components>
823             </xcal:vcalendar>

```

823 Note that the entire Sequence is Anchored when a single Interval within the Sequence is Anchored.

3.1.5 State Changes

825 A common service interaction is to request that, at a certain time, a discrete state change will occur. It
826 could be that the price will rise. It could be that a report will be run. Such a communication has no logical
827 Duration. WS-Calendar communicates state changes through use of an Interval with the Duration
828 explicitly set to zero time. Because the Duration is explicit, it will not be over-ridden through
829 inheritance.inherent Duration.

830 Specifications that normatively reference and claim conformance with WS-Calendar SHALL define the
831 business meaning of zero duration Intervals.

832 While this specification extends iCalendar through the use of Intervals in Sequences, the pre-existing
833 elements of iCalendar remain in place, and more are defined periodically. State Changes can be handled
834 in one of two ways today. As iCalendar is continually extended, other ways may become available
835 tomorrow. Specifications that claim conformance to WS-Calendar SHALL state how they will
836 communicate state changes.

837 [RFC5545] specified the use of a VEVENT with a start date and time, but no end date and time and no
838 duration. WS-Calendar introduces the communication of state changes through use of an Interval with the
839 Duration explicitly set to zero time “P”. Because the Duration is explicit, it is not be over-ridden through
840 inheritance.

841

842 Example 3-7 State Change communication using Zero Duration Interval

```

843 <xcal:interval xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
844   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
845   <xcal:interval>
846     <xcal:properties>
847       <xcal:uid>
848           <xcal:text>#1bac9f0-cdd1-4f78-9a83-
849           e8f2446fe205@examples.oasis-open.org04998c75-86fb-429b-8206-
850           0a95559feb96</xcal:text>
851       </xcal:uid>
852       <xcal:duration>

```

```

853 |           <xcal:duration>T0PT0M</xcal:duration>
854 |       </xcal:duration>
855 |       <xcal:dtstart>
856 |           <xcal:parameters>
857 |               <xcal:tzid>
858 |                   <xcal:text>America/New_York</xcal:text>
859 |               </xcal:tzid>
860 |           </xcal:parameters>
861 |           <xcal:date-time>20110315T1615002011-05-28T16:15:00</xcal:date-
862 |           time>
863 |           </xcal:dtstart>
864 |       </xcal:properties>
865 |   </xcal:interval>

```

Example 3-8 State Change communication using Event without Duration or End

```

867 <xcal:vevent>
868     <xcal:properties>
869         <xcal:uid>
870             <xcal:text>abfccdb0-41b5-46c5-a71d-226cea632034</xcal:text>
871         </xcal:uid>
872         <xcal:dtstart>
873             <xcal:parameters>class
874                 <xcal:tzid>
875                     <xcal:text>America/New_York</xcal:text>
876                 </xcal:tzid>
877             </xcal:parameters>
878             <xcal:date-time>2011-05-28T16:15:00</xcal:date-time>
879         </xcal:dtstart>
880     </xcal:properties>
881     <xcal:components/>
882 </xcal:interval>
883 </xcal:vevent>

```

3.2 Attachments and Timely PerformanceTolerance

While iCalendar expresses time and intervals, WS-Calendar associates those intervals with specific services and service performance characteristics. In iCalendar componentsComponents, the ATTACH componentAttachment is used to include information outside the scope of traditional Calendar services. WS-Calendar extends the ATTACH elementAttachment to support payloads developed in other specifications. WS-Calendar also defines a new class of parametersProperty for iCalendar componentsComponents that specifies the Tolerance for variation in temporal performance requirements of the that still results in successful delivery of service.

3.2.1 Attachment and the Artifact

The WS-Calendar Each Interval contains an Attachment to provide Attach component provides a container for delivering a payload or for referencing an external service. This payload would be is transported within WS-Calendar the Interval either because it describes a service that is or can be provided over an Interval, or whose service qualities vary over several Intervals in a Sequence. As the Technical Committee cannot know all the specifications that may incorporate WS-Calendar, this specification cannot discuss the contents of this payload. WS-Calendar does expect, however, that these payloads will respect and extend the inheritance and conformance rules herein specified.

The payload may be in-line, i.e., contained within the WS-Calendar Attach, or it may be found by reference. WS-Calendar supports references either to another section of the same XML document sharing the same message as WS-Calendar element, or to an external service or specification. The WS-Calendar Attach can be thought of as having three options: "perform as described here", or "perform as described below", or "perform as described elsewhere."

The WS-Calendar Attach has three options for communicating interval-based information as below.

947 Table 3-3: Elements of a WS-Calendar Attachment

Attachment Element	Use	Discussion	Deleted
Artifact	any in-line XML (xs:any) An attachment must have at least one artifact or reference	Unevaluated (by WS-Calendar) container for payload describing service.	
uriUri	[XPOINTER] An attachment must have at least one of artifact or reference	Points to external XML, or XML located elsewhere in the document	
Text	Any text (xs:text)	The use of text in WS-Calendar an Attachment is allowed by but not defined in this specification.	

948 Specifications that incorporate WS-Calendar may wish to restrict these choices through conformance
949 requirements.

950 Example 3-9: Use of an Attachment with inline XML artifact

```

951 <xcal:interval xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
952   xmlns:xcalpayload="urn:ietf:params:xml:ns:icalendar-2.0">
953     <xcal:properties>
954       <xcal:uid>
955         <xcal:text>38db45b7-0e95-4034-af56-90901cc5b892@examples.oasis-
956         open.org</xcal:text>
957       </xcal:uid>
958       <xcal:duration>
959         <xcal:duration>T10H</xcal:duration>
960       </xcal:duration>
961       <xcal:x-wscalendar-attach>
962         <xcalnot:a:real:artifact>
963           <xx:payload xmlns:xx="urn:externally:defined:artifact">
964             <xx:units>furlongs</xx:units>
965             <xx:quantity>14</xx:quantity>
966           </xx:payload>
967         </xcal:artifact>
968       </xcal:x-wscalendar-attach>
969     </xcal:properties>
970     <xcal:components />
971   </xcal:interval>
```

972 The Artifact is of type xs:any, allowing compliant XML from any namespace to be submitted as a payload.
973 Per the conformance rules, the payload should be Fully Bound before evaluation.

974 Example 3-8: Use of an Attachment with external reference

```

975 <xcal:interval xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
976   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
977     <xcal:properties>
978       <xcal:uid>
979         <xcal:parameters/>
980         <xcal:text>a3e868ad-91e4-46ab-9281-
981         2505d03421e00@example.oasis-open.org9c829c35-061a-466e-98f5-
982         ec1fe7b49d6a</xcal:text>
983       </xcal:uid>
984       <xcal:duration>
985         <xcal:parameters/>
986         <xcal:duration>T10HPT10H</xcal:duration>
987       </xcal:duration>
```

```

988     </xcal:duration>
989     <xcal:x-wsCalendar-wsCalendar-attach>
990         <xcal:parameters/>payload:payload
991         <xcal:uri>http://examples.oasis-
992 open.org/reference/external</xcal:uri>
993             <payload:units>furlongs</payload:units>
994             <payload:quantity>11</payload:quantity>
995         </payload:payload>
996     </xcal:x-wsCalendar-wsCalendar-attach>
997     </xcal:properties>
998 </xcal:interval>
999 </xcal:components/>
```

1000 <!--The Artifact is any element derived from the Attach, allowing compliant XML from any namespace to be
1001 submitted as a payload. As per the rules of any specification claiming conformance, the payload should
1002 be Fully Bound before evaluation for completeness.

1003 Example 3-10: Use of an Attachment with external reference

```

1004 <xcal:interval>
1005     <xcal:properties>
1006         <xcal:uid>
1007             <xcal:text>ad289a5e-44b0-4e28-9cbc-d61b715f5427</xcal:text>
1008         </xcal:uid>
1009         <xcal:duration>
1010             <xcal:duration>PT10H</xcal:duration>
1011         </xcal:duration>
1012         <xcal:x-wsCalendar-attach>
1013             <xcal:uri>http://examples.oasis-
1014 open.org/ref/external</xcal:uri>
1015         </xcal:x-wsCalendar-attach>
1016     </xcal:properties>
```

1017

1018 Specifying</xcal:interval>

3.2.2 Tolerance: What is Timely Performance

1020 ~~WS-Calendar elements adapt the iCalendar objects to make interaction requirements explicit. For~~
1021 ~~example, in human schedule interactions, different organizations have their own expectations. Meetings~~
1022 ~~may start on the hour or within 5 minutes of the hour. As agents scheduled in those organizations, people~~
1023 ~~learn the expected precision. The Tolerance parameter in WS-Calendar make interaction requirements~~
1024 ~~explicit. In human schedule interactions, different organizations have their own expectations. Meetings~~
1025 ~~may start on the hour or within 5 minutes of the hour. As agents scheduled in those organizations, people~~
1026 ~~learn the expected precision. WS-CalendarFor services~~, that precision must be explicit to prevent
1027 interoperation problems.

1028 ServiceAction coordination between systems requires precise communication about expectations for the
1029 timeliness of performance. WS-Calendar defines The Tolerance parameters. Tolerance parameters
1030 are parameter added to any iCalendar component to makeComponent makes explicit the tolerance for
1031 time imprecision within a scheduled event. Tolerance can be set for applied to each Interval or for to an
1032 entire Sequence.

1033 The Tolerance componentProperty refines the meaning of time-related service communication-between
1034 services. All elements of the Tolerance parameterProperty use the Duration element as defined in
1035 [RFC5545].

1036 Table 3-4: Performance Tolerance Elements

Performance Tolerance Element	Definition	Discussion
<u>startBeforeToleranceStart</u> <u>Before Tolerance</u>	A Duration enumerating how far before the requested start time the requested service may commence.	Indicates <u>how far before the requested start time the requested service may commence, for example</u> , if a service that begins at 1:57 is compliant with a request to start at 2:00
<u>startAfterToleranceStart</u> <u>After Tolerance</u>	A Duration enumerating how far after the requested start time the requested service may commence.	Indicates <u>how far after the requested start time the requested service may commence , for example</u> , if a service that begins at 2:01 is compliant with a request to start at 2:00
<u>endBeforeToleranceEnd</u> <u>Before Tolerance</u>	A Duration enumerating how far before scheduled end time may end.	Indicates <u>how far before scheduled end time may end, for example</u> , if a service that ends at 1:57 is compliant with a request to end at 2:00
<u>endAfterToleranceEnd</u> <u>After Tolerance</u>	A Duration enumerating how far after the scheduled end time the requested service may commence.	Indicates <u>how far after the scheduled end time the requested service may commence., for example</u> , if a service that ends at 2:01 is compliant with a request to end at 2:00
<u>durationLongToleranceDuration</u> <u>Long Tolerance</u>	A Duration indicating by how much the performance Duration may exceed the Duration specified in the Interval . It may information exchange. Duration Long Tolerance SHALL NOT be used when Start and End Tolerances are both specified.	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.
<u>durationShortToleranceDuration</u> <u>Short Tolerance</u>	A Duration indicating by how much the performance Duration may fall short of exceed the Duration specified in the Interval . It may information exchange. Duration Short Tolerance SHALL NOT be used when Start and End Tolerances are both specified.	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.

Deleted C

Deleted C

Performance Tolerance Element	Definition	Discussion
granularity	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 significant When used in Tolerance , Granularity of 1 second defines the tracking and reporting requirements for a service.

Deleted C

Deleted C

1084 Tolerance is part of the core WS-Calendar service definition. Similar products or services, identical except
 1085 for different Tolerance characteristics may appear in different markets. The ability to perform within
 1086 Tolerance influences the price offered and the service selected.

1087 Note that Tolerance parameter does not indicate time, but only Duration. A Tolerance parameter
 1088 associated with an **unscheduled** Interval does not change when that Interval is scheduled.
Tolerance parameters are optional components of each WS-Calendar attachment.

1090 **Example 3-9: Performance Component**

```

1091 <xcal:interval xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
1092   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
1093   <xcal:properties>
1094     <xcal:uid>
1095     <xcal:parameters>
1096       <xcal:startbeforetolerance>
1097         <xcal:duration xs:type="xcal:DurationPropType">
1098           <xcal:duration>T1M</xcal:duration>
1099         </xcal:duration>
1100       </xcal:startbeforetolerance>
1101       <xcal:startsaftertolerance>
1102         <xcal:duration xs:type="xcal:DurationPropType">
1103           <xcal:duration>T0M</xcal:duration>
1104         </xcal:duration>
1105       </xcal:startsaftertolerance>
1106       <xcal:durationlongtolerance>
1107         <xcal:duration xs:type="xcal:DurationPropType">
1108           <xcal:duration>T0M</xcal:duration>
1109         </xcal:duration>
1110       </xcal:durationlongtolerance>
1111       <xcal:durationshorttolerance>
1112         <xcal:duration xs:type="xcal:DurationPropType">
1113           <xcal:duration>T0M</xcal:duration>
1114         </xcal:duration>
1115       </xcal:durationshorttolerance>
1116     </xcal:parameters>
1117     <xcal:text>d79e8b20-68db-43bf-8919-
1118 4e397264a654@examples.oasis-open.org</xcal:text>
1119   </xcal:uid>
1120   <xcal:duration>
1121     <xcal:duration>T30M</xcal:duration>
1122   </xcal:duration>
1123 </xcal:properties>
1124 <xcal:components/>
1125 </xcal:interval>
```

1126 In the example, the service can start as much as 1 minute earlier than the scheduled time, and
 1127 must start no later than the scheduled time. Whenever the service starts, the service must execute for
 1128 exactly the Duration indicated.

1129 Generally, the implementer should refrain from expressing unnecessary or redundant Tolerance
 1130 characteristics.

1131 **3.2.3 Expressing Service and Tolerance**

1132 Services, references and Tolerance each appear in the example below

1133 Example 3-11: Interval with inline XML artifact and optional specified Performance

