



# Schedule Signals and Streams Version 1.0

## Committee Specification 01

18 September 2016

### Specification URIs

#### This version:

<http://docs.oasis-open.org/ws-calendar/streams/v1.0/cs01/streams-v1.0-cs01.pdf> (Authoritative)  
<http://docs.oasis-open.org/ws-calendar/streams/v1.0/cs01/streams-v1.0-cs01.html>  
<http://docs.oasis-open.org/ws-calendar/streams/v1.0/cs01/streams-v1.0-cs01.doc>

#### Previous version:

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

#### Latest version:

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

### Technical Committee:

OASIS Web Services Calendar (WS-Calendar) TC

#### Chair:

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

#### Editors:

Toby Considine ([toby.considine@unc.edu](mailto:toby.considine@unc.edu)), University of North Carolina at Chapel Hill  
William T. Cox ([wtcoc@CoxSoftwareArchitects.com](mailto:wtcoc@CoxSoftwareArchitects.com)), Individual

### Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- XML schema: <http://docs.oasis-open.org/ws-calendar/streams/v1.0/cs01/xsd/ws-calendar-streams-v1.0.xsd>

### Related work:

This specification is related to:

- WS-Calendar Platform Independent Model (PIM) Version 1.0. Edited by W.T. Cox and Toby Considine. Latest version: <http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.pdf>
- WS-Calendar Version 1.0. Edited by Toby Considine and Mike Douglass. Latest version: <http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.pdf>
- *WS-Calendar Minimal PIM-Conformant Schema Version 1.0*. Edited by Toby Considine and William T. Cox. Latest version: <http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.html>.

### Declared XML namespace:

- <http://docs.oasis-open.org/ws-calendar/ns/streams/201606>

### Abstract:

There is a common need to communicate information linked to repetitive intervals of time, for history, for telemetry, for projections, for bids. Much of the information in each interval can be

inferred from the surrounding intervals. The document defines a normative structure for conveying time series of information that is conformant with the WS-Calendar Platform Independent Model (PIM). Specifications that conform to the WS-Calendar PIM can be transformed into each other and into the WS-Calendar 1.0 model. We term these conveyances “Streams”.

**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” location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ws-calendar#technical](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-calendar