```
1134 <xcal:interval xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
1135   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-
1136   2.0payload="urn:not:a:real:artifact">
1137   —<xcal:properties>
1138     ———<xcal:uid>
1139       —————<xcal:text>70e487fe-45e6-40d3-a2ae-
1140         51749e7e8e8f0examples.oasis-open.org030c603c-9e06-4dd4-8354-
1141           69dc3fa4d253</xcal:text>
1142     ———</xcal:uid>
1143     ———<xcal:duration>
1144     —<xcal:parameters>
1145       —<xcal:startbeforetolerance>
1146         ———<xcal:duration xs:type="xcal:DurationPropType">
1147           —<xcal:parameters />
1148             —>PT3H30M</xcal:duration>T10M
1149         —</xcal:duration>
1150       —</xcal:startbeforetolerance>
1151       —<xcal:startaftertolerance>
1152         —<xcal:duration xs:type="xcal:DurationPropType">
1153           —<xcal:parameters />
1154             —<xcal:duration>T0M</xcal:duration>
1155           —<xcal:duration />
1156         —</xcal:duration>
1157       —</xcal:startaftertolerance>
1158     —<xcal:parameters>
1159       —<xcal:duration>T3H30M</xcal:duration>
1160     —</xcal:duration>
1161     ———<xcal:x-wscalendarwsCalendar-attach>
1162       —<xcal:artifact>
1163         <**:payload xmlns:**="urn:externally:defined:artifact">
1164           <**>————<payload:payload>
1165             <payload:units>furlongs</**payload:units>
1166           <**>
1167             <payload:quantity>14</**11</payload:quantity>
1168           <**>————</payload:payload>
1169         —</xcal:artifact>
1170       ———<xcal:x-wscalendarwsCalendar-attach>
1171         —<xcal:tolerance>
1172           —<xcal:tolerate>
1173             —<xcal:startbefore>PT10M</xcal:startbefore>
1174               —<xcal:startafter>PT0M</xcal:startafter>
1175             —</xcal:tolerate>
1176           —</xcal:tolerance>
1177         —</xcal:properties>
1178       —<xcal:components />
1179     </xcal:interval>
```

1180 **3.3 Using Sequences: referencing, modifying, and remote access**

1181 Sequences can define specific progressions of performance or state within a wide range of services and
1182 specifications. They become more useful as they can be re-used or modified. A Sequence that is not fully
1183 specified can be adapted and re-used without re-statement. An abstract Sequence can become a service
1184 through iterative referencing.

1185 An entire Sequence can become scheduled by scheduling a single Interval in a Sequence. A single
1186 Sequence can become scheduled multiple times by repeated reference through different Gluons. The
1187 terminology describing this was introduced in Table 1-6.

1188 As a Sequence is reified through reference, WS-Calendar specifies how additional information is applied
1189 or not applied to each Interval through a chain of references. We refer to this process as inheritance.
1190 Derivative specification can take advantage of inheritance by defining specific rules that conform to the
1191 WS-Calendar inheritance pattern.

1192 This section describes how to create ~~references~~ References to Sequences, including remote
1193 ~~references~~ References, the rules that allow schedule-related information to become more complete
1194 through those references, and how to specify conforming rules in derivative specifications.

1195 3.3.1 References and Inheritance.

1196 Sequences are composed of Intervals for which a set of temporal relations have been defined.
1197 ~~[RFC5545] also defines the "PARENT", "CHILD" and "SIBLING" relationships, in which one component~~
1198 ~~references another by UID.~~ In WS-Calendar, we ~~refer to~~ refer to a Sequence by creating a ~~relationship~~
1199 ~~with Relation of type "CHILD" that references the UID of any single~~ Interval in the Sequence. ~~We refer to~~
1200 ~~As defined in~~ Table 1-6, the Interval within a Sequence that ~~has~~ is the target of this ~~relationship~~
1201 ~~as reference is~~ the Designated Interval. ~~The referring Component is named the Parent.~~

1202 Wherever ~~there is "missing" information in~~ the Designated Interval, it can ~~be inherited is inherited inherit~~
1203 ~~that information from the referring component; we use the "CHILD" relationship to reference the~~
1204 ~~designated Interval Component.~~ These references may be local or remote. Some, but not all, of the
1205 information can be inherited by the other Intervals in the Sequence.

1206 Adding additional references can further specify information in the Sequence through inheritance; these
1207 additional references ~~are~~ created by specifying an additional ~~component~~ Gluon that has a ~~parent relation~~
1208 ~~to Relation that references~~ the previous referring ~~component~~ Component as a CHILD. In this way, we can
1209 create a grand-parent and a great grand-parent.

1210 ~~A Remote Reference is a Relation to a Component external to the conveying message. A Component in~~
1211 ~~a message may reference a component already known to the receiving system. In this way, a remote~~
1212 ~~Sequence can be invoked (and scheduled) without re-definition or re-transmission.~~

1213 Each ~~parent~~ Parent bequeaths information to its ~~child~~ Child. A ~~child~~ Child inherits this information in accord
1214 with the inheritance rules. If the child is itself a parent, it bequeaths its information, the ~~bound~~ Bound result
1215 of its internal information and its inheritance, to its child. Information to complete the specification of a
1216 Sequence flows in this way from parent to child, from the outer reference to the inner Sequence.

1217 Inheritance by the ~~designated~~ Designated Interval is governed by slightly different inheritance rules than
1218 the other Intervals in the Sequence. In particular, only the ~~designated~~ Designated Interval can inherit the
1219 start date and time from its parent. The starting date and times of other Intervals in a Sequence are
1220 computed using the temporal relationships within the Sequence. Other information can be inherited by all
1221 Intervals in a Sequence. ~~Full inheritance rules are specified at [reference]. The semantics used for~~
1222 ~~inheritance is in~~ Table 1-5: Semantics: Inheritance and conformance rules for Inheritance are found in
1223 ~~Section 0.~~

1224 3.3.1.1 Introducing the Gluon

1225 The referring ~~components~~ Components described in 3.3.1 are named Gluons. In physics, gluons are
1226 particles that affect the exchanges of force between quarks, but are not themselves quarks. By analogy,
1227 WS-Calendar Gluons affect the referencing and binding of Intervals in a Sequence, but are not
1228 themselves Intervals or part of Sequences. Because Intervals can inherit almost any property from a
1229 Gluon, Gluons ~~must~~ contain most of the same information elements as Intervals. Because Intervals can
1230 contain information payloads for specifications that use WS-Calendar, ~~and these payloads can inherit~~
1231 ~~information from gluons in the same way Intervals do,~~ Gluons ~~must be able to~~ can contain information
1232 payloads from those specifications as well. ~~Gluons are described in the next section.~~

1248

3.3.1.11.1.1.1 Introducing the Gluon

1249

~~WS-Calendar~~ Gluons are used to referencing reference and bind the Intervals in a Sequence, but are not themselves Intervals or part of Sequences. ~~Gluons must contain most of the same information elements as Intervals, because Intervals~~ Gluons can inherit almost any property from a Gluon. When Intervals are used in other specifications, they contain payloads for that, or portions of payloads, which are not defined in ~~WS-Calendar~~ this specification. Information from Gluons can also hold the same payloads, and conforming specifications MUST define inheritance rules that govern inheritance within these payloads. Conformance rules, including those for inheritance conformance, are discussed in section 0 ~~Conformance and Rules for WS-Calendar and Referencing Specifications~~ Conformance and Rules for WS-Calendar and Referencing Specifications.

1250

~~WS-Calendar~~ Gluon is in essence an the Interval component profiled down to minimal elements for which inheritance rules defined, and able to carry a conforming informational payload. (See Appendix ~~Overview of WS-Calendar, its Antecedents and its Use~~) ~~Calendar~~ Overview of WS-Calendar, its Antecedents and its Use) Gluons use iCalendar relations to apply service information to Sequences.

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

Table 3-5: ~~Calendar~~ Gluon Elements

Calendar Gluon Element	Use	Discussion	Deleted
dtStamp <u>DtStamp</u>	[XCAL]:dtstamp <i>Mandatory</i>	Time and date that Calendar Gluon object <u>artifact</u> was created	
<u>Uid</u>	<i>Mandatory</i>	Used to enable unambiguous referencing of each Gluon object	
<u>Summary</u>	<i>Text</i> <i>Optional</i>	Text describing the Calendar Gluon	
<u>child</u> <u>Child</u>	As defined in [RFC5998]	A Calendar Gluon must have a link to at least one CHILD.	
dtStart <u>Duration</u>		dateTime Start time for the Designated Interval. If specified, a Duration is potentially inherited by all Intervals in the referred-to Sequence.	A Calendar Gluon may either have a dtStart or a dtEnd, but may not have both.
dtEnd <u>DtStart</u>		DateTime. Scheduled completion time for A Gluon may either have a dtStart or a dtEnd, but may not have both. DtStart is inherited by the Designated Interval. Optional	An Calendar Gluon may either have a dtStart or a dtEnd, but may not have both.

Calendar Gluon Element	Use	Discussion
durationDtEnd	Duration Optional	If specified, a Duration is potentially inherited by all Intervals in the referred-to Sequence. A Gluon may have either a dtStart or a dtEnd, but may not have both. DtEnd is inherited by the Designated Interval, in which it is used with the Bound Duration to compute the Bound dtStart.
WsCalendar-Attach	WSCalendar:Attachment Optional	The Attach used as a base class for extension by conforming specifications. Each contains the informational payload used by incorporating specifications defined in that specification. Defined in section 3.2.2.
AvailabilityAvailability	Availability, Optional See Table 3-7	Provides Referred to as Availability, provides information as to when information the service process can be scheduled.

1293 It is important to distinguish between the general model of the Gluon in WS-Calendar and the more
 1294 specific requirements of an incorporating specification. At its minimum, a Gluon may be only a pointer to a
 1295 sequence, containing only a link to its child. A Gluon may alternately include information completing (or
 1296 partially completing) the information in a Sequence; that information may vary based on what is
 1297 required to make the information payload actionable within any particular transaction.
 1298 Because the properties of the **Calendar** Gluon are bequeathed to the child Sequence, they can stand for
 1299 the elements in any Interval in the Sequence, as defined in the Conformance Section. An inherited
 1300 element can even serve as a substitute for an Interval mandatory element. For example, Duration is
 1301 mandatory for all Intervals. Intervals are able to inherit Duration from a parent. A single Duration in the
 1302 Parent can be inherited by each Interval in a Sequence.
 1303 In this way, a Sequence in which every Interval does not have a Duration, could be made complete
 1304 through inheritance. If one of those Intervals does include a Duration, the Bound Duration would be its
 1305 own, rather than that it inherited from a Parent of the Sequence.
 1306 There is a critical distinction between an individual Gluon, which may be only a pointer to a sequence, or
 1307 may have information completing (or partially completing) the information in a Sequence, and what is
 1308 required to make the information payload actionable within any particular transaction.

3.3.1.2 Availability

1310 One additional use for gluons is to expose a Sequence for remote invocation. The service offered may
 1311 be only sometimes unavailable. WS-Calendar incorporates the iCalendar extension
 1312 [Vavailability] to expose this schedule.

1313 #[Vavailability] offers a means to describe recurring temporal patterns, such as weekdays from 9:00-5:00,
 1314 Thursday mornings until July, and thereafter Tuesday evening as well. A Vavailability component is likely
 1315 that a collection of Availability components, each with its data boundaries and its recurrence patterns. The
 1316 parameters and properties are those defined in iCalendar, the structure is defined in the referenced
 1317 [Vavailability], and the artifact is an optional Component of a Gluon.

1318 A requestor may not be aware of all aspects of the Sequence. A service requester is aware only of when
 1319 he wants the service and for how long. These are properties requestor does know, however, the desired
 1320 Start and Duration of the Designated Interval. [Vavailability] in a Gluon is Availability will be interpreted
 1321 as a filter only on the Designated Interval, but on no others.

1322 Table 3-7: Availability elements with specified use in WS-Calendar

Vavailability Element	Use	Discussion
-----------------------	-----	------------

Availability Element	Use	Discussion
dtStamp	[XCAL]:dtstamp <i>Mandatory</i>	Time and date that Vailability object was created
Uid	<i>Mandatory</i>	Used to enable unambiguous referencing of each Vailability object
Summary	Text <i>Optional</i>	Text describing the Vailability
dtStart	xcal:dateTime Start time for the Availability. <i>Mandatory</i>	All time before the dtStart is considered unavailable..
dtEnd	xcal:dateTime. End point for availability. <i>Optional</i>	If present, all recurrence and other patterns inside the Available objects ends with the dtEnd. Either a dtStart or a dtEnd, may be present, but not both.
duration	Duration <i>Optional</i>	If present, the duration beginning with the dtStart is time for which availability is specified. Either a dtStart or a dtEnd, may be present, but not both,
Availability	Availability, one to many occurrences. See Table 3-8	At least one Availability is required to state the pattern when the service is available
Granularity	xcal:duration <i>Optional</i>	Granularity, when used in Availability, limits when a service can be scheduled. For example, a 15 minute granularity on a 9:00 dtStart, implies that legal dtStarts are 9:00, 9:15, 9:30, 9:45, ...

1323 If WS-Calendar adds a single optional parameter to the [Availability] component. When a Granularity
 1324 component is applied, it further defines the acceptable service invocation. Granularity is discussed in the
 1325 next section.

1326 3.3.1.3 Granularity used as part of Availability

1327 Granularity can be applied both to Vailability is not terminated by either an dtEnd or bounded by a
 1328 duration, then the end of the Vailability is undefined. If Vailability is so terminated, then that
 1329 termination bounds any recurrence patterns defined in the Available elements.
 1330 Available elements define the actual times during which the resource or service is available for invocation
 1331 or scheduling.

1332 *Table 3-8:Availability elements contained within Vailability.*

Availability Element	Use	Discussion
Uid	<i>Mandatory</i>	Used to enable unambiguous referencing of each Availability object
Summary	Text <i>Optional</i>	Text describing the Availability

Availability Element	Use	Discussion
dtStart	<code>xcal:dateTime</code> Start time for the Availability. <i>Mandatory</i>	All time before the dtStart is considered unavailable..
dtEnd	<code>xcal:dateTime</code> End point for availability. <i>Optional</i>	If present, all recurrence and other patterns inside the Available objects ends with the dtEnd. Either a dtStart or a dtEnd, may be present, but not both.
duration	Duration <i>Optional</i>	If present, the duration beginning with the dtStart is time for which availability is specified. Either a dtStart or a dtEnd, may be present, but not both,
RRule	<code>xcal:rrule</code>	Defines how often the availability interval, defined as either the period bounded by the dtStart and dtEnd OR beginning with the dtStart and lasting for the duration, repeats
Granularity	<code>xcal:duration</code> <i>Optional</i>	Granularity, when used in Availability, limits when a service can be scheduled. For example, a 15 minute granularity on a 9:00 dtStart, implies that legal dtStarts are 9:00, 9:15, 9:30, 9:45, ...

1333 3.3.1.3 Granularity

1334 Granularity affects both Availability and Availability (the collection) and to Availability (the individual rule).
 1335 If Granularity is specified, then it communicates the expectation that services that invoke WS-Calendar
 1336 specifications conforming services should request them in multiples of only Start times that match the
 1337 Granularity.

1338 For example, assume the Designated Interval of a service Sequence has a Duration of One Hour, and is
 1339 available on weekdays from 9:00 to 8:30 until 11:00. Without Granularity suggests that if, the
 1340 durationService can be specified, then it should be specified as one of 15M, 30M, 45M, 1H, 1H15M,
 1341 1H30M, 1H45M or 2H. The sameScheduled at any time that does not start before 8:30, nor end after
 1342 11:00. If a Granularity indicates that the service can only be specified in of 30 minutes "PT30M" is
 1343 applied, the Scheduled Starts are limited to 8:30, 9:00, 8:30, and 10:00, i.e., integral multiples of the
 1344 Duration of the Granularity following the dtStart, i.e., at 9:00 (0x), 9:15 (1x), 9:30 (2x), 9:45(3x), and so
 1345 onbeginning at the beginning of the available window.

1346 3.3.2 Calendar Gluons and Sequences

1347 WS-Calendar Gluons express common service requirements for an entire Sequence. If a Gluon is parent
 1348 to an Interval in a Sequence, then the Gluon's Attachment expresses service attributes inheritable by all
 1349 Intervals in the Sequence.

1350 In this example, the Sequence in the previous example is expressed using a Calendar Gluon.

1351 Example 3-12: Sequence with Performance defined in the Calendar Gluon

```

1352 <xcal:vcalendar xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
1353   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0"
1354   xs:type="xcal:VcalendarType">
1355   —<xcal:properties/>
1356   <xcal:vcalendar>
```

```

1357 <xcal:components>
1358     <xcal:gluon>
1359         <xcal:properties>
1360             <xcal:uid>
1361                 <xcal:text>2f9d675e-88b3-457d-a6e1-
1362 3045aec1816d6@examples.oasis-open.org5ffaa487-206f-46e8-b3e5-
1363 958b37477cab</xcal:text>
1364         </xcal:uid>
1365         <xcal:related-to>
1366             <xcal:parameters>
1367                 <xcal:reltype>
1368                     <xcal:text>CHILD</xcal:text>
1369                 </xcal:reltype>
1370             </xcal:parameters>
1371         </xcal:uid>
1372         <xcal:related-to>
1373             <xcal:parameters>
1374                 <xcal:reltype>
1375                     <xcal:text>FS</xcal:text>
1376                 </xcal:reltype>
1377             </xcal:parameters>
1378             <xcal:uid>c53d40cc-5e9e-44a4-9674-
1379 6ad492e76021@examples.oasis-open.org2a7de3f0-54c5-4a31-9856-
1380 6a94e6c82902</xcal:uid>
1381         </xcal:related-to>
1382         <xcal:dtstart>
1383             <xcal:parameters>
1384                 <xcal:tzid>
1385
1386                 <xcal:text>America/New_York</xcal:text>
1387                 </xcal:tzid>
1388             </xcal:related-to>
1389             <xcal:dtstart>
1390                 <xcal:parameters>
1391                     <xcal:tzid>
1392
1393             <xcal:text>America/New_York</xcal:text>
1394             </xcal:tzid>
1395             <xcal:startbeforetolerance>
1396                 <xcal:duration>
1397             *s:type="xcal:DurationPropType">
1398
1399                 <xcal:duration>T0M</xcal:duration>
1400                     </xcal:duration>
1401                     <xcal:startbeforetolerance>
1402                     <xcal:startaftertolerance>
1403                     <xcal:duration>
1404             *s:type="xcal:DurationPropType">
1405
1406                 <xcal:duration>T0M</xcal:duration>
1407                     </xcal:duration>
1408                     <xcal:stardaftertolerance>
1409                     <xcal:durationlongtolerance>
1410                     <xcal:duration>
1411             *s:type="xcal:DurationPropType">
1412
1413                 <xcal:duration>T5M</xcal:duration>
1414                     </xcal:duration>
1415                     <xcal:durationlongtolerance>
1416                     <xcal:durationshorttolerance>
1417                     <xcal:duration>
1418             *s:type="xcal:DurationPropType">

```

```

1419
1420 <xcal:duration>T0M</xcal:duration>
1421                                     </xcal:duration>
1422                                     </xcal:durationshorttolerance>
1423                                     <xcal:granularity>
1424                                     <xcal:duration
1425 xs:type="xcal:DurationPropType">
1426
1427 <xcal:duration>T5S</xcal:duration>
1428                                     </xcal:duration>
1429                                     </xcal:granularity>
1430                                     </xcal:parameters>
1431                                     <xcal:date-time>20110315T084500002011-05-
1432 28T08:45:00</xcal:date-time>
1433                                     </xcal:dtstart>
1434                                     <xcal:tolerance>
1435                                     <xcal:tolerate>
1436
1437 <xcal:durationlong>PT5M</xcal:durationlong>
1438
1439 <xcal:durationshort>PT0M</xcal:durationshort>
1440                                     <xcal:granularity>PT5S</xcal:granularity>
1441                                     </xcal:tolerate>
1442                                     </xcal:tolerance>
1443                                     <xcal:x-wscalendarsCalendar-attach>
1444                                     <xcal:artifact><payload:payload>
1445                                     <payload:units>furlongs</payload:units>
1446                                     <payload:quantity>11</payload:quantity>
1447                                     </payload:payload>
1448                                     </xcal:x-wscalendarsCalendar-attach>
1449                                     </xcal:properties>
1450                                     </xcal:gluon>
1451                                     <xcal:interval>
1452                                     <xcal:properties>
1453                                     <xcal:uid>
1454                                     <xcal:text>c53d40cc-5e9e-44a4-9674-
1455 6ad492e76021@examples.oasis-open.org</xcal:text>
1456                                     </xcal:uid>
1457                                     </xcal:properties>
1458                                     </xcal:interval>
1459                                     <xcal:interval>
1460                                     <xcal:properties>
1461                                     <xcal:uid>
1462                                     <xcal:parameters/>
1463                                     <xcal:text>67319fa7-28b3-4abe-91b8-
1464 e595fc2948a8@examples.oasis-open.org</xcal:text>
1465                                     </xcal:uid>
1466                                     <xcal:related_to>
1467                                     <xcal:parameters>
1468                                     <xcal:reltype>
1469                                     <xcal:text>FS</xcal:text>
1470                                     </xcal:reltype>
1471                                     </xcal:parameters>
1472                                     <xcal:uid>c53d40cc-5e9e-44a4-9674-
1473 6ad492e76021@examples.oasis-open.org</xcal:uid>
1474                                     </xcal:related_to>
1475                                     </xcal:properties>
1476                                     <xcal:components/>
1477                                     </xcal:gluon>
1478                                     <xcal:interval>
1479                                     <xcal:properties>
1480                                     <xcal:uid>
1481                                     </xcal:interval>
```

```

1482 <xcal:interval>
1483   <xcal:properties>
1484     <xcal:uid>
1485       <xcal:text>2a7de3f0-54c5-4a31-9856-
1486       6a94e6c82902</xcal:text>
1487     </xcal:uid>
1488   </xcal:properties>
1489 </xcal:interval>
1490 <xcal:interval>
1491   <xcal:properties>
1492     <xcal:uid>
1493       <xcal:text>1886ebd8-a5a9-4aa8-9b6b-
1494       2869e4b711de</xcal:text>
1495     </xcal:uid>
1496   <xcal:related-to>
1497     <xcal:parameters>
1498       <xcal:reltype>
1499         <xcal:text>FS</xcal:text>
1500       </xcal:reltype>
1501     </xcal:parameters>
1502   <xcal:uid>2a7de3f0-54c5-4a31-9856-
1503   6a94e6c82902</xcal:uid>
1504   </xcal:related-to>
1505 </xcal:properties>
1506 </xcal:interval>
1507 <xcal:interval>
1508   <xcal:properties>
1509     <xcal:uid>
1510       <xcal:text>e6e3e351-77ee-4e27-abee-
1511       8e5c1d9ef6db@examples.oasis-open.org62e518b4-65df-4fea-b64a-
1512       91a762ae173a</xcal:text>
1513     </xcal:uid>
1514   <xcal:related-to>
1515     <xcal:parameters>
1516       <xcal:reltype>
1517         <xcal:text>FS</xcal:text>
1518       </xcal:reltype>
1519     </xcal:parameters>
1520   </xcal:uid>
1521   <xcal:duration>
1522     <xcal:duration>T30M</xcal:duration>
1523   </xcal:duration>
1524   <xcal:related-to>
1525     <xcal:parameters>
1526       <xcal:reltype>
1527         <xcal:text>FS</xcal:text>
1528       </xcal:reltype>
1529     </xcal:parameters>
1530   <xcal:uid>67319fa7-28b3-4abe-91b8-
1531   e595fe2948a8@examples.oasis-open.org1886ebd8-a5a9-4aa8-9b6b-
1532   2869e4b711de</xcal:uid>
1533   </xcal:related-to>
1534   </<xcal:related-to>>
1535     <xcal:duration>PT30M</xcal:duration>
1536   </xcal:duration>
1537   </xcal:properties>
1538   <xcal:components/>
1539 </xcal:interval>
1540 </xcal:components>
1541 </xcal:vcalendar>

```

1542 Note that the performance expectations, identical for each Interval, have moved into the **Calendar** Gluon.
1543 Not also that while the duration for all Intervals in the partition is set in the **Calendar** Gluon, Interval 3

1544 overrides that with a half-hour duration assigned locally. This [Calendar](#) Gluon happens to be related to
1545 the first Interval in the Sequence; there are specific use cases (discussed below) which require it to be
1546 linked to other Intervals.

1547 3.3.3 Inheritance rules for [Calendar](#) Gluons

1548 In general, the rule is that anything specified in the Parent [Calendar](#) Gluon applies to each Child. The
1549 Parent of an Interval in a Sequence is parent to all Intervals in the Sequence. As a Sequence creates
1550 single temporal relationship, assigning a start time (dtstart) to any Interval allows computation of the
1551 starting time for each of them.

1552 *Table 3-6 Gluon Inheritance rules*

Attribute	Inheritance Rules
General	A Interval or Calendar Gluon inherits its attributes through it's the parent. Local specification of an attributes overrides any inheritance.
Duration	Follows general rules
Temporal Relation	Relationship Type and Gap only are inherited. Either may be overridden locally. To specify no gap when a parent specifies a gap, an explicit zero duration gap must be specified. Related-to is not inherited.
Performance	Performance is either inherited intact or overridden completely. There are no rules for recombining partial Performance objects through inheritance.
Artifacts	Artifacts hold payload from other specifications. Elements within Artifacts are inherited in accord with the rules in those specifications, which must be consistent the inheritance rules in WS-Calendar. Artifacts are evaluated for completeness and conformance only after processing inheritance.
Schedules Schedule	Schedules Schedule , i.e., the start date and time, are inherited only by the designated Designated Interval. The start date and times of other Intervals are computed by reference to the designated Designated Interval. Between the Gluon bequeathing a schedule and the Designated Interval, an intervening Gluon may set availability Availability . It is up to the application or to the specification incorporating WS-Calendar to assert whether an Interval that is outside the availability Availability is conforming or not.
Availability	Availability communicates restrictions on when a service is offered. Service availability is interpreted for the Designated Interval only. If there are two availability Availability objects, they are evaluated for the union of the two availabilities. For example, if I am available all week from 2:00 to 6:00 in one, and available all day Tuesday in the other, then after inheritance, there remains only 2:00 to 6:00 on Tuesday,

1553 3.3.4 Optimizing the expression of a Partition

1554 [Partitions are Sequences composed](#) A Partition is a set of consecutive Intervals. [A](#)The expressions of a
1555 Partition can be [further](#) optimized by bringing the [relationship](#)
[Relation and Duration](#) into the Gluon. Notice
1556 that while the type of the relationship is defined in the [Calendar](#) Gluon, the Temporal Relation for each
1557 Interval must still be expressed within the Interval.

1558 *Example 3-13: Partition with Duration and Relationship defined in the [Calendar](#) Gluon*

```

1559 <xcal:vcalendar xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
1560   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0"
1561   xs:type="xcal:VcalendarType">
1562     <xcal:properties/>
1563   <xcal:vcalendar>
1564     <xcal:components>
1565       <xcal:gluon>
1566         <xcal:properties>
1567           <xcal:uid>
1568             <xcal:parameters/>
1569             <xcal:text>97e504ed-263e-447d-95a6-
1570 d59b97422ede@examples.oasis-open.org</xcal:text>
1571 6e8448d8e125</xcal:uid>
1572 </xcal:uid>
1573 <xcal:related_to>
1574 <xcal:parameters>
1575 <xcal:reltype>
1576   <xcal:text>CHILD</xcal:text>
1577 </xcal:reltype>
1578 </xcal:parameters>
1579 <xcal:uid>9b1c1ae8-ca4f-4065-9cf6-
1580 45c53e709e55@examples.oasis-open.org</xcal:uid>
1581 </xcal:related_to>
1582 <xcal:related_to>
1583 <xcal:parameters>
1584 <xcal:reltype>
1585   <xcal:text>FS</xcal:text>
1586 </xcal:reltype>
1587 </xcal:parameters>
1588 <xcal:uid>9b1c1ae8-ca4f-4065-9cf6-
1589 45c53e709e55@examples.oasis-open.org</xcal:uid>
1590 </xcal:related_to>
1591 <xcal:duration>
1592   <xcal:duration>T15M</xcal:duration>
1593 </xcal:duration>
1594 <xcal:xs:wcalendar-attach>
1595 <xcal:parameters/>
1596 <xcal:artifact/>
1597 </xcal:xs:wcalendar-attach>
1598 </xcal:properties>
1599 <xcal:components/>
1600 </xcal:gluon>
1601 <xcal:interval>
1602 <xcal:properties>
1603 <xcal:uid>
1604   <xcal:text>9b1c1ae8-ca4f-4065-9cf6-
1605 45c53e709e55@examples.oasis-open.org</xcal:text>
1606 </xcal:uid>
1607 <xcal:related_to>
1608   <xcal:uid>9b1c1ae8-ca4f-4065-9cf6-
1609 45c53e709e55@examples.oasis-open.org</xcal:uid>
1610 </xcal:related_to>
1611 <xcal:properties>
1612 <xcal:components/>
1613 </xcal:interval>
1614 <xcal:interval>
1615 <xcal:properties>
1616 <xcal:uid>
1617   <xcal:text>50149441-18e8-4c8b-9cf6-
1618 e8da3e671895@examples.oasis-open.org</xcal:text>
1619 </xcal:uid>
1620 <xcal:related_to>

```

```

1621                                     <xcal:uid>50149441-18e8-4c8b-9e0f-
1622                                     e8da3e671895@examples.oasis-open.org</xcal:uid>
1623                                     </xcal:related_to>
1624                                     </xcal:properties>
1625                                     </xcal:interval>
1626                                     <xcal:properties>
1627                                     <xcal:uid>
1628                                     <xcal:parameters/>
1629                                     <xcal:text>661e6127-9e06-429c-b641-
1630                                     205a31df64d1@examples.oasis-open.org</xcal:text>
1631                                     </xcal:uid>
1632                                     <xcal:related_to>
1633                                     <xcal:uid>661e6127-9e06-429c-b641-
1634                                     205a31df64d1@examples.oasis-open.org</xcal:uid>
1635                                     </xcal:related_to>
1636                                     </xcal:properties>
1637                                     </xcal:interval>
1638                                     <xcal:interval>
1639                                     <xcal:properties>
1640                                     <xcal:uid>
1641                                     <xcal:parameters/>
1642                                     <xcal:text>c4457e5d-a848-4878-8571-
1643                                     2f35ed02e594@examples.oasis-open.org</xcal:text>
1644                                     </xcal:uid>
1645                                     <xcal:related_to>
1646                                     <xcal:uid>c4457e5d-a848-4878-8571-
1647                                     2f35ed02e594@examples.oasis-open.org</xcal:uid>
1648                                     </xcal:related_to>
1649                                     </xcal:properties>
1650                                     </xcal:interval>
1651                                     <xcal:interval>
1652                                     <xcal:properties>
1653                                     <xcal:uid>
1654                                     <xcal:text>13ffa401-dd7a-48cc-980a-
1655                                     8f19aa91fd58@examples.oasis-open.org</xcal:text>
1656                                     </xcal:uid>
1657                                     <xcal:related_to>
1658                                     <xcal:uid>13ffa401-dd7a-48cc-980a-
1659                                     8f19aa91fd58@examples.oasis-open.org</xcal:uid>
1660                                     </xcal:related_to>
1661                                     </xcal:properties>
1662                                     </xcal:interval>
1663                                     <xcal:interval>
1664                                     <xcal:properties>
1665                                     <xcal:uid>
1666                                     <xcal:text>c5ef94b3-4514-4093-b8a7-
1667                                     70cd7bd174b1@examples.oasis-open.org</xcal:text>
1668                                     </xcal:uid>
1669                                     <xcal:related_to>
1670                                     <xcal:parameters>
1671                                     <xcal:uid>c5ef94b3-4514-4093-b8a7-
1672                                     70cd7bd174b1@examples.oasis-open.org</xcal:uid>
1673                                     </xcal:related_to>
1674                                     </xcal:properties>
1675                                     </xcal:interval>
1676                                     <xcal:interval>
1677                                     <xcal:properties>
1678                                     <xcal:uid>
1679                                     <xcal:text>38f4e770-5f3d-466f-9222-
1680                                     f1c801a43657@examples.oasis-open.org</xcal:text>
1681                                     </xcal:uid>
1682                                     <xcal:related_to>
1683

```

```

1684      <xcal:uid>38f4e770-5f3d-466f-9222-
1685      f1e801a43657@examples.oasis-open.org</xcal:uid>
1686      </xcal:related-to>
1687      </xcal:properties>
1688      </xcal:interval>
1689      <xcal:interval>
1690          <xcal:properties>
1691              <xcal:uid>
1692                  <xcal:text>daa6e916-de0a-4c1a-a852-
1693                  1670fd97d47e@examples.oasis-open.org</xcal:text>
1694                  </xcal:uid>
1695                  <xcal:related-to>
1696                      <xcal:uid>daa6e916-de0a-4c1a-a852-
1697                      1670fd97d47e@examples.oasis-open.org</xcal:uid>
1698                      </xcal:related-to>
1699                  </xcal:properties>
1700          </xcal:interval>
1701      </xcal:components>
1702  </xcal:calendar>
```

1703 This Partition shows 8 consecutive 15 minute intervals as part of a 2 hour partition.

1704 *Example 3-13: Partition with Duration, Relationship, and Gap defined in the Calendar Gluon*

```

1705 <xcal:vecalendar xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
1706   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0"
1707   xs:type="xcal:VcalendarType">
1708     <xcal:components>
1709       <xcal:gluon>
1710         <xcal:properties>
1711             <xcal:uid>
1712                 <xcal:text>c7496e78-6d71-4118-b42a-
1713                 641fefe02a9@examples.oasis-open.org</xcal:text>
1714             </xcal:uid>
1715             <xcal:related-to>
1716                 <xcal:parameters>
1717                     <xcal:reltype>
1718                         <xcal:text>CHILD</xcal:text>
1719                     </xcal:reltype>
1720                 </xcal:parameters>
1721                 <xcal:uid>5b7b5f46-fbc4-455e-9e60-
1722                 7639463aea4e@examples.oasis-open.org</xcal:uid>
1723                 a9207004a4cc</xcal:uid>
1724             <xcal:gap>
1725                 <xcal:duration>PT10M</xcal:duration>
1726             </xcal:gap>
1727             <xcal:related-to>
1728                 <xcal:duration>
1729                     <xcal:duration>PT50M</xcal:duration>
1730                 </xcal:duration>
1731                 <xcal:x-wsCalendar-attach>
1732                     <payload:payload>
1733                         <payload:units>students</payload:units>
1734                         <payload:quantity>23</payload:quantity>
1735                     </payload:payload>
1736                 </xcal:x-wsCalendar-attach>
1737                 <xcal:tolerance>
1738                     <xcal:tolerate>
1739                         <xcal:startbefore>P</xcal:startbefore>
1740                         <xcal:startafter>PT5M</xcal:startafter>
1741                     </xcal:tolerate>
1742                 </xcal:tolerance>
1743             </xcal:properties>
```

```

1744      <xcal:components/>
1745    </xcal:gluon>
1746    <xcal:interval>
1747      <xcal:properties>
1748        <xcal:uid>
1749          <xcal:text>bfe0040e-a5e0-4558-bbe1-
1750          a9207004a4cc</xcal:text>
1751        </xcal:uid>
1752      </xcal:properties>
1753    </xcal:interval>
1754    <xcal:interval>
1755      <xcal:properties>
1756        <xcal:uid>
1757          <xcal:text>7dabb353-8d0a-44c5-a81e-
1758          e49268fd85f3</xcal:text>
1759        </xcal:uid>
1760      <xcal:related-to>
1761        <xcal:parameters>
1762          <xcal:reltype>
1763            <xcal:text/>
1764          </xcal:reltype>
1765        </xcal:parameters>
1766      <xcal:uid>bfe0040e-a5e0-4558-bbe1-
1767      a9207004a4cc</xcal:uid>
1768      </xcal:related-to>
1769    </xcal:properties>
1770  </xcal:interval>
1771  <xcal:interval>
1772    <xcal:properties>
1773      <xcal:uid>
1774        <xcal:text>4c9ca94b-0b47-4244-b72d-
1775        e5333e1f7e43</xcal:text>
1776      </xcal:uid>
1777      <xcal:related-to>
1778        <xcal:parameters>
1779          <xcal:reltype>
1780            <xcal:text/>
1781          </xcal:reltype>
1782        </xcal:parameters>
1783      <xcal:uid>bfe0040e-a5e0-4558-bbe1-
1784      a9207004a4cc</xcal:uid>
1785      </xcal:related-to>
1786    </xcal:properties>
1787  </xcal:interval>
1788  <xcal:interval>
1789    <xcal:properties>
1790      <xcal:uid>
1791        <xcal:text>397068e3-ca9a-4b3a-a1e0-
1792        76a45671bb2f</xcal:text>
1793      </xcal:uid>
1794      <xcal:related-to>
1795        <xcal:parameters>
1796          <xcal:reltype>
1797            <xcal:text>FS</xcal:text></>
1798          </xcal:reltype>
1799          <xcal:gap>
1800          <xcal:duration>
1801          <x:type="xcal:DurationPropType">
1802            </>
1803          </xcal:parameters></>
1804          <xcal:duration>T10M</xcal:duration>
1805          <xcal:uid>bfe0040e-
1806          a5e0-4558-bbe1-a9207004a4cc</xcal:durationuid>
1807          </xcal:related-to>
1808        </xcal:properties>

```

```

1807      </xcal:interval>
1808      <xcal:interval>
1809          <xcal:properties>
1810              <xcal:uid>
1811                  _____ </<xcal:duration>text>c97dfe10-1e2d-
1812 42c2-bdc5-41a5ca17c98a</xcal:text>
1813          </xcal:uid>
1814          <xcal:related-to>
1815              </xcal:gap>
1816          <!--<xcal:parameters>
1817          <xcal:related-to>reltype>
1818          <xcal:duration>text/>
1819          </<xcal:reltype>
1820          </xcal:parameters/>>
1821          <xcal:duration>T50M<xcal:uid>bfe0040e-a5e0-4558-bbe1-
1822 a9207004a4cc</xcal:duration>
1823          </xcal:uid>
1824          <xcal:related-to>
1825          </xcal:properties>
1826          <xcal:interval>
1827              <xcal:properties>
1828                  <xcal:uid>
1829                      <xcal:text>bf22e5e1-891f-40eb-9012-
1830 89577482f38d</xcal:text>
1831                  </xcal:uid>
1832                  <xcal:related-to>
1833                      </xcal:duration>
1834                      <xcal:x-wscalendar-attach>
1835                          <xcal:parameters/>>
1836                          <xcal:artifact/>reltype>
1837                      <!--<xcal:x-wscalendar-attach>text/>
1838                      </xcal:properties>reltype>
1839                  </xcal:gluon>
1840                  <xcal:interval>
1841                      <xcal:properties>
1842                          <xcal:uid>
1843                              <!--<xcal:parameters/>>
1844                              <xcal:uid>bfe0040e-a5e0-4558-bbe1-
1845 a9207004a4cc</xcal:uid>
1846                      </xcal:related-to>
1847                      </xcal:properties>
1848                      <xcal:interval>
1849                          <xcal:interval>
1850                              <xcal:properties>
1851                                  <xcal:uid>
1852                                      <xcal:text>5b7b5f46-fbc4-455e-9c60-
1853 7639463aca4e@examples.oasis-open.org</xcal:text>
1854 2dae1d86f10b</xcal:text>
1855          </xcal:uid>
1856          <xcal:related-to>
1857              <xcal:uid>
1858                  <xcal:related-to>
1859                      <xcal:uid>5b7b5f46-fbc4-455e-9c60-
1860 7639463aca4e@examples.oasis-open.org</xcal:uid>
1861                  </xcal:related-to>
1862          </xcal:properties>
1863          <xcal:interval>
1864              <xcal:interval>
1865                  <xcal:properties>
1866                      <xcal:uid>
1867                          <xcal:parameters/>>
1868                          <xcal:text>43da0574-d00b-41e8-8a47-
1869 70767f63da78@examples.oasis-open.org</xcal:text>

```

```

1870                                     </xcal:uid>
1871                                     <xcal:related-to>
1872                                         <xcal:uid>43da0574-d00b-41e8-8a47-
1873 70767f63da78@examples.oasis-open.org</xcal:uid>
1874                                     </xcal:related-to>
1875                                     </xcal:properties>
1876                                     </xcal:interval>
1877                                     <xcal:interval>
1878                                         <xcal:propertiesreltype>
1879                                         <xcal:uid>text/>
1880                                         <!-->
1881                                         </xcal:reltype>
1882                                         </xcal:parameters/>
1883                                         <xcal:text>d586e62f-617b-4207-a937-
1884 9a0ec8d45b5c@examples.oasis-open.org</xcal:text>
1885                                         </xcal:uid>
1886                                         <xcal:related-to>
1887                                         <xcal:uid>d586e62f-617b-4207-a937-
1888 9a0ec8d45b5c@examples.oasis-open.org</xcal:uid>
1889                                         </xcal:related-to>
1890                                         </xcal:properties>
1891                                         </xcal:interval>
1892                                         <xcal:interval>
1893                                         <xcal:properties>
1894                                         <xcal:uid>bfe0040e-a5e0-4558-bbe1-
1895 a9207004a4cc</xcal:uid>
1896                                         </xcal:related-to>
1897                                         </xcal:properties>
1898                                         </xcal:interval>
1899                                         <xcal:interval>
1900                                         <xcal:properties>
1901                                         <xcal:uid>
1902                                         <xcal:parameters/>
1903                                         <xcal:text>848af3d9-5b1b-4b4f-a353-
1904 39a4b2e857f9@examples.oasis-open.org5b1e382f-4ff0-43fe-9535-
1905 1d98ca2a852d</xcal:text>
1906                                         </xcal:uid>
1907                                         <xcal:related-to>
1908                                         <xcal:uid>848af3d9-5b1b-4b4f-a353-
1909 39a4b2e857f9@examples.oasis-open.org</xcal:parameters>
1910                                         <xcal:uidreltype>
1911                                         <!-->
1912                                         </xcal:related-to><xcal:related-to>text/>
1913                                         </xcal:propertiesreltype>
1914                                         </xcal:interval>
1915                                         <xcal:interval>
1916                                         <xcal:properties>
1917                                         <xcal:uid>
1918                                         <!-->
1919                                         </xcal:parameters/>
1920 8dbfe6448ce8@examples.oasis-open.org</xcal:text>
1921                                         <xcal:uid>bfe0040e-
1922 a5e0-4558-bbe1-a9207004a4cc</xcal:uid>
1923                                         </xcal:related-to>
1924                                         </xcal:properties>
1925                                         </xcal:interval>
1926                                         </xcal:components>
1927                                         </xcal:vcalendar>
1928                                         <xcal:related-to>
1929                                         <xcal:uid>e088de06-770c-44b8-9abf-
1930 8dbfe6448ce8@examples.oasis-open.org</xcal:uid>
1931                                         </xcal:related-to>
1932                                         </xcal:properties>

```

```

1933   <xcal:properties>
1934     <xcal:uid>
1935     <xcal:parameters/>
1936     <xcal:text>c0be8725-383d-4019-96a0-
1937 3a3a7e19fe83@examples.oasis-open.org</xcal:text>
1938   </xcal:uid>
1939   <xcal:related-to>
1940     <xcal:uid>c0be8725-383d-4019-96a0-
1941 3a3a7e19fe83@examples.oasis-open.org</xcal:uid>
1942   </xcal:related-to>
1943 </xcal:properties>
1944 </xcal:interval>
1945 <xcal:interval>
1946   <xcal:properties>
1947     <xcal:uid>
1948     <xcal:parameters/>
1949     <xcal:text>2a4b8df8-47da-474b-9a2e-
1950 771c3f8ef915@examples.oasis-open.org</xcal:text>
1951   </xcal:uid>
1952   <xcal:related-to>
1953     <xcal:uid>2a4b8df8-47da-474b-9a2e-
1954 771c3f8ef915@examples.oasis-open.org</xcal:uid>
1955   </xcal:related-to>
1956 </xcal:properties>
1957 </xcal:interval>
1958 <xcal:interval>
1959   <xcal:properties>
1960     <xcal:uid>
1961     <xcal:parameters/>
1962     <xcal:text>fad65470-0e03-4dea-8669-
1963 90d1b49769c2@examples.oasis-open.org</xcal:text>
1964   </xcal:uid>
1965   <xcal:related-to>
1966     <xcal:uid>fad65470-0e03-4dea-8669-
1967 90d1b49769c2@examples.oasis-open.org</xcal:uid>
1968   </xcal:related-to>
1969 </xcal:properties>
1970 </xcal:interval>
1971 </xcal:components>
1972 </xcal:calendar>

```

ThisThe Partition [above](#) shows a school schedule in which classes start one hour apart. Each class is for 50 minutes, and there is a 10 minute gap between each as students move between classes. Classes may not begin before the schedule, but they may start up to five minutes late.

3.3.5 Controlling Start Times in Service Advertisements

3.3.5 Notifying Partners of Process Availability

A Sequence has not been scheduled until it has a start [timedate](#) and [date-time](#). Sometimes it is useful to [control](#)[limit](#) the possible start-times. For example, consider a service that is only available at 9:00 AM each day. It has not yet been scheduled, so [#sits](#) dtStart is empty. The Availability object, expressed either in the [designated interval](#)[Designated Interval](#), or in the lineage of Gluons, is used to restrict this offering.

Example 3-14: Availability

```

1984 <xcal:vavailability xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
1985   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
1986   <xcal:properties>
1987     <xcal:uid>

```

```

1988          <xcal:text>eed68bda-ec20-4814-b48b-
1989 2ff75ee568210examples.oasis-open.org9ae65e93-4c68-4811-aa10-
1990 fcdbacb7ba79</xcal:text>
1991      </xcal:uid>
1992      <xcal:dtstart>
1993          <xcal:parameters>
1994              <xcal:tzid>
1995                  <xcal:text>America/New_York</xcal:text>
1996              </xcal:tzid>
1997          </xcal:parameters>
1998          <xcal:date-time>20110301T000000002011-03-
1999 01T00:00:00</xcal:date-time>
2000      </xcal:dtstart>
2001      <xcal:dtend>
2002          <xcal:parameters>
2003              <xcal:tzid>
2004                  <xcal:text>America/New_York</xcal:text>
2005              </xcal:tzid>
2006          </xcal:parameters>
2007          <xcal:date-time>201103031T000000002011-03-
2008 31T00:00:00</xcal:date-time>
2009      </xcal:dtend>
2010      </xcal:properties>
2011      <xcal:components>
2012          <xcal:available>
2013              <xs:type="xcal:properties">AvailableType">
2014          <xcal:properties>
2015              <xcal:dtstart>
2016              <xcal:dtstart>
2017              <xcal:parameters>
2018              <xcal:tzid>
2019          <xcal:text>America/New_York</xcal:text>
2020          </xcal:tzid>
2021          </xcal:parameters>
2022          <xcal:date>20110301T090000-time>2011-03-
2023 01T09:00:00</xcal:date-time>
2024      </xcal:dtstart>
2025      <xcal:dtend>
2026      </xcal:dtstart>
2027      <xcal:dtend>
2028          <xcal:parameters>
2029          <xcal:tzid>
2030          <xcal:text>America/New_York</xcal:text>
2031          </xcal:tzid>
2032          <xcal:dtstart>
2033          <xcal:dtstart>
2034          <xcal:parameters>
2035          <xcal:date>20110301T110000-time>2011-03-
2036 01T11:00:00</xcal:date-time>
2037      </xcal:dtend>
2038      <xcal:rrule>
2039          <xcal:recur>
2040              <xcal:freq>WEEKLY</xcal:freq>
2041              <xcal:byday>MO</xcal:byday>
2042              <xcal:byday>TU</xcal:byday>
2043              <xcal:byday>WE</xcal:byday>
2044              <xcal:byday>TH</xcal:byday>
2045          </xcal:recur>
2046      </xcal:rrule>
2047      <xcal:properties>
2048      </xcal:available>
2049          <xcal:available> xs:type="xcal:AvailableType">
2050          <xcal:properties>

```

```

2051          <xcal:dtstart>
2052          <xcal:properties>
2053          <xcal:dtstart>
2054          <xcal:parameters>
2055          <xcal:tzid>
2056
2057          <xcal:text>America/New_York</xcal:text>
2058          </xcal:tzid>
2059          </xcal:parameters>
2060          <xcal:date>20110301T150000</xcal:date><xcal:time>2011-03-
2061 01T15:00:00</xcal:date-time>
2062          </xcal:dtstart>
2063          <xcal:dtend>
2064          <xcal:parameters>
2065          <xcal:tzid>
2066
2067          <xcal:text>America/New_York</xcal:text>
2068          </xcal:tzid>
2069          </xcal:parameters>
2070          <xcal:date>20110301T160000</xcal:date><xcal:time>2011-03-
2071 01T16:00:00</xcal:date-time>
2072          </xcal:dtend>
2073          <xcal:rrule>
2074          <xcal:recur>
2075          <xcal:freq>WEEKLY</xcal:freq>
2076          <xcal:byday>FR</xcal:byday>
2077          </xcal:recur>
2078          </xcal:rrule>
2079          </xcal:properties>
2080          </xcal:available>
2081      </xcal:components>
2082  </xcal:vavailability>
```

The [Vavailability](#) above describes service availability for the month of March, 2011, i.e., it has a start date of March 1 and an end date of March 31. Within that period, there are two schedules, described by the two availability artifacts. The first specifies that starting on March 1, there is a window of 9-11 am, Eastern Time, on Monday, Tuesday, Wednesday, and Thursday each week. The second specifies another window of availability from 3:00 PM (15:00) to 4:00 PM (16:00) on Fridays. These schedules are each valid only through March 31, the dtEnd of the encompassing Vavailability. If neither date nor duration were specified, then the end of the schedules would be indefinite.

The example above uses daily schedules with a weekly recurrence. The full breadth of recurrence rules is described in [\[iCalendar\]](#).

3.3.5.1 Combining a Gluon and Availability.

Consider the school schedule in the partition example in [Example 3-13](#)[Section 3.3.4](#). Optimizing the expression of a Partition that is used in several examples. The school has a single valid start time, at 8:00. The service can be refined by advertising its Availability as beginning at 9:00 on the first day. Availability re-occurs on a weekly schedule, only on the weekdays Monday, Tuesday, Thursday, and Friday. Furthermore, the schedule can only be invoked during the Fall semester, from September 1, to December 15.

With a [granularity](#)[Granularity](#) of one hour set, the schedule can only begin on the time that the Availability begins, or at one hour intervals thereafter. If the Availability Window is only from 8:00 with a Duration of one hour, then the service is advertised only for a start at this hour.

The example below illustrates how to use the Vavailability object contained in a gluon to publish [availability](#)[Availability](#) on a pre-existing sequence.

Example 3-15 Gluon publishing availability of pre-existing sequence

```
<xcal:gluon>
```

```

2106 <xcal:gluon xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
2107   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
2108     <xcal:properties/>
2109       <xcal:uid>
2110         <xcal:parameters/>
2111           <xcal:text>4b8bcc8a-570a-4d23-8059-
2112 3f55b090da35@examples.oasis-open.org</xcal:text>
2113       </xcal:uid>
2114       <xcal:related-to>
2115         <xcal:parameters>
2116           <xcal:reltype>
2117             <xcal:text>CHILD</xcal:text>
2118           </xcal:reltype>
2119         </xcal:parameters>
2120         <xcal:uid>c7496e78-6d71-4118-b42a-641f1efe02a9@examples.oasis-
2121 open.org</xcal:uid>
2122       </xcal:related-to>
2123     </xcal:properties>
2124     <xcal:components>
2125       <xcal:vavailability>
2126         <xcal:properties>
2127           <xcal:uid>
2128             <xcal:text>e83ef824-aeb6-45ab-8bb0-
2129 8648a37e92f6@examples.oasis-open.org7cf50215-e50e-46a9-9377-
2130 797aff409ae3</xcal:text>
2131           </xcal:uid>
2132           <xcal:dtstart>
2133             <xcal:parameters>
2134               <xcal:tzid>
2135
2136             <xcal:text>America/New_York</xcal:text>
2137               </xcal:tzid>
2138             </xcal:parameters>
2139             <xcal:date-time>20110901T000000002011-09-
2140 01T00:00:00</xcal:date-time>
2141           </xcal:dtstart>
2142           <xcal:dtend>
2143             <xcal:parameters>
2144               <xcal:tzid>
2145
2146             <xcal:text>America/New_York</xcal:text>
2147               </xcal:tzid>
2148             </xcal:parameters>
2149             <xcal:date-time>20111201T000000002011-12-
2150 17T00:00:00</xcal:date-time>
2151           </xcal:dtend>
2152         </xcal:properties>
2153         <xcal:components>
2154           <xcal:available xs:type="xcal:AvailableType">
2155             <xcal:properties>
2156               <xcal:dtstart>
2157                 <xcal:parameters>
2158                   <xcal:tzid>
2159
2160             <xcal:text>America/New_York</xcal:text>
2161               </xcal:tzid>
2162             </xcal:parameters>
2163             <xcal:date>20110901T08000000-
2164 time>2011-09-01T08:00:00</xcal:date-time>
2165           </xcal:dtstart>
2166           <xcal:dtend>
2167             <xcal:parameters>
2168               <xcal:tzid>

```

```

2169
2170     <xcal:text>America/New_York</xcal:text>
2171     </xcal:tzid>
2172     </xcal:parameters>
2173     <xcal:date>20110901T09000000-
2174     time>2011-09-01T09:00:00</xcal:date-<time>
2175     </xcal:dtend>
2176     <xcal:rrule>
2177         <xcal:recur>
2178             <xcal:freq>WEEKLY</xcal:freq>
2179             <xcal:byday>MO</xcal:byday>
2180             <xcal:byday>WE</xcal:byday>
2181             <xcal:byday>FR</xcal:byday>
2182         </xcal:recur>
2183     </xcal:rrule>
2184     </xcal:properties>
2185     </xcal:available>
2186     </xcal:components>
2187     </xcal:vavailability>
2188     </xcal:components>
2189 </xcal:gluon>
```

2190 In the example above, the general classroom schedule has been referenced by a new gluon, and
2191 established the availability for the Fall semester. The new gluon references the pre-existing gluon that
2192 establishes the sequence as a partition.

2193 This double inheritance, in which a Sequence inherits from a **Calendar**-Gluon which inherits from a
2194 **Calendar** Gluon is a useful pattern for advertising or scheduling a service.

2195 3.3.6 Other Scheduling Scenarios

2196 Sometimes, the invoker of a service is interested only in single Interval of the Sequence, but the entire
2197 Sequence is required. In the example below, the second Interval is advertised, i.e., the **Calendar**-Gluon
2198 points to the second Interval. The first Interval might be a required ramp-period, during which the
2199 underlying process is “warming up”, and which may bring some lesser service to market during that ramp
2200 time. The ramp-down time at the end is similarly fixed. The entire Service offering is represented by the
2201 exposed (it has a public URI) **Calendar**-Gluon.

2202 Example 3-16: Standard Sequence with Ramp-Up and Ramp Down

```

2203 <?xml version="1.0" encoding="utf-16"?>
2204 <xcal:icalendar xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
2205   xmlns:xcal="urn:ietf:params:xml:ns:icalendar-2.0">
2206     <xcal:vcalendar xs:type="xcal:VcalendarType">
2207       <xcal:components>
2208         <xcal:gluon>
2209           <xcal:properties>
2210             <xcal:uid>
2211               <xcal:text>26e1fa7e-aeac-429d-ab7a-
2212               f6d92ef9afc20examples.oasis-open.orgfcdf9ebc7-32f4-4a8c-9dbd-
2213               23749e6b324e</xcal:text>
2214             </xcal:uid>
2215             <xcal:related-to>
2216               <xcal:parameters>
2217                 <xcal:uid>
2218                 <xcal:related-to>
2219                 <xcal:parameters>
2220                   <xcal:reltype>
2221
2222                   <xcal:text>CHILD</xcal:text>
2223                   <xcal:reltype>
2224               </xcal:parameters>
```

```

2225                                     <xcal:uid>429dddae-d6b8-418e-a897-
2226 d57e6e83052b@examples.oasis-open.org</xcal:uid>
2227 5a78a02d30b8</xcal:uid>
2228                                         </xcal:related-to>
2229                                         <xcal:dtstart>
2230                                         <xcal:parameters>
2231                                         <xcal:tzid>
2232
2233                                         <xcal:text>America/New_York</xcal:text>
2234                                         </xcal:tzid>
2235                                         </xcal:parameters>
2236                                         <xcal:parameters>
2237                                         <xcal:tzid>
2238                                         <xcal:text>America/New_York</xcal:text>
2239                                         </xcal:tzid>
2240                                         </xcal:parameters>
2241                                         <xcal:date-
2242 time>20110315T084500002011-05-28T08:45:00</xcal:date-time>
2243                                         </xcal:dtstart>
2244                                         <xcal:duration>
2245
2246                                         <xcal:duration>T2HPT2H</xcal:duration>
2247                                         </xcal:duration>
2248                                         </xcal:duration>
2249                                         <xcal:x-wscalendarwsCalendar-attach>
2250                                         <xcal:artifact>
2251                                         <***:payload xmlns:***="urn:not:a:real:artifact">
2252                                         <***>
2253                                         <payload:payload>
2254                                         <payload:units>fortnights</payload:units>
2255
2256                                         <payload:quantity>14</***></payload:quantity>
2257                                         </***>
2258                                         </payload:payload>
2259                                         </xcal:artifact>
2260                                         <xcal:x-wscalendarwsCalendar-attach>
2261                                         </xcal:properties>
2262                                         <xcal:components/>
2263                                         </xcal:gluon>
2264                                         <xcal:interval>
2265                                         <xcal:properties>
2266                                         <xcal:uid>
2267                                         <xcal:text>00564efc-975d-42a1-9f3a-
2268 eb4267e893500@examples.oasis-open.org</xcal:text>
2269                                         </xcal:uid>
2270                                         <xcal:duration>
2271                                         <xcal:duration>T10MPT10M</xcal:duration>
2272                                         </xcal:duration>
2273                                         </xcal:properties>
2274                                         </xcal:interval>
2275                                         <xcal:interval>
2276                                         <xcal:properties>
2277                                         <xcal:uid>
2278                                         <xcal:text>a4cde8b9-ed43-4ca7-9eb0-
2279 5a78a02d30b8</xcal:text>
2280                                         </xcal:uid>
2281                                         <xcal:related-to>
2282                                         <xcal:parameters>
2283                                         <xcal:reltype>
2284                                         <xcal:text>FS</xcal:text>
2285                                         </xcal:reltype>
2286                                         </xcal:parameters>

```

```

2287 <xcal:uid>c5473482-2ecb-4a9b-b53c-
2288 91ff5614dc40</xcal:uid>
2289 </xcal:related-to>
2290 <xcal:x-wscalendarwsCalendar-attach>
2291 <xcal:parameters />
2292 <xcal:artifaet>
2293 <xxx: payload
2294 xmlns:xx="urn:not:a:real:artifact">
2295 <xx:warmUp>fixed content</xx:warmUp>
2296 </xx:payload>
2297 </xcal:artifaet>
2298
2299 <payload:units>furlongs</payload:units>
2300 <payload:quantity>11</payload:quantity>
2301 </payload:payload>
2302 </xcal:x-wscalendarwsCalendar-attach>
2303 </xcal:properties>
2304 </xcal:interval>
2305 <xcal:interval>
2306 <xcal:properties>
2307 <xcal:uid>
2308 </xcal:properties>
2309 </xcal:interval>
2310 <xcal:interval>
2311 <xcal:properties>
2312 <xcal:uid>
2313 <xcal:text>429dddae-d6b8-418e-a897-
2314 d57e6e83052b@examples.oasis-open.orgde2aa95b-f930-44d5-bc2b-
2315 1356f8732879</xcal:text>
2316 </xcal:uid>
2317 <xcal:related-to>
2318 <xcal:parameters>
2319 <xcal:reltype>
2320 <xcal:text>FS</xcal:text>
2321 </xcal:reltype>
2322 </xcal:parameters>
2323 </xcal:uid>
2324 <xcal:related-to>
2325 <xcal:parameters>
2326 <xcal:reltype>
2327 <xcal:text>FS</xcal:text>
2328 </xcal:reltype>
2329 <xcal:parameters>
2330 <uid>a4cde8b9-ed43-4ca7-9eb0-5a78a02d30b8</xcal:uid>00564efe-975d-
2331 42a1-9f3a-eb4267e89350@examples.oasis-open.org</xcal:uid>
2332 </xcal:related-to>
2333 <xcal:duration>
2334 <xcal:duration>PT5M</xcal:duration>
2335 </xcal:duration>
2336 </xcal:x-wscalendarwsCalendar-attach>
2337 <xcal:artifaet>
2338 <xxx:payload xmlns:xx="urn:not:a:real:artifact">
2339 <xx: payload>
2340 <payload:units>furlongs</xxx:payload:units>
2341 </xxx:>
2342 <payload:quantity>11</payload:quantity>
2343 </xcal:artifaet>
2344 </payload:payload>
2345 </xcal:x-wscalendarwsCalendar-attach>
2346 </xcal:properties>
2347 </xcal:interval>
2348 </xcal:components>
2349 </xcal:vcalendar>
```

```

2398     </xcal:properties>
2399     </xcal:interval>
2400     <xcal:interval>
2401     <xcal:properties>
2402     <xcal:uid>
2403     <xcal:text>59e717e3-7330-4cf3-8d57-f3239e4bc254@examples.oasis-
2404 open.org</xcal:text>
2405     </xcal:uid>
2406     <xcal:duration>
2407     <xcal:duration>T5M</xcal:duration>
2408     </xcal:duration>
2409     <xcal:related-to>
2410     <xcal:parameters>
2411     <xcal:reltype>
2412     <xcal:text>FS</xcal:text>
2413     </xcal:reltype>
2414     </xcal:parameters>
2415     <xcal:uid>429dddaec-d6b8-418e-a897-d57c6e83052b@examples.oasis-
2416 open.org</xcal:uid>
2417     </xcal:related-to>
2418     <xcal:x-wscalendar-attach>
2419     <xcal:artifact>
2420     <xx:payload xmlns:xx="urn:not:a:real:artifact">
2421     <xx:coolDown>fixed content</xx:coolDown>
2422     </xx:payload>
2423     </xcal:artifact>
2424     </xcal:x-wscalendar-attach>
2425     </xcal:properties>
2426     </xcal:interval>
2427     </xcal:components>
2428   </xcal:vcalendar>
2429 </xcal:icalendar>

```

2430 The underlying sequence has a fixed warm up and cool down (intervals 1 and 3). The Gluon shares a
2431 payload with Interval 2, which has no duration. Interval 2 inherits the quantity (14) and the duration (2H)
2432 from the Gluon.

2433 If expressed all at once, the Gluon merely provides a handle for the Sequence. A more useful expression
2434 would have the Gluon separate, or perhaps inheriting its information from a market agreement. This
2435 enables the service interaction to express that Start Time, Duration and Quantity. All three are inherited,
2436 in this case, only by the Designated Interval.

2437 3.4 Time Stamps

2438 Time stamps are used everywhere in inter-domain service performance analysis and have particular use
2439 in smart grids to support event forensics. Time stamps are often may be assembled and collated from
2440 events across multiple time zones and from multiple systems.

2441 Different systems may track time and therefore record events with different levels of Tolerance. It is not
2442 unusual for a time-stamped event from a domain with low Tolerance to appear to have occurred
2443 after one or more time-stamped events from a domain with high Tolerance time-stamps that it caused. A
2444 fully qualified time-stamp includes the granularity measure.

2445 *Table 3-7: Aspects Elements of Time Stamps*

Time Stamp Element	Definition (Normative)	Note (Non-Normative)	Deleted
--------------------	------------------------	----------------------	---------

Time Stamp Element	Definition (Normative)	Note (Non-Normative)
timestamp <u>Time Stamp</u>	WS-Calendar:time A fully <ins>Fully</ins> qualified date and time of event. Mandatory.	May include two objects as defined above.
accuracy <u>Accuracy</u>	A Duration defining the accuracy of the TimeStamp value. Mandatory.	Identifies whether an interval of a particular duration (resp. starting at a particular time) is indeed an interval of the mentioned duration plus or minus some number of milliseconds, seconds and minutes (resp. an interval starting at the mentioned time plus or minus some number of milliseconds, seconds and minutes.)
timeStampR <u>realmTime</u> Stamp <u>Realm</u>	<p>Of type Uri, shall identify the system where the TimeStamp value originated. The value of this element shall be set by:</p> <ul style="list-style-type: none"> • The component at the realm border in a particular inter-domain interaction or, • By any component able to accurately set it within a system or sub-system. <p>In the latter case, nothing prevents the component at the realm border to overwrite it without any notice. Optional.</p>	<p>A set of points identifies the system where the TimeStamp value originated. A set of recordings 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.</p> <p>A system border is crossed in an interaction when the 2 communication partners are not synchronized based on the same time source.</p> <p>See the example below for more information.</p>

Deleted

Deleted

Deleted

Time Stamp Element	Definition (Normative)	Note (Non-Normative)
<u>leapSeconds KnownLeapSeconds Known</u>	xs:bool 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.
<u>clockFailure Clock Failure</u>	xs:bool 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 This may put in doubt the advisability of direct comparison of this timestamp should be ignored information from this system and from foreign systems.
<u>clockNotSynchronized Clock Not Synchronized</u>	xs:bool 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.
<u>timeSourceAccuracy Time Source Accuracy</u>	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.

Deleted C

2450 3.4.1 Time Stamp Realm Discussion

2451 Within a single system, or synchronized system of systems, one can sort the temporal order of event by
 2452 sorting them by TimeStamp. Determining the order of events is the first step of event forensics. This
 2453 assumption does not apply when events are gathered across systems.

2454 Different systems may not have synchronized time, or may synchronize time against different sources.
2455 This means different system clocks may drift apart. It may be that a later timestamp from one system
2456 occurred before an earlier timestamp in another. As this drift is unknown, it cannot be automatically
2457 corrected for without additional information.

2458 | The TimeStampRealm element identifies which system created an event time-stamp. The
2459 | TimeStampRealm identifies a source system in inter-domain interactions (a system of systems). For
2460 | example: <http://SystemA.com> and <http://SystemB.com>
2461 identify 2 systems. This example assumes SystemA and SystemB do not have a common time source.

2462 | The TimeStampRealm can also be used to identify sub-systems in intra-domain interactions (sub-systems
2463 | of a system). For example: <http://SystemA.com/SubSystem1> and
2464 | <http://SystemA.com/SubSystem2> identify 2 subsystems of the same
2465 | higher level system. In cases where the upper level SystemA does not have a global time source for
2466 | synchronizing all of its sub-systems, it can be useful to identify sub-systems in such a way this
2467 | manner.

2468 |

2469 4 PART TWO: Calendar Update and Synchronization 2470 with RESTful Services

2471 4.1 Calendar Services

2472 The Service interactions are built upon and make the same assumptions about structure as the CalDAV
2473 protocol defined in [RFC4791] and related specifications. It does NOT require nor assume the WebDAV
2474 nor CalDAV protocol but does make use of some of the same elements and structures in the CalDAV
2475 XML namespace.

2476 Calendar resources, for example events and tasks are stored as named resources (files) inside special
2477 collections (folders) known as "Calendar Collections".

2478 These services can be looked upon as a layer built on top of CalDAV and defines the basic operations
2479 which allow creation, retrieval, update and deletion. In addition, query, and free-busy operations are
2480 defined to allow efficient, partial retrieval of calendar data.

2481 These services assume a degree of conformity with CalDAV is established such that services built in that
2482 manner do not have a significant mismatch. It is assumed that some WS-Calendar services will be built
2483 without any CalDAV support.

2484 4.1.1 Overview of the protocol

2485 The protocol is an HTTP based RESTfull protocol using a limited set of methods. Each request may be
2486 followed by a response containing status information.

2487 The following methods are specified in the protocol description, PUT, POST, GET, DELETE. To avoid
2488 various issues with certain methods being blocked clients may use the X-HTTP-Method-Override: header
2489 to specify the intended operation. Servers SHOULD behave as if the named method was used.

```
2490 POST /user/fred/calendar/ HTTP/1.1
2491 ...
2492 X-HTTP-Method-Override: PUT
2493 Properties
```

2494 A service or resource will have a number of properties which describe the current state of that service or
2495 resource. These properties are accessed through a GET on the target resource or service with an
2496 ACCEPT header specifying application/xrd+xml. See Section 4.1.1.3.6

2497 The following operations are defined by this specification:

- 2498 • Retrieval and update of service and resource properties
- 2499 • Creation of a calendar object
- 2500 • Retrieval of a calendar object
- 2501 • Update of a calendar object
- 2502 • Deletion of a calendar object
- 2503 • Query
- 2504 • Free-busy query

2505 4.1.1.1 Calendar Object Resources

2506 The same restrictions apply to Calendar Object Resources as specified in CalDAV [RFC4791] section
2507 4.2. An additional constraint for CalWS is that no timezone specifications are transferred.

2508	4.1.1.2 Timezone information
2509	It is assumed that the client and server each have access to a full set of up-to-date timezone information.
2510	Timezones will be referenced by a timezone identifier from the full set of Olson data together with a set of well-known aliases defined [TZDB]. CalWS services may advertise themselves as timezone servers
2511	through the server properties object.
2512	
2513	4.1.1.3 Issues not addressed by this specification.
2514	A number of issues are not addressed by this version of the specification, either because they should be
2515	addressed elsewhere or will be addressed at some later date.
2516	4.1.1.3.1 Access Control
2517	It is assumed that the targeted server will set an appropriate level of access based on authentication. This
2518	specification will not attempt to address the issues of sharing or Access Control Lists (ACLs).
2519	4.1.1.3.2 Provisioning
2520	The protocol will not provide any explicit provisioning operations. If it is possible to authenticate or
2521	address a principal's calendar resources then they MUST be automatically created if necessary or
2522	appropriate
2523	4.1.1.3.3 Copy/Move
2524	These operations are not yet defined for this version of the CalWS protocol. Both operations raise a
2525	number of issues. In particular implementing a move operation through a series of retrievals, insertions
2526	and deletions may cause undesirable side effects. Both these operations will be defined in a later version
2527	of this specification.
2528	4.1.1.3.4 Creating Collections
2529	We will not address the issue of creating collections within the address space. The initial set is created by
2530	provisioning.
2531	4.1.1.3.5 Retrieving collections
2532	This operation is currently undefined. A GET on a collection may fail or return a complete calendar object
2533	representing the collection.
2534	4.1.1.3.6 Setting service and resource properties.
2535	These operations are not defined in this version of the specification. In the future it will be possible to
2536	define or set the properties for the service or resources within the service.
2537	4.1.1.4 CalWS Glossary
2538	4.1.1.4.1 Hrefs
2539	An href is a URI reference to a resource, for example
2540	<del style="background-color: #f0f0f0;">"http://example.org/user/fred/calendar/event1.ics".
2541	The URL above reflects a possible structure for a calendar server. All URLs should be absolute or path-
2542	absolute following the rules defined in RFC4918 Section 8.3.

2543	4.1.1.4.2 Calendar Object Resource
2544	A calendar object resource is an event, meeting or a task. Attachments are resources but NOT calendar object resources. An event or task with overrides is a single calendar resource entity.
2545	
2546	4.1.1.4.3 Calendar Collection
2547	A folder only allowed to contain calendar object resources.
2548	4.1.1.4.4 Scheduling Calendar Collection
2549	A folder only allowed to contain calendar resources which is also used for scheduling operations.
2550	Scheduling events placed in such a collection will trigger implicit scheduling activity on the server.
2551	4.1.1.4.5 Principal Home
2552	The collection under which all the resources for a given principal are stored. For example, for principal "fred" the principal home might be "/user/fred/"
2553	
2554	4.1.2 Error conditions
2555	Each operation on the calendar system has a number of pre-conditions and post-conditions that apply.
2556	A "precondition" for a method describes the state of the server that must be true for that method to be performed.
2557	A "post-condition" of a method describes the state of the server that must be true after that method has been completed. Any violation of these conditions will result in an error response in the form of a CalWS XML error element containing the violated condition and an optional description.\
2558	
2559	
2560	Each method specification defines the preconditions that must be satisfied before the method can succeed. A number of post-conditions are generally specified which define the state that must exist after the execution of the operation. Preconditions and post-conditions are defined as error elements in the CalWS XML namespace.
2561	
2562	
2563	
2564	4.1.2.1 Example: error with CalDAV error condition
2565	<pre><?xml version="1.0" encoding="utf-8" xmlns:CW="Error! Reference source not found." xmlns:C="urn:ietf:params:xml:ns:caldav" ?> <CW:error> <C:supported-filter> <C:prop-filter name="X-ABC-CUID"/> </C:supported-filter> <CW:description>Unknown property </CW:description> </CW:error></pre>
2566	
2567	
2568	
2569	
2570	
2571	
2572	
2573	
2574	4.2 Properties and link relations
2575	4.2.1 Property and relation-type URIs
2576	In the XRD entity returned properties and related services and entities are defined by absolute URIs
2577	which correspond to the extended relation type defined in [web-linking] Section 4.2. These URIs do NOT
2578	correspond to any real entity on the server and clients should not attempt to retrieve any data at that target.
2579	
2580	Certain of these property URIs correspond to CalDAV preconditions. Each URL is prefixed by the CalWS
2581	relations and properties namespace http://docs.oasis-open.org/ns/wscal/calws . These properties which
2582	correspond to CalDAV properties have the additional path element "caldav/", for example
2583	http://docs.oasis-open.org/ns/wscal/calws/caldav/supported-calendar-data
2584	corresponds to

2585 ~~CalDAV:supported-calendar-data~~

2586 In addition to those CalDAV properties, the CalWS specification defines a number of other properties and
2587 link relations with the URI prefix of <http://docs.oasis-open.org/ns/wscal/calws>.

2588 **4.2.2 supported-features property.**

2589 <http://docs.oasis-open.org/ns/wscal/calws/supported-features>

2590 This property defines the features supported by the target. All resources contained and managed by the
2591 service should return this property. The value is a comma-separated list containing one or more of the
2592 following

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

2594 ~~<Property type="http://docs.oasis-open.org/ns/wscal/calws/supported-features">~~
2595 ~~>calendar-access</Property>~~

2596 **4.2.3 max-attendees-per-instance**

2597 <http://docs.oasis-open.org/ns/wscal/calws/max-attendees-per-instance>

2598 Defines the maximum number of attendees allowed per event or task.

2599 **4.2.4 max-date-time**

2600 <http://docs.oasis-open.org/ns/wscal/calws/max-date-time>

2601 Defines the maximum date/time allowed on an event or task

2602 **4.2.5 max-instances**

2603 <http://docs.oasis-open.org/ns/wscal/calws/max-instances>

2604 Defines the maximum number of instances allowed per event or task

2605 **4.2.6 max-resource-size**

2606 <http://docs.oasis-open.org/ns/wscal/calws/max-resource-size>

2607 Provides a numeric value indicating the maximum size of a resource in octets that the server is willing to
2608 accept when a calendar object resource is stored in a calendar collection.

2609 **4.2.7 min-date-time**

2610 <http://docs.oasis-open.org/ns/wscal/calws/min-date-time>

2611 Provides a DATE-TIME value indicating the earliest date and time (in UTC) that the server is willing to
2612 accept for any DATE or DATE-TIME value in a calendar object resource stored in a calendar collection.

2613 **4.2.8 description**

2614 <http://docs.oasis-open.org/ns/wscal/calws/description>

2615 Provides some descriptive text for the targeted collection.

2616 **4.2.9 timezone-service relation.**

2617 <http://docs.oasis-open.org/ns/wscal/calws/timezone-service>

2618 The location of a timezone service used to retrieve timezone information and specifications. This may be
2619 an absolute URL referencing some other service or a relative URL if the current server also provides a
2620 timezone service.

```
2621 <Link rel="http://docs.oasis-open.org/ns/wscal/calws/timezone-service"  
2622   href="http://example.com/tz" />
```

4.2.10 principal-home relation.

<http://docs.oasis-open.org/ns/wscal/calws/principal-home>

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

```
2626 <Link rel="http://docs.oasis-open.org/ns/wscal/calws/principal-home"  
2627   href="http://example.com/user/fred" />
```

4.2.11 current-principal-freebusy relation.

<http://docs.oasis-open.org/ns/wscal/calws/current-principal-freebusy>

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

```
2631 <Link rel="http://docs.oasis-open.org/ns/wscal/calws/current-principal-freebusy"  
2632   href="http://example.com/freebusy/user/fred" />
```

4.2.12 principal-freebusy relation.

<http://docs.oasis-open.org/ns/wscal/calws/principal-freebusy>

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

```
2636 <Link rel="http://docs.oasis-open.org/ns/wscal/calws/principal-freebusy"  
2637   href="http://example.com/freebusy" />
```

4.2.13 child-collection relation.

<http://docs.oasis-open.org/ns/wscal/calws/child-collection>

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

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

```
2645 <Link rel="http://http://docs.oasis-open.org/ns/wscal/calws/child-collection"  
2646   href="http://example.com/calws/user/fred/calendar">  
2647   <Title xml:lang="en">Calendar</Title>  
2648   <Property type="http://docs.oasis-open.org/ns/wscal/calws/collection"  
2649     xsi:nil="true" />  
2650   <Property type="http://docs.oasis-open.org/ns/wscal/calws/calendar-  
2651     collection"  
2652     xsi:nil="true" />  
2653 </Link>
```

4.2.14 created link property

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

Appears within a link relation describing collections or entities. The value is a date-time as defined in [RFC3339](#) Section 5.6

```
2658 <Property type="http://docs.oasis-open.org/ns/wscal/calws/created"  
2659   >1985-04-12T23:20:50.52Z</Property>
```

4.2.15 last-modified property

<http://docs.oasis-open.org/ns/wscal/calws/last-modified>

2662	Appears within an xrd object describing collections or entities. The value is the same format as would appear in the Last-Modified header and is defined in [RFC2616], Section 3.3.1.
2663	<pre><Property type="http://docs.oasis-open.org/ns/wscal/calws/last-modified"> <!--Mon, 12 Jan 1998 09:25:56 GMT--></Property></pre>
2664	
2665	
2666	<h3>4.2.16 displayname property</h3>
2667	<pre>http://docs.oasis-open.org/ns/wscal/calws/displayname</pre>
2668	Appears within an xrd object describing collections or entities. The value is a localized name for the entity or collection.
2669	
2670	<pre><Property type="http://docs.oasis-open.org/ns/wscal/calws/displayname"> <!--My Calendar--></Property></pre>
2671	
2672	<h3>4.2.17 timezone property</h3>
2673	<pre>http://docs.oasis-open.org/ns/wscal/calws/timezone</pre>
2674	Appears within an xrd object describing collections. The value is a text timezone identifier.
2675	<pre><Property type="http://docs.oasis-open.org/ns/wscal/calws/timezone"> <!--America/New_York--></Property></pre>
2676	
2677	<h3>4.2.18 owner property</h3>
2678	<pre>http://docs.oasis-open.org/ns/wscal/calws/owner</pre>
2679	Appears within an xrd object describing collections or entities. The value is a server specific uri.
2680	<pre><Property type="http://docs.oasis-open.org/ns/wscal/calws/owner"> <!--/principals/users/mike--></Property></pre>
2681	
2682	<h3>4.2.19 collection link property</h3>
2683	<pre>http://docs.oasis-open.org/ns/wscal/calws/collection</pre>
2684	Appears within a link relation describing collections or entities. The property takes no value and indicates that this child element is a collection.
2685	
2686	<pre><Property type="http://docs.oasis-open.org/ns/wscal/calws/collection"> <!--xsi:nil="true"--></pre>
2687	
2688	<h3>4.2.20 calendar-collection link property</h3>
2689	<pre>http://docs.oasis-open.org/ns/wscal/calws/calendar-collection</pre>
2690	Appears within a link relation describing collections or entities. The property takes no value and indicates that this child element is a calendar collection.
2691	
2692	<pre><Property type="http://docs.oasis-open.org/ns/wscal/calws/calendar-collection"> <!--xsi:nil="true"--></pre>
2693	
2694	<h3>4.2.21 CalWS:privilege-set XML element</h3>
2695	<pre>http://docs.oasis-open.org/ns/wscal/calws:privilege-set</pre>
2696	Appears within a link relation describing collections or entities and specifies the set of privileges allowed to the current authenticated principal for that collection or entity.
2697	
2698	<pre><!ELEMENT calws:privilege-set (calws:privilege*)> <!ELEMENT calws:privilege ANY></pre>
2699	
2700	Each privilege element defines a privilege or access right. The following set is currently defined
2701	<ul style="list-style-type: none"> • CalWS: Read - current principal has read access

2702 • CalWS: Write — current principal has write access
2703 <calWS:privilege-set>
2704 <calWS:privilege><calWS:read></calWS:privilege>
2705 <calWS:privilege><calWS:write></calWS:privilege>
2706 </calWS:privilege-set>

2707 **4.3 Retrieving Collection and Service Properties**

2708 Properties, related services and locations are obtained from the service or from service resources in the
2709 form of an XRD document as defined by [XRD-1.0].

2710 Given the URL of a CalWS service a client retrieves the service XRD document through a GET on the
2711 service URL with an ACCEPT header specifying application/xrd+xml.

2712 Retrieving resource properties is identical to obtaining service properties, that is, execute a GET on the
2713 target URL with an ACCEPT header specifying application/xrd+xml.

2714 The service properties define the global limits and defaults. Any properties defined on collections within
2715 the service hierarchy override those service defaults. The service may choose to prevent such overriding
2716 of defaults and limits when appropriate.

2717 **4.3.1 Request parameters**

- 2718 • None

2719 **4.3.2 Responses:**

- 2720 • 200: OK
2721 • 403: Forbidden
2722 • 404: Not found

2723 **4.3.3 Example - retrieving server properties:**

```
2724 >>>Request
2725
2726   GET / HTTP/1.1
2727   Host: example.com
2728   ACCEPT:application/xrd+xml
2729
2730 >>>Response
2731   <XRD xmlns="http://docs.oasis-open.org/ns/xri/xrd-1.0"
2732     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
2733     <Expires>1970-01-01T00:00:00Z</Expires>
2734     <Subject>http://example.com/calws</Subject>
2735     <Property type="http://docs.oasis-open.org/ns/wscal/calws/created">
2736       >1970-01-01</Property>
2737
2738     <Link rel="http://docs.oasis-open.org/ns/wscal/calws/timezone-service"
2739       href="http://example.com/tz" />
2740
2741     <calWS:privilege-set>
2742       <calWS:privilege><calWS:read></calWS:privilege>
2743     </calWS:privilege-set>
2744
2745     <Link rel="http://docs.oasis-open.org/ns/wscal/calws/principal-home"
2746       type="collection"
2747       href="http://example.com/calws/user/fred">
2748       <Title xml:lang="en">Fred's calendar home</Title>
2749     </Link>
```

```

2751 <Link rel="http://docs.oasis-open.org/ns/wscl/calws/child-collection"
2752   type="calendar,scheduling"
2753   href="http://example.com/calws/user/fred/calendar">
2754 <Title xml:lang="en">Calendar</Title>
2755 </Link>
2756
2757 <Property type="http://docs.oasis-open.org/ns/wscl/calws/max-instances">
2758   >1000</Property>
2759
2760 <Property type="http://docs.oasis-open.org/ns/wscl/calws/max-attendees-
2761 per-instance">
2762   >100</Property>
2763
2764 </XRD>
2765

```

4.4 Creating Calendar Object Resources

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

4.4.1 Request parameters

- action=create

4.4.2 Responses:

- 201:created
- 403:Forbidden - no access

4.4.3 Preconditions for Calendar Object Creation

- **CalWS:target-exists:** The target of a PUT must exist. Use POST to create entities and PUT to update them.
- **CalWS:not-calendar-data:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST be a supported media type (i.e., iCalendar) for calendar object resources;
- **CalWS:invalid-calendar-data:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST be valid data for the media type being specified (i.e., MUST contain valid iCalendar data);
- **CalWS:invalid-calendar-object-resource:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST obey all restrictions specified in Calendar Object Resources (e.g., calendar object resources MUST NOT contain more than one type of calendar component, calendar object resources MUST NOT specify the iCalendar METHOD property, etc.);
- **CalWS:unsupported-calendar-component:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST contain a type of calendar component that is supported in the targeted calendar collection;
- **CalWS:uid-conflict:** The resource submitted in the PUT request, or targeted by a COPY or MOVE request, MUST NOT specify an iCalendar UID property value already in use in the targeted calendar collection or overwrite an existing calendar object resource with one that has a different UID property value. Servers SHOULD report the URL of the resource that is already making use of the same UID property value in the CalWS:href element

—<!ELEMENT uid-conflict (CalWS:href)>

- 2796 • **CalWS:invalid-calendar-collection-location:** In a COPY or MOVE request, when the Request-
 2797 URI is a calendar collection, the Destination-URI MUST identify a location where a calendar
 2798 collection can be created;
- 2799 • **CalWS:exceeds-max-resource-size:** The resource submitted in the PUT request, or targeted by a
 2800 COPY or MOVE request, MUST have an octet size less than or equal to the value of the
 2801 CalDAV:max-resource-size property value on the calendar collection where the resource will be
 2802 stored;
- 2803 • **CalWS:before-min-date-time:** The resource submitted in the PUT request, or targeted by a COPY
 2804 or MOVE request, MUST have all of its iCalendar DATE or DATE-TIME property values (for each
 2805 recurring instance) greater than or equal to the value of the CalDAV:min-date-time property value
 2806 on the calendar collection where the resource will be stored;
- 2807 • **CalWS:after-max-date-time:** The resource submitted in the PUT request, or targeted by a COPY
 2808 or MOVE request, MUST have all of its iCalendar DATE or DATE-TIME property values (for each
 2809 recurring instance) less than the value of the CalDAV:max-date-time property value on the calendar
 2810 collection where the resource will be stored;
- 2811 • **CalWS:too-many-instances:** The resource submitted in the PUT request, or targeted by a COPY
 2812 or MOVE request, MUST generate a number of recurring instances less than or equal to the value
 2813 of the CalDAV:max-instances property value on the calendar collection where the resource will be
 2814 stored;
- 2815 • **CalWS:too-many-attendees-per-instance:** The resource submitted in the PUT request, or
 2816 targeted by a COPY or MOVE request, MUST have a number of ATTENDEE properties on any one
 2817 instance less than or equal to the value of the CalDAV:max-attendees-per-instance property value
 2818 on the calendar collection where the resource will be stored;

2819 **4.4.4 Example - successful POST:**

```

2820      >>>Request
2821
2822      POST /user/fred/calendar/?action=create HTTP/1.1
2823      Host: example.com
2824      Content-Type: application/xml+calendar; charset="utf-8"
2825      Content-Length: ?
2826
2827      <?xml version="1.0" encoding="utf-8" ?>
2828      <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
2829          <vcalendar>
2830              ...
2831          </vcalendar>
2832      </icalendar>
2833
2834      >>>Response
2835
2836      HTTP/1.1 201 Created
2837      Location: http://example.com/user/fred/calendar/event1.ics

```

2838 **4.4.5 Example - unsuccessful POST:**

```

2839      >>>Request
2840
2841      POST /user/fred/readcalendar/?action=create HTTP/1.1
2842      Host: example.com
2843      Content-Type: text/text; charset="utf-8"
2844      Content-Length: ?
2845
2846      This is not an xml calendar object
2847

```

```
2848    >>>Response  
2849  
2850        HTTP/1.1 403 Forbidden  
2851        <?xml version="1.0" encoding="utf-8"?>  
2852        xmlns:D="DAV;"  
2853        xmlns:C="urn:ietf:params:xml:ns:caldaav"?>  
2854        <D:error>  
2855            <C:supported-calendar-data/>  
2856            <D:description>Not an icalendar object</D:description>  
2857        </D:error>
```

2858 **4.5 Retrieving resources**

2859 A simple GET on the href will return a named resource. If that resource is a recurring event or task with
2860 overrides, the entire set will be returned. The desired format is specified in the ACCEPT header. The
2861 default form is application/xml+calendar

2862 **4.5.1 Request parameters**

2863 • none

2864 **4.5.2 Responses:**

- 2865 • 200: OK
- 2866 • 403: Forbidden - no access
- 2867 • 406 The requested format specified in the accept header is not supported.

2868 **4.5.3 Example - successful fetch:**

```
2869 >>>Request  
2870  
2871     GET /user/fred/calendar/event1.ics HTTP/1.1  
2872     Host: example.com  
2873  
2874 >>>Response  
2875  
2876     HTTP/1.1 200 OK  
2877     Content-Type: application/xml+calendar; charset="utf-8"  
2878     Content-Length: ?  
2879  
2880     <?xml version="1.0" encoding="utf-8"?>  
2881     <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">  
2882         <vecalendar>  
2883             ...  
2884         </vecalendar>  
2885     </icalendar>
```

2886 **4.5.4 Example - unsuccessful fetch:**

```
2887 >>>Request  
2888  
2889     PUT /user/fred/calendar/noevent1.ics HTTP/1.1  
2890     Host: example.com  
2891  
2892 >>>Response  
2893  
2894     HTTP/1.1 404 Not found
```

2895 4.6 Updating resources

2896 Resources are updated with the PUT method targeted at the resource href. The body of the request
2897 contains a complete new resource which effectively replaces the targeted resource. To allow for optimistic
2898 locking of the resource use the if-match header.

2899 When updating a recurring event all overrides and master must be supplied as part of the content.

2900 Preconditions as specified in Section 4.4.3 are applicable.

2901 4.6.1 Responses:

- 2902 • 200: OK
- 2903 • 304: Not modified entity was modified by some other request
- 2904 • 403: Forbidden no access, does not exist etc. See error response

2906 Example 4.1: Successful Update

```
2907 >>>Request
2908
2909 PUT /user/fred/calendar/event1.ics HTTP/1.1
2910 Host: example.com
2911 Content-Type: application/xml+calendar; charset="utf-8"
2912 Content-Length: ?
2913
2914 <?xml version="1.0" encoding="utf-8" ?>
2915 <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
2916   <vcalendar>
2917     ...
2918   </vcalendar>
2919 </icalendar>
2920
2921 >>>Response
2922
2923 HTTP/1.1 200 OK
```

2924 Example 4.2: Unsuccessful Update

```
2925 >>>Request
2926
2927 PUT /user/fred/readcalendar/event1.ics HTTP/1.1
2928 Host: example.com
2929 Content-Type: application/xml+calendar; charset="utf-8"
2930 Content-Length: ?
2931
2932 <?xml version="1.0" encoding="utf-8" ?>
2933 <icalendar xmlns="urn:ietf:params:xml:ns:icalendar-2.0">
2934   <vcalendar>
2935   ...
2936   </vcalendar>
2937 </icalendar>
2938
2939 >>>Response
2940
2941 HTTP/1.1 403 Forbidden
2942 Content-Type: application/xml; charset="utf-8"
2943 Content-Length: xxxx
2944
2945 <?xml version="1.0" encoding="utf-8"
2946   xmlns:D="DAV:"?
2947   xmlns:CW=" http://docs.oasis-open.org/ws-calendar/CalWS" ?>
```

```
2948 <CW:error>
2949   <CW:target-exists/>
2950   <CW:description>Target of update must exist</C:description>
2951 </CW:error>
```

2952 **4.7 Deletion of resources**

2953 Delete is defined in [RFC 2616] Section 9.7. In addition to conditions defined in that specification, servers
2954 must remove any references from the deleted resource to other resources. Resources are deleted with
2955 the DELETE method targeted at the resource URL. After a successful completion of a deletion a GET on
2956 that URL must result in a 404 Not Found status.

2957 **4.7.1 Delete for Collections**

2958 Delete for collections may or may not be supported by the server. Certain collections are considered
2959 undeletable. On a successful deletion of a collection all contained resources to any depth must also be
2960 deleted.

2961 **4.7.2 Responses:**

- 2962 • 200: OK
- 2963 • 403: Forbidden – no access
- 2964 • 404: Not Found

2965 **4.8 Querying calendar resources**

2966 Querying provides a mechanism by which information can be obtained from the service through possibly
2967 complex queries. A list of icalendar properties can be specified to limit the amount of information returned
2968 to the client. A query takes the parts

- 2969 • Limitations on the data returned
- 2970 • Selection of the data
- 2971 • Optional timezone id for floating time calculations.

2972 The current specification uses CalDAV multiget and calendar-query XML bodies as specified in [RFC
2973 4791] with certain limitations and differences.

- 2974 1. The POST method is used for all requests, the action being identified by the outer element.
- 2975 2. While CalDAV servers generally only support [RFC 5545] and assume that as the default, the
2976 delivery format for CalWS will, by default, be [draft-xcal].
- 2977 3. The CalDAV query allows the specification of a number of DAV properties. Specification of these
2978 properties, with the exception of DAV:getetag, is considered an error in CalWS.
- 2979 4. The CalDAV:propnames element is invalid.

2980 With those differences, the CalDAV specification is the normative reference for this operation.

2981 **4.8.1 Limiting data returned**

2982 This is achieved by specifying one of the following

- 2983 • CalDAV:allprop return all properties (some properties are specified as not being part of the allprop
2984 set so are not returned)
- 2985 • CalDAV:prop An element which contains a list of properties to be returned . May only contain
2986 DAV:getetag and CalDAV:calendar-data

2987 Of particular interest, and complexity, is the calendar-data property which can contain a time range to limit
2988 the range of recurrences returned and/or a list of calendar properties to return.

2989 4.8.2 Pre/postconditions for calendar queries

2990 The preconditions as defined in in [RFC 4791] Section 7.8 apply here. CalDav errors may be reported by
2991 the service when preconditions or postconditions are violated.

2992 4.8.3 Example: time range limited retrieval

2993 This example shows the time-range limited retrieval from a calendar which results in 2 events, one a
2994 recurring event and one a simple non-recurring event.

```
2995 >> Request <<  
2996  
2997 POST /user/fred/calendar/ HTTP/1.1  
2998 Host: calws.example.com  
2999 Depth: 1  
3000 Content-Type: application/xml; charset="utf-8"  
3001 Content-Length: xxxx  
3002  
3003 <?xml version="1.0" encoding="utf-8" ?>  
3004 <C:calendar-query xmlns:D="DAV:"  
3005             xmlns:C="urn:ietf:params:xml:ns:caldav">  
3006 <D:prop>  
3007     <D:getetag/>  
3008     <C:calendar-data content-type="application/xml+calendar">  
3009         <C:comp name="VCALENDAR">  
3010             <C:prop name="VERSION"/>  
3011             <C:comp name="VEVENT">  
3012                 <C:prop name="SUMMARY"/>  
3013                 <C:prop name="UID"/>  
3014                 <C:prop name="DTSTART"/>  
3015                 <C:prop name="DTEND"/>  
3016                 <C:prop name="DURATION"/>  
3017                 <C:prop name="RRULE"/>  
3018                 <C:prop name="RDATE"/>  
3019                 <C:prop name="EXRULE"/>  
3020                 <C:prop name="EXDATE"/>  
3021                 <C:prop name="RECURRENCE-ID"/>  
3022             </C:comp>  
3023         </C:comp>  
3024     </C:calendar-data>  
3025 </D:prop>  
3026 <C:filter>  
3027     <C:comp-filter name="VCALENDAR">  
3028         <C:comp-filter name="VEVENT">  
3029             <C:time-range start="20060104T000000Z"  
3030                 end="20060105T000000Z"/>  
3031     </C:comp-filter>  
3032 </C:comp-filter>  
3033 </C:filter>  
3034 </C:calendar-query>  
3035  
3036 >> Response <<  
3037  
3038 HTTP/1.1 207 Multi-Status  
3039 Date: Sat, 11 Nov 2006 09:32:12 GMT  
3040 Content-Type: application/xml; charset="utf-8"  
3041 Content-Length: xxxx  
3042  
3043 <?xml version="1.0" encoding="utf-8" ?>
```

```

3044 <D:multistatus xmlns:D="DAV:">
3045   xmlns:C="urn:ietf:params:xml:ns:caldav">
3046   <D:response>
3047     <D:href>http://cal.example.com/bernard/work/abed2.ics</D:href>
3048     <D:propstat>
3049       <D:prop>
3050         <D:getetag>"fffff-abed2"</D:getetag>
3051         <C:calendar-data content-type="application/xml+calendar">
3052           <xc:icalendar>
3053             xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
3054             <xc:vcalendar>
3055               <xc:properties>
3056                 <xc:calscale><text>GREGORIAN</text></xc:calscale>
3057                 <xc:prodid>
3058                   <xc:text>//Example Inc.//Example Calendar//EN</xc:text>
3059                 </xc:prodid>
3060                 <xc:version><xc:text>2.0</xc:text></xc:version>
3061               </xc:properties>
3062               <xc:components>
3063                 <xc:vevent>
3064                   <xc:properties>
3065                     <xc:dtstart>
3066                       <xc:parameters>
3067                         <xc:tzid>US/Eastern<xc:tzid>
3068                       <xc:parameters>
3069                         <xc:date-time>20060102T120000</xc:date-time>
3070                       </xc:dtstart>
3071                       <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
3072                     <xc:summary>
3073                       <xc:text>Event #2</xc:text>
3074                     </xc:summary>
3075                     <xc:uid>
3076                       <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
3077                     </xc:uid>
3078                     <xc:rrule>
3079                       <xc:recur>
3080                         <xc:freq>DAILY</xc:freq>
3081                         <xc:count>5</xc:count>
3082                       </xc:recur>
3083                     </xc:rrule>
3084                   </xc:properties>
3085                 </xc:vevent>
3086
3087                 <xc:vevent>
3088                   <xc:properties>
3089                     <xc:dtstart>
3090                       <xc:parameters>
3091                         <xc:tzid>US/Eastern<xc:tzid>
3092                       <xc:parameters>
3093                         <xc:date-time>20060104T140000</xc:date-time>
3094                       </xc:dtstart>
3095                       <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
3096                     <xc:summary>
3097                       <xc:text>Event #2 bis</xc:text>
3098                     </xc:summary>
3099                     <xc:uid>
3100                       <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
3101                     </xc:uid>
3102                     <xc:recurrence-id>
3103                       <xc:parameters>
3104                         <xc:tzid>US/Eastern<xc:tzid>
3105                         <xc:parameters>
3106                           <xc:date-time>20060104T120000</xc:date-time>

```

```

3107      </xc:recurrence-id>
3108      <xc:rrule>
3109      <xc:recur>
3110      <xc:freq>DAILY</xc:freq>
3111      <xc:count>5</xc:count>
3112      </xc:recur>
3113      </xc:rrule>
3114      </xc:properties>
3115      </xc:vevent>
3116
3117      <xc:vevent>
3118      <xc:properties>
3119      <xc:dtstart>
3120      <xc:parameters>
3121      <xc:tzid>US/Eastern<xc:tzid>
3122      <xc:parameters>
3123      <xc:date-time>20060106T140000</xc:date-time>
3124      </xc:dtstart>
3125      <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
3126      <xc:summary>
3127      <xc:text>Event #2 bis bis</xc:text>
3128      </xc:summary>
3129      <xc:uid>
3130      <xc:text>00959BC664CA650E933C892C@example.com</xc:text>
3131      </xc:uid>
3132      <xc:recurrence-id>
3133      <xc:parameters>
3134      <xc:tzid>US/Eastern<xc:tzid>
3135      <xc:parameters>
3136      <xc:date-time>20060106T120000</xc:date-time>
3137      </xc:recurrence-id>
3138      <xc:rrule>
3139      <xc:recur>
3140      <xc:freq>DAILY</xc:freq>
3141      <xc:count>5</xc:count>
3142      </xc:recur>
3143      </xc:rrule>
3144      </xc:properties>
3145      </xc:vevent>
3146      <xc:components>
3147      </xc:vcalendar>
3148      </xc:icalendar>
3149      </C:calendar-data>
3150      </D:prop>
3151      <D:status>HTTP/1.1 200 OK</D:status>
3152      </D:propstat>
3153      </D:response>
3154      <D:response>
3155      <D:href>http://cal.example.com/bernard/work/abcd3.ics</D:href>
3156      <D:propstat>
3157      <D:prop>
3158      <D:getetag>"fffff abcd3"</D:getetag>
3159      <C:calendar-data content-type="application/xml+calendar">
3160      <xcal:icalendar
3161          xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
3162      <xc:vcalendar>
3163      <xc:properties>
3164      <xc:calscale><text>GREGORIAN</text></xc:calscale>
3165      <xc:prodid>
3166      <xc:text> //Example Inc.//Example Calendar//EN</xc:text>
3167      </xc:prodid>
3168      <xc:version><xc:text>2.0</xc:text></xc:version>
3169      </xc:properties>

```

```

3170 <xc:components>
3171   <xc:vevent>
3172     <xc:properties>
3173       <xc:dtstart>
3174         <xc:parameters>
3175           <xc:tzid>US/Eastern<xc:tzid>
3176         <xc:parameters>
3177           <xc:date-time>20060104T100000</xc:date-time>
3178         </xc:dtstart>
3179       <xc:duration><xc:duration>PT1H</xc:duration></xc:duration>
3180     <xc:summary>
3181       <xc:text>Event #3</xc:text>
3182     </xc:summary>
3183   <xc:uid>
3184     <xc:text>DC6C50A017428C5216A2F1CD@example.com</xc:text>
3185   </xc:uid>
3186   <xc:rrule>
3187     <xc:recur>
3188       <xc:freq>DAILY</xc:freq>
3189       <xc:count>5</xc:count>
3190     </xc:recur>
3191   </xc:rrule>
3192   </xc:properties>
3193 </xc:vevent>
3194 </xc:components>
3195 </xc:vcalendar>
3196 </xc:icalendar>
3197 </C:calendar-data>
3198 </D:prop>
3199 <D:status>HTTP/1.1 200 OK</D:status>
3200 </D:propstat>
3201 </D:response>
3202 </D:multistatus>

```

3203 4.9 Free-busy queries

3204 Free-busy queries are used to obtain free-busy information for a calendar-collection or principals. The
 3205 result contains information only for events to which the current principal has sufficient access.

3206 When targeted at a calendar collection the result is based only on the calendaring entities contained in
 3207 that collection. When targeted at a principal free-busy URL the result will be based on all information
 3208 which affect the principals free-busy status, for example availability.

3209 The possible targets are:

- 3210 • A calendar collection URL
- 3211 • The XRD link with relation CalWS/current-principal-freebusy
- 3212 • The XRD link with relation CalWS/principal-freebusy with a principal given in the request.

3213 The query follows the specification defined in [FreeBusy Read URL] with certain limitations. As an
 3214 authenticated user to the CalWS service scheduling read-freebusy privileges must have been granted. As
 3215 an unauthenticated user equivalent access must have been granted to unauthenticated access.

3216 Freebusy information is returned by default as xcalendar vfreebusy components, as defined by [draft-
 3217 xcav]. Such a component is not meant to conform to the requirements of VFREEBUSY components in
 3218 [RFC 5546]. The VFREEBUSY component SHOULD conform to section "4.6.4 Free/Busy Component" of
 3219 [RFC 5545]. A client SHOULD ignore the ORGANIZER field..

3220 Since a Freebusy query can only refer to a single user, a client will already know how to match the result
 3221 component to a user. A server MUST only return a single vfreebusy component.

3222 4.9.1 ACCEPT header

3223 The Accept header is used to specify the format for the returned data. In the absence of a header the
3224 data should be returned as specified in [draft-xcal], that is, as if the following had been specified

```
3225   ACCEPT: application/xml+calendar
```

3226 4.9.2 URL Query Parameters

3227 None of these parameters are required except for the conditions noted below. Appropriate defaults will be
3228 supplied by the server.

3229 4.9.2.1 start

3230 **Default:** The default value is left up to the server. It may be the current day, start of the current
3231 month, etc.

3232 **Description:** Specifies the start date for the Freebusy data. The server is free to ignore this value and
3233 return data in any time range. The client must check the data for the returned time range.

3234 **Format:** A profile of an [RFC3339] Date/Time. Fractional time is not supported. The server MUST
3235 support the expanded version e.g.

```
3236   2007-01-02T13:00:00-08:00
```

3237 It is up to the server to interpret local date/times.

3238 **Example:**

```
3239   2007-02-03T15:30:00-0800
```

```
3240   2007-12-01T10:15:00Z
```

3241 **Notes:** Specifying only a start date/time without specifying an end date/time or period should be
3242 interpreted as in [RFC 5545]. The effective period should cover the remainder of that day.

3243 Date-only values are disallowed as the server cannot determine the correct start of the day. Only
3244 UTC or date/time with offset values are permitted.

3245 4.9.2.2 end

3246 **Default:** Same as start

3247 **Description:** Specifies the end date for the Freebusy data. The server is free to ignore this value.

3248 **Format:** Same as start

3249 **Example:** Same as start

3250 4.9.2.3 period

3251 **Default:** The default value is left up to the server. The recommended value is "P42D".

3252 **Description:** Specifies the amount of Freebusy data to return. A client cannot specify both a period
3253 and an end date. Period is relative to the start parameter.

3254 **Format:** A duration as defined in section 4.3.6 of [RFC 5545]

3255 **Example:**

```
3256   P42D
```

3257 4.9.2.4 account

3258 **Default:** none

3259 **Description:** Specifies the principal when the request is targeted at the XRD-CalWS/principal-
3260 freebusy. Specification of this parameter is an error otherwise.

3261 **Format:** Server specific

3262 **Example:**

```
3263     fred  
3264     /principals/users/jim  
3265     user1@example.com
```

4.9.3 URL parameters - notes

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

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

4.9.4 HTTP Operations

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

4.9.5 Response Codes

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

- 200 OK
- 302 Found
- 400 Start parameter could not be understood / End parameter could not be understood / Period parameter could not be understood
- 401 Unauthorized
- 403 Forbidden
- 404 The data for the requested principal is not currently available, but may be available later.
- 406 The requested format in the accept header is not supported.
- 410 The data for the requested principal is no longer available
- 500 General server error

4.9.6 Examples

The following are examples of URLs used to retrieve Freebusy data for a user:

```
3290     http://www.example.com/freebusy/user1@example.com?  
3291     start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00  
3292  
3293     http://www.example.com/freebusy/user1@example.com?  
3294     start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00  
3295  
3296     http://www.example.com/freebusy/user1@example.com  
3297  
3298     http://www.example.com/freebusy?user=user%201@example.com&  
3299     start=2008-01-01T00:00:00Z&end=2008-12-31T00:00:00Z
```

3300 Some Request/Response Examples:

3301 A URL with no query parameters:

```
3302     >>> Request <<  
3303     GET /freebusy/bernard/ HTTP/1.1  
3304     Host: www.example.com
```

```

3305
3306 >> Response <<
3307 HTTP/1.1 200 OK
3308 Content-Type: application/xml+calendar; charset="utf-8"
3309 Content-Length: ****
3310
3311 <xc:icalendar xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
3312 <xc:vcalendar>
3313 <xc:properties>
3314 <xc:calscale><text>GREGORIAN</text></xc:calscale>
3315 <xc:prodid>
3316 <xc:text>//Example Inc.//Example Calendar//EN</xc:text>
3317 </xc:prodid>
3318 <xc:version><xc:text>2.0</xc:text></xc:version>
3319 </xc:properties>
3320 <xc:components>
3321 <xc:vfreebusy>
3322 <xc:properties>
3323 <xc:uid>
3324 <xc:text>7cef34-54a3d2@example.com</xc:text>
3325 </xc:uid>
3326 <xc:dtstart>
3327 <xc:dateTime>20060101T000000Z</xc:dateTime>
3328 </xc:dtstart>
3329 <xc:dtend>
3330 <xc:dateTime>20060108T000000Z</xc:dateTime>
3331 </xc:dtend>
3332 <xc:dtstamp>
3333 <xc:dateTime>20050530T123421Z</xc:dateTime>
3334 </xc:dtstamp>
3335 <xc:freebusy>
3336 <xc:parameters>
3337 <xc:fbtype>BUSYTENTATIVE</xc:fbtype>
3338 <xc:parameters>
3339 <xc:period>20060102T100000Z/20060102T120000Z</xc:period>
3340 </xc:freebusy>
3341 <xc:freebusy>
3342 <xc:period>20060103T100000Z/20060103T120000Z</xc:period>
3343 </xc:freebusy>
3344 <xc:freebusy>
3345 <xc:period>20060104T100000Z/20060104T120000Z</xc:period>
3346 </xc:freebusy>
3347 <xc:freebusy>
3348 <xc:parameters>
3349 <xc:fbtype>BUSYUNAVAILABLE</xc:fbtype>
3350 <xc:parameters>
3351 <xc:period>20060105T100000Z/20060105T120000Z</xc:period>
3352 </xc:freebusy>
3353 <xc:freebusy>
3354 <xc:period>20060106T100000Z/20060106T120000Z</xc:period>
3355 </xc:freebusy>
3356 </xc:vfreebusy>
3357 </xc:components>
3358 </xc:vcalendar>
3359 <xc:icalendar>

```

A URL with start and end parameters:

```

3360
3361 >> Request <<
3362 GET /freebusy/user1@example.com?start=2007-09-01T00:00:00-08:00&end=2007-09-
3363 31T00:00:00-08:00
3364 HTTP/1.1
3365 Host: www.example.com
3366

```

```

3367 >> Response <<
3368 HTTP/1.1 200 OK
3369 Content-Type: application/xml+calendar; charset="utf-8"
3370 Content-Length: ****
3371
3372 <xc:icalendar xmlns:xc="urn:ietf:params:xml:ns:icalendar-2.0">
3373   <xc:vcalendar>
3374     <xc:properties>
3375       <xc:calscale><text>GREGORIAN</text></xc:calscale>
3376       <xc:prodid>
3377         <xc:text>//Example Inc.//Example Calendar//EN</xc:text>
3378       </xc:prodid>
3379       <xc:version><xc:text>2.0</xc:text></xc:version>
3380     </xc:properties>
3381     <xc:components>
3382       <xc:vfreebusy>
3383         <xc:properties>
3384           <xc:uid>
3385             <xc:text>76cf34-54a3d2@example.com</xc:text>
3386           </xc:uid>
3387           <xc:dtstart>
3388             <xc:date-time>20070901T000000Z</xc:date-time>
3389           </xc:dtstart>
3390           <xc:dtend>
3391             <xc:date-time>20070931T000000Z</xc:date-time>
3392           </xc:dtend>
3393           <xc:dtstamp>
3394             <xc:date-time>20050530T123421Z</xc:date-time>
3395           </xc:dtstamp>
3396           <xc:freebusy>
3397             <xc:period>20070915T230000Z/20070916T010000Z</xc:period>
3398           </xc:freebusy>
3399         </xc:vfreebusy>
3400       </xc:components>
3401     </xc:vcalendar>
3402   <xc:icalendar>

```

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

```

3403
3404 >> Request <<
3405 GET /freebusy/user1@example.com?start=2012-12-01T00:00:00-08:00&end=2012-12-
3406 31T00:00:00-08:00
3407 HTTP/1.1
3408 Host: www.example.com
3409
3410 >> Response <<
3411 HTTP/1.1 404 No data
3412

```

3413 **54 Conformance and Rules for WS-Calendar and** 3414 **Referencing Specifications**

3415 **5.14.1 Introduction**

3416 This section specifies conformance related to the ~~semantic model and RESTful Services. While the~~
3417 ~~semantic model applies to all WS-Calendar implementations; the other conformance statements are~~
3418 ~~relevant only to those using those services.~~

3419 ~~If information modell~~ If the implementer is merely using WS-Calendar as part of a larger business or service
3420 communication, they SHALL follow not only the semantic rules herein, but SHALL also conform to the
3421 rules for specifying inheritance in referencing standards.

3422 **5.24.2 Semantic Conformance Rules for WS-Calendar**

3423 There are five kinds of conformance that must be addressed for WS-Calendar and specifications that
3424 reference WS-Calendar.

- 3425 • Conformance to the ***inheritance rules*** in WS-Calendar, including the direction of inheritance
- 3426 • ***Specific attributes*** for each type that MUST or MUST NOT be inherited.
- 3427 • ***Conformance rules*** that Referencing Specifications MUST follow
- 3428 • Description of ***Covarying attributes*** with respect to the Reference Specification
- 3429 • ***Semantic Conformance*** for the information within the artifacts exchanged.

3430 We address each of these in the following sections.

3431 **5.2.14.2.1 Inheritance in WS-Calendar**

3432 In this section we define rules that define inheritance including direction.

3433 **I1: Proximity Rule** Within a given lineage, inheritance is evaluated through each Parent to the Child
3434 before what the Child bequeaths is evaluated.

3435 **I2: Direction Rule** Intervals MAY inherit attributes from the nearest gluon subject to the Proximity Rule
3436 and Override Rule, provided those attributes are defined as Inheritable.

3437 **I3: Override Rule** If and only if there is no value for a given attribute of a Gluon or Interval, that Gluon or
3438 Interval SHALL inherit the value for that attribute from its nearest Ancestor in conformance to the
3439 Proximity Rule.

3440 **I4: Comparison Rule** Two Sequences are equivalent if a comparison of the respective Intervals
3441 succeeds as if each Sequence were fully Bound and redundant Gluons are removed.

3442 | **I5: Designated Interval Inheritance** [To facilitate composition of Sequences] the Designated Interval in
3443 | the ultimate Ancestor of a Gluon is the Designated Interval of the composed Sequence.⁸ Special
3444 | conformance rules for Designated Intervals apply only to the Interval linked from the Designator Gluon.

3445 | **I6: Start Time Inheritance** When a start time is specified through inheritance, that start time is inherited
3446 | only by the Designated Interval; the start time of all other Intervals are computed through the durations
3447 | and temporal relationships within the Sequence. The designatedDesignated Interval is the Interval whose
3448 | parent is at the end of the lineage.

3449 | **5.2.24.2.2 Specific Attribute Inheritance in WS-Calendar**

3450 | In WS-Calendar the following attributes MUST be inherited in conformance to the Rules (same for Gluons
3451 | and Intervals):

- 3452 | • dtStart
- 3453 | • dtEnd
- 3454 | • duration
- 3455 | • designatedIntervalDuration
- 3456 | • Designated Interval (Gluon, special upward inheritance rule)
- 3457 | • performance
- 3458 | • performanceInterval
- 3459 | • Tolerance

3460 | In WS-Calendar the following attributes MUST NOT be inherited

- 3461 | • UID (Gluons and Intervals)
- 3462 | • Temporal Relationships (between Intervals)

3463 | **4.2.3 General Conformance Issues**

3464 | This specification is general purpose. Standards that claim conformance to this specification may need to
3465 | restrict the variability inherent in the expressions of Date and Time to improve interoperation within their
3466 | own interactions. Aspects of Date and Time that may reward attention and conformance statements
3467 | include:

- 3468 | • Precision – Does the conforming specification express time in Hours or in milliseconds. Consider
3469 | a standard format recommendation.
- 3470 | • Time Zones and UTC – Business interactions have a “natural” choice of local, time zone, or UTC
3471 | based expression of time. Intents may be local, as they tie to the business processes that drive
3472 | them. Tenders may be Time-zone based, as they are driven by the local business process, but
3473 | may require future action across changes in time and in time zone. Transaction recording may
3474 | demand UTC, for complete unambiguity. The specification cannot require one or another, but
3475 | particular business processes may require appropriate conformance statements.

⁸ We are assuming here that Sequences can be composed to form new Sequences. This needs detailed discussion as the rules for Designated Intervals cannot easily be applied to a Sequence of Sequences.

- 3476 • **Business Purpose** – Because WS-Calendar is general purpose, it does not distinguish between
 3477 different exchanges that may have different purposes. For example, a general indication of
 3478 capability and/or timeliness may be appropriate for a market tender, and an unanchored
 3479 Sequence may be appropriate. In the same specification, performance execution could require
 3480 merely the Gluon to Anchor the Interval. If the distinction between Unanchored and Anchored
 3481 Interval is critical for a set of interactions, the referencing specification SHALL indicate the proper
 3482 form for a given exchange.

4.2.4 Covarying Elements

3484 Some elements of WS-Calendar objects may be **covarying**, meaning that they change together. Such
 3485 elements are treated as a single element for inheritance, they are either inherited together or the child
 3486 keeps its current values intact. This becomes important if one or more of a covarying set have default
 3487 values. In that case, if any are present, then inheritance should deem they are all present, albeit some
 3488 perhaps in their default values.

5.2.34.2.5 Conformance of Intervals in WS-Calendar

5.2.3.14.2.5.1 Intervals

3491 WS-Calendar Intervals SHALL have a Duration.

3492 Intervals MAY have a StartTime.Start Time.

3493 Intervals SHALL NOT include an END time.have a Duration AND a dtStart OR a dtEnd. If a non-compliant
 3494 Interval is received with an END time, it may both a dtStart and a dtEnd, then the dtEnd SHALL be
 3495 ignored.

3496 Within a Sequence, a maximum of a single Interval MAY have a dtStart or a dtEnd.

5.2.3.24.2.5.2 Other Elements

3498 A performanceTolerance Property component SHALL notNOT include Start, Stop, and Duration
 3499 elements. Two out of the three elements is acceptable, but not three.

3500 In Partitions, the Description, Summary and Priority of each Interval SHALL be excluded.

3501 A Calendar Gluon may have either a dtStart or a dtEnd, but may not have both.

5.2.44.2.6 Conformance of Bound Intervals and Sequences in WS-Calendar

3503 Actionable services require Bound Intervals as part of a Bound Sequence. Services may include Intervals
 3504 that are not bound for informational or negotiation purposes. Some of these are modeled and
 3505 described as constraints in the UML models that have been produced separately.

- 3506 • Intervals SHALL have values assigned for dtStart and duration, either explicitly or through
 3507 inheritance

- Intervals SHALL have no value assigned for dtEnd⁹
- Within a Sequence at most the Designated Interval may have dtStart and duration with a value specified or inherited.¹⁰
- If Sequences are composed to create other Sequences, then the Designated Intervals within the composing Sequence are ignored.
- Any specification claiming conformance to WS-Calendar MUST satisfy all of the following conditions:
 - Follow the same style of inheritance (per the Rules)
 - Specify attribute inheritability in the specification claiming conformance
 - Specify whether certain sets of elements must be inherited as a group or specify that all elements can be inherited or not on an individual basis

5.3 Conformance Rules for RESTful Services

~~Still to come~~

5.44.3 Conformance Rules for Specifications Claiming Conformance to WS-Calendar

Specifications that claim conformance to WS-Calendar SHALL specify inheritance rules for use within their specification. These rules SHALL NOT violate override the Proximity, Direction, or Override Rules. If the specification includes covariant elements, those elements ~~SHAL~~SHALL be clearly designated in the specification.

~~Specifications that normatively reference and claim conformance with WS-Calendar SHALL define the business meaning of zero duration Intervals.~~

~~Specifications that normatively reference and claim conformance with WS-Calendar SHALL define the business meaning of zero duration Intervals.~~

4.4 Security Considerations

~~Part 1 of WS-Calendar describes an informational model. Specifications claiming conformance with WS-Calendar may use the schedule and interval communication as but a small part of their overall communications.~~

⁹~~While VTOD objects allow for all three of dtStart, dtEnd, and duration, the scheduling use for automation is simpler if only dtStart and duration are used.~~

¹⁰~~Note that composition of Sequences to create other Sequences raises issues both of inheritance direction and the meaning of subSequences. We suggest an approach of ignoring Designated Intervals with respect to the composed Sequence as simpler than having the new subSequences change form and not be reusable.~~

3536 Communications that claim conformance to this specification should select the communication and the
3537 well-known methods to secure that communication appropriate to the information exchanged and paying
3538 heed to the costs of both communication failure and of inappropriate disclosure. To the extent that normal
3539 schedule servers are used, the capabilities of security of those systems should be considered as well.
3540 Those concerns are out of scope for this specification.
3541 Specifications which do not use the REST or SOAP interactions face similar concerns in designing the
3542 authentication, authorization, interactions, and storage of the information artifacts produced. Such
3543 concerns are out of scope within this general model.

3544 Acknowledgements

3545 The following individuals have participated in the creation of this specification and are gratefully
3546 acknowledged:

3547 Participants:

3548 Bruce Bartell, Southern California Edison
3549 Brad Benson, Trane
3550 Edward Cazalet, Individual
3551 Toby Considine, University of North Carolina at Chapel Hill
3552 William Cox, Individual
3553 Sharon Dinges, Trane
3554 | Mike [Douglas, Douglass](#), Rensselaer Polytechnic Institute
3555 Craig Gemmill, Tridium, Inc.
3556 Girish Ghatikar, Lawrence Berkeley National Laboratory
3557 Gerald Gray, Southern California Edison
3558 | [David Hardin, ENERNOC](#)
3559 Gale Horst, Electric Power Research Institute (EPRI)
3560 Gershon Janssen, Individual
3561 Ed Koch, Akuacom Inc.
3562 Benoit Lepeuple, LonMark International*
3563 Carl Mattocks, CheckMi*
3564 Robert Old, Siemens AG
3565 Alexander Papaspouropoulos, Technische Universitat Dortmund
3566 | [Joshua Phillips, ISO/RTO Council \(IRC\)](#)
3567 Jeremy J. Roberts, LonMark International
3568 | David Thewlis, CalConnect
3569 |
3570

3571 The Calendaring and Scheduling Consortium (CalConnect) TC-XML committee worked closely with WS-
3572 Calendar Technical Committee, bridging to developing IETF standards and contributing the services
3573 definitions that make up Services in Section 4. The Technical Committee gratefully acknowledges their
3574 assistance and cooperation as well. Contributors to TC XML include:

3575 Cyrus Daboo, Apple
3576 | Mike [Douglas, Douglass](#), Rensselaer Polytechnic Institute
3577 Steven Lees, Microsoft
3578 Tong Li, IBM
3579
3580

3581 An Introduction to Internet Calendaring

3582 The WS-Calendar Technical Committee thanks CalConnect for contributing this overview of iCalendar
3583 and its use.

3584 **iCalendar**

3585 **History**

3586 The iCalendar specification was first produced by the IETF in 1998 as RFC 2445 [1]. Since then it has
3587 become the dominant standard for calendar data interchange on the internet and between devices
3588 (desktop computers, mobile phones etc.). The specification was revised in 2009 as RFC 5545 [4].
3589 Alongside iCalendar is the iTIP specification (RFC 2446 [2] and revised as RFC 5546[5]) that defines how
3590 iCalendar is used to carry out scheduling operations (for example, how an organizer can invite attendees
3591 to a meeting and receive their replies). This forms the basis for email-based scheduling using iMIP (the
3592 specification that describes how to use iTIP with email - RFC [24476047](#) [3]).
3593 iCalendar itself is a text-based data format. However, an XML format is also available, providing a one-to-
3594 one mapping to the text format (draft [7]).
3595 iCalendar data files typically have a .ics file name extension. Most desktop calendar clients can import or
3596 export iCalendar data, or directly access such data over the Internet using a variety of protocols.

3597 **Data model**

3598 The iCalendar data format has a well defined data model. "iCalendar objects" encompass a set of
3599 "iCalendar **components**Components" each of which contains a set of "iCalendar properties" and possibly
3600 other sub-**components**.Components. An iCalendar property consists of a name, a set of optional
3601 parameters (specified as "key-value" pairs) and a value.
3602 iCalendar **components**Components include:
3603 "VEVENT" which represents an event
3604 "VTODO" which represents a task or to-do
3605 "VJOURNAL" which represents a journal entry
3606 "VFREEBUSY" which represents periods of free or busy time information
3607 "VTIMEZONE" which represents a timezone definition (timezone offset and daylight saving rules)
3608 "VALARM" is currently the only defined sub-**component**Component and is used to set alarms or
3609 reminders on events or tasks.
3610 Properties include:
3611 "DTSTART" which represents a start time for a **component**Component
3612 "DTEND" which represents an end time for a **component**Component
3613 "SUMMARY" which represents a title or summary for a **component**Component
3614 "RRULE" which can specify rules for repeating events or tasks (for example, every day, every week on
3615 Tuesdays, etc.)
3616 "ORGANIZER" which represents the calendar user who is organizing an event or assigning a task
3617 "ATTENDEE" which represents calendar users attending an event or assigned a task
3618 In addition to this data model and the pre-defined properties, the specification defines how all those are
3619 used together to define the semantics of calendar objects and scheduling. The semantics are basically a
3620 set of rules stating how all the **components**Components and properties are used together to ensure that
3621 all iCalendar products can work together to achieve good interoperability. For example, a rule requires

3622 that all events must have one and only one "DTSTART" property. The most important part of the
3623 iCalendar specification is the semantics of the calendaring model that it represents. The use of text or
3624 XML to encode those is secondary.

3625 **Scheduling**

3626 The iTIP specification defines how iCalendar objects are exchanged in order to accomplish the key task
3627 needed to schedule events or tasks. An example of a simple workflow is as follows:

- 3628 1. To schedule an event, an organizer creates the iCalendar object representing the event and adds
3629 calendar users as attendees.
- 3630 2. The organizer then sends an iTIP "REQUEST" message to all the attendees.
- 3631 3. Upon receipt of the scheduling message, each attendee can decide whether they want to attend
3632 the meeting or not.
- 3633 4. Each attendee can then respond back to the organizer using an iTIP "REPLY" message
3634 indicating their own attendance status.

3635 iTIP supports other types of scheduling messages, for example, to cancel meetings, add new instances to
3636 a repeating meeting, etc.

3637 **Extensibility**

3638 | iCalendar was designed to be extensible, allowing for new [components](#)[Components](#), properties and
3639 parameters to be defined as needed. A registry exists to maintain the list of standard extensions with
3640 references to their definitions to ensure anyone can use them and work well with others.

3641 **Calendar data access and exchange protocols**

3642 **Internet Calendar Subscriptions**

3643 An Internet calendar subscription is simply an iCalendar data file made available on a web server. Users
3644 can use this data in two ways:

3645 The data can be downloaded from the web server and then imported directly into an iCalendar aware
3646 client. This solution works well for calendar data that is not likely to change over time (for example
3647 the list of national holidays for the next year).

3648 Calendar clients that support "direct" subscriptions can use the URL to the calendar data on the web
3649 server to download the calendar data themselves. Additionally, the clients can check the web
3650 server on a regular basis for updates to the calendar data, and then update their own cached
3651 copy of it. This allows calendar data that changes over time to be kept synchronized.

3652 **CalDAV**

3653 CalDAV is a calendar access protocol and is defined in RFC 4791 [6]. The protocol is based on WebDAV
3654 which is an extension to HTTP that provides enhanced capabilities for document management on web
3655 servers.

3656 CalDAV is used in a variety of different environments, ranging from very large internet service providers,
3657 to large and small corporations or institutions, and to small businesses and individuals.

3658 CalDAV clients include desktop applications, mobile devices and browser-based solutions. It can also be
3659 used by "applets", for example, a web page panel that displays a user's upcoming events.

3660 One of the key aspects of CalDAV is its data model. Simply put, it defines a "calendar home" for each
3661 calendar user, within which any number of "calendars" can be created. Each "calendar" can contain any
3662 number of iCalendar objects representing individual events, tasks or journal entries. This data model
3663 ensures that clients and servers can interoperate well.

3664 In addition to providing simple operations to read, write and delete calendar data, CalDAV provides a
3665 querying mechanism to allow clients to fetch calendar data matching specific criteria. This is commonly
3666 used by clients to do "time-range" queries, i.e., find the set of events that occur within a given start/end
3667 time period.
3668 CalDAV also supports access control allowing for features such as delegated calendars and calendar
3669 sharing.
3670 CalDAV also specifies how scheduling operations can be done using the protocol. Whilst it uses the
3671 semantics of the iTIP protocol, it simplifies the process by allowing simple calendar data write operations
3672 to trigger the sending of scheduling messages, and it has the server automatically process the receipt of
3673 scheduling messages. Scheduling can be done with other users on the CalDAV server or with calendar
3674 users on other systems (via some form of "gateway").

3675 **ActiveSync/SyncML**

3676 ActiveSync and SyncML are technologies that allow multiple devices to synchronize data with a server,
3677 with calendar data being one of the classes of data supported. These have typically been used for low-
3678 end and high-end mobile devices.

3679 **CalWS**

3680 CalWS is a web services calendar access API developed by The Calendaring and Scheduling
3681 Consortium and the OASIS organization, to be used as part of the Oasis WS-Calendar standard. It
3682 provides an API to access and manipulate calendar data stored on a server. It follows a similar data
3683 model to CalDAV and has been designed to co-exist with a CalDAV service offering the same data.

3684 **iSchedule**

3685 iSchedule is a protocol to allow scheduling between users on different calendaring systems and across
3686 different internet domains. It transports iTIP scheduling messages using HTTP between servers. Servers
3687 use DNS and various security mechanisms to determine the authenticity of messages received.
3688 It has been specifically designed to be independent of any calendar system in use at the endpoints, so
3689 that it is compatible with many different systems. This allows organizations with different calendar
3690 systems to exchange scheduling messages with each other, and also allows a single organization with
3691 multiple calendar systems (for example due to mergers, or different departmental requirements) to
3692 exchange scheduling messages between users of each system.

3693 **References**

- 3694 [1] <https://datatracker.ietf.org/doc/rfc2445/> : 'Internet Calendaring and Scheduling Core Object
3695 Specification'
- 3696 [2] <https://datatracker.ietf.org/doc/rfc2446/> : 'iCalendar Transport-Independent Interoperability Protocol'
- 3697 [3] <https://datatracker.ietf.org/doc/rfc2447rfc6047/> : 'iCalendar Message-Based Interoperability Protocol'
- 3698 [4] <https://datatracker.ietf.org/doc/rfc5545/> : 'Internet Calendaring and Scheduling Core Object
3699 Specification'
- 3700 [5] <https://datatracker.ietf.org/doc/rfc5546/> : 'iCalendar Transport-Independent Interoperability Protocol'
- 3701 [6] <https://datatracker.ietf.org/doc/rfc4791/> : 'Calendaring Extensions to WebDAV'
- 3702 [7] <https://datatracker.ietf.org/doc/draft-daboo-et-al-icalendar-in-xml/> : 'xCal: The XML format for
3703 iCalendar'

3705

Overview of WS-Calendar, its Antecedents and its Use

3706 iCalendar has long been the predominant message format for an Internet user to send meeting requests
 3707 and tasks to other Internet users by email. The recipient can respond to the sender easily or counter
 3708 propose another meeting date/time. iCalendar support is built into all major email systems and email
 3709 clients. While SMTP is the predominant means to transport iCalendar messages, protocols including
 3710 WebDAV and SyncML are used to transport collections of iCalendar information. No similar standard for
 3711 service interactions has achieved similar widespread use.

3712 The Calendar and Scheduling Consortium (CalConnect), working within the IETF, updated the iCalendar
 3713 standard in the summer of 2009 to support extension ([RFC5545]). In 2010, the same group defined
 3714 [XCAL], a canonical XML serialization for iCalendar, currently (08/21/2008) on the recommended
 3715 standards track within the IETF. This specification supports extensions, including handling non-standard,
 3716 i.e., non-iCalendar, data during message storage and retrieval.

3717 WS-Calendar builds on this work, and consists of extensions to the vocabulary of iCalendar, along with
 3718 standard services to extend calendaring and scheduling into service interactions. iCalendar consists of a
 3719 number of fields that support the delivery, update, and synchronization of calendar messages and a list
 3720 of [components](#). [Components](#) The [components](#) can specify defined relationships between
 3721 each other.

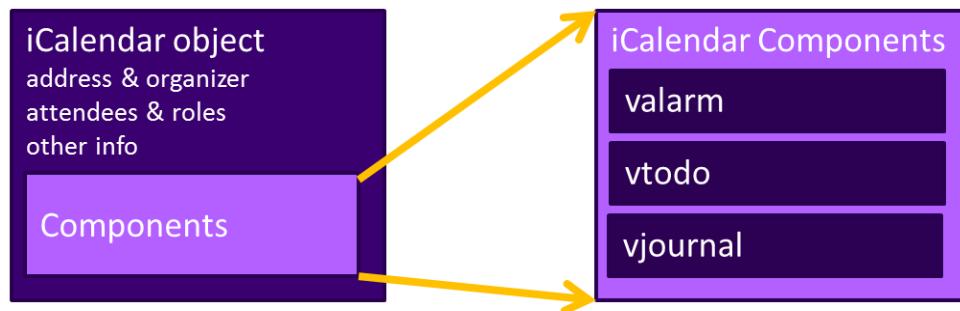
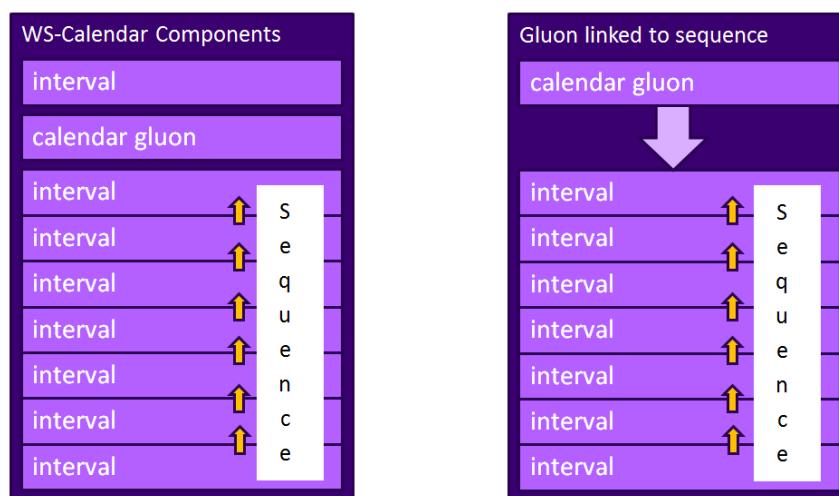
3722
3723

Figure 1: iCalendar overview

3724 WS-Calendar defines the Interval, a profile of the [vtodo-component](#) **VTODO Component** requiring only a
 3725 duration and an artifact to define service delivery and performance. WS-Calendar also defines the
 3726 CalendarGluon [component](#) **Component**, a container for holding only a service delivery and performance
 3727 artifact, to associate with a [component](#) **Component** or group of [components](#).



3728

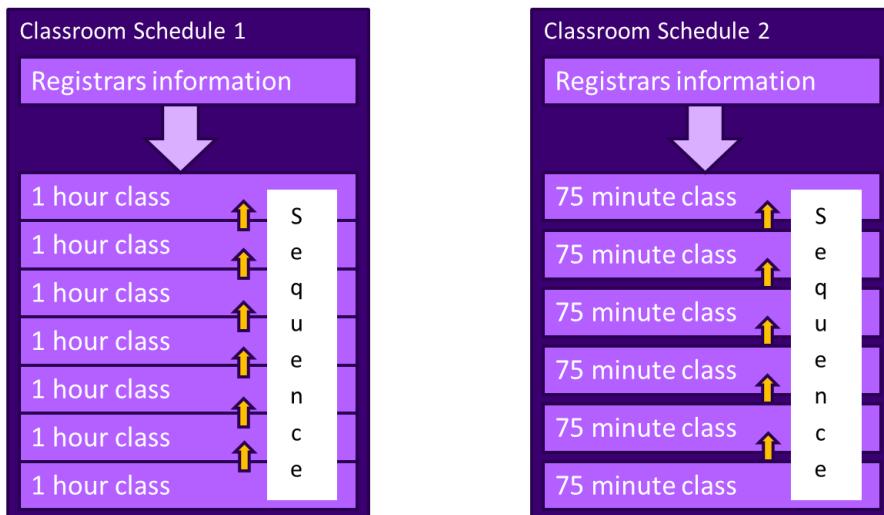
3729 *Figure 2: WS-Calendar and EMIX*

3730 A set of Intervals that have defined temporal relationships is a Sequence. Temporal relationships express
3731 how the occurrence of one Interval is related to another. For example, Interval B may begin 10 minutes
3732 after Interval A completes, or Interval D may start 5 minutes after Interval C starts. [An CalendarA](#) Gluon
3733 linked to a Sequence defines service performance for all Intervals in the Sequence. Because each
3734 Interval has its own service performance contract, specifications built on WS-Calendar can define rules
3735 for inheritance and over-rides with a Sequence.
3736 The Partition is a sub-class of a Sequence in which all Intervals follow consecutively with no lag time.
3737 Intervals in a Partition normally have the same Duration, but WS-Calendar does support overriding the
3738 duration on an individual basis.

3739 **Scheduling Sequences**

3740 A Sequence is a general pattern of behaviors and results that does not require a specific schedule. A
3741 publishing service may advertise a Sequence with no schedule, i.e., no specific time for performance.
3742 When the Sequence is invoked or contracted, a specific performance time is added. In the original
3743 iCalendar [componentsComponents](#), this would add the starting date and time (dtStart) to the
3744 [componentComponent](#). In WS-Calendar, we add the starting date and time only to the first Interval of a
3745 Sequence; the performance times for all other Intervals in the Sequence are derived from that one start
3746 time.

3747 **Academic Scheduling example**



3748

3749 *Figure 3: Classroom Scheduling Example*

3750 A college campus uses two schedules to schedule its buildings. In Schedule 1, classes start on the hour,
3751 and follow one after another; each class starts on the hour. In the second schedule, each class lasts an
3752 hour and a quarter, and there is a fifteen minute gap between classes; classes start on the half hour. On
3753 many campuses, the Sequence in Schedule 1 may describe classes taught on Monday, Wednesday, and
3754 Friday. Schedule 2 may describe classes taught on Tuesday and Thursday.

3755 The registrar's office knows some key facts about each classroom, including whether it hosts a class
3756 during a particular period, and the number of students that will be in that class. The college wishes to
3757 optimize the provision of building services for each class. Such services may include adequate ventilation
3758 and comfortable temperatures to assure alert students. Other services may ensure that the classroom
3759 projection systems and A/V support services are warmed up in advance of a class, or powered off when a
3760 classroom is vacant.

3761 Although most classes meet over typical schedule for the week (M-W-F or Tu-Th), some classes may not
3762 meet on Friday, or may have a tutorial section one day a week. The registrar's system, ever mindful of

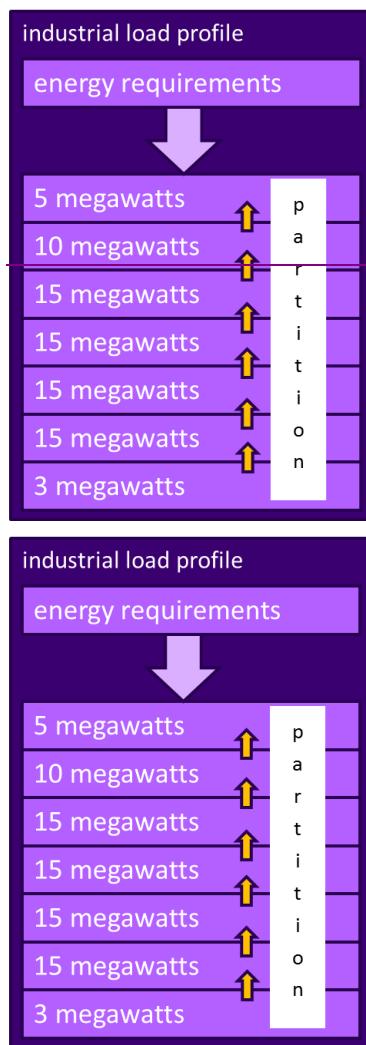
3763 student privacy, shares only minimal information with the building systems such as how many students
3764 will be supported.

3765 | The Registrar's system schedule building systems using the [Calendar](#) Gluon (registrar's information) and
3766 the student counts for each Interval, and schedules the Sequence in classroom schedule 1 three days a
3767 week for the next 10 weeks. The Registrar's system also schedules the Sequence in classroom schedule
3768 2 two days a week, also for 10 weeks.

3769 This example demonstrates a system (A) that offers services using either of two Sequences. Another
3770 business system (B) with minimal knowledge of how (A) works determines the performance requirements
3771 for (A). The business system (B) communicates what these expectations are by scheduling the
3772 Sequences offered by (A).

3773 Market Performance schedule

3774 A factory relies on an energy-intensive process which is performs twice a year for eight weeks. The
3775 factory has some flexibility about scheduling the process; it can perform the work in either the early
3776 morning or the early evening; it avoids the afternoon when energy costs are highest. The factory works up
3777 a detailed profile of when it will need energy to support this process.



3780 *Figure 4: Daily Load Profile for Market Operations Example*

3781 Factory management has decided that they want to use only renewable energy products for this process.
3782 They approach two regional wind farms with the intent of making committed purchases of wind energy.

3783 The wind farms consider their proposals taking into account the seasonal weather forecasts they use to
3784 project their weather capacity, and considering the costs that may be required to buy additional wind
3785 energy on the spot market to make up any shortfalls.

3786 Each energy supplier submits of the same Sequence, a schedule, i.e. a daily starting time, and a price for
3787 the season's prodUTCion:production. After considering the bids, and other internal costs of each
3788 proposal, the factory opts to accept a contract for the purchase of a fixed load profile (Partition), using the
3789 evening wind generation from one of the suppliers. This contract specifies Schedules of load purchases
3790 (starting data and time for the Sequence) for each day.

Revision History

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
1.0 WD 12	2010-09-14	Toby Considine Dave Thewlis	Edits for clarity and flow following changes in WD11, updated examples based upon XSD artifacts. Adding final contribution from CalConnect for Services.

1.0 WD 13		Toby Considine	Mechanistic processing of trivial comments for grammar, spelling, etc.
1.0 WD 14	2011-01-17	Toby Considine	Added Conformance rules, redefined inheritance, added terminology section in Section 1, added language on separability of information model, REST, and SOAP sections
1.0 WD 15	2011-01-27	Toby Considine	Pulled more definitions into Terminology Section, re-factored into multiple tables, Added Availability. Have not updated examples.
1.0 WD 15	2011-01-29	Toby Considine	Re-added footers to document (?!?) Added disclaimers on completeness prior to committee spec draft.
1.0 WD16	2011-02-07	Toby Considine	Minor changes to prepare for CSD as directed by TC
1.0 WD17	2011-03-01	Toby Considine	Reworked all examples, responded to numerous Jira editorial comments, eliminated "Mixed Inheritance of Schedule", introduced Vavailability, eliminated UML chapter which confused more than enlightened.
1.0 WD18	2011-03-16	Toby Considine William Cox	Tightened language, spelling and grammar, consolidated chapters into "larger sections" Corrected to use CHILD link instead of PARENT in conformance with RFC5545. Replaced LINK language that was leftover from earlier schemas.
1.0 WD19	2011-03-19	Toby Considine	Changes to namespace to prepare for CSD, PR02, as directed by TC vote on 3/18/2011

3792
3793

1.0 WD20	2011-05-12	Toby Considine	Mechanical edits. Rebuilt document to remove cross-reference corruption (table and example lists), applied grammatical and punctuation changes from PR02, simple global replaces of terms. Reference checks. Refinement of logic of Duration/DtStart. Eliminated redefinition of VAVAILABILITY.
1.0 WD21	2011-05-16	Toby Considine	More Jira edits, especially unscrambling dtStart, dtEnd, and Duration, Vavailability, and many reference checks.
1.0 WD22	2011-05-20	William Cox	Eliminated Parts two and Three
1.0 WD23	2011-05-23	Toby Considine	Minor edits for clarity, final Jira issues
1.0 WD24	2011-05-26	Toby Considine	Examples updated
1.0 WD25	2011-05-26	William Cox	Eliminated remaining references to Parts Two

			and Three, corrected internal links
--	--	--	-------------------------------------

3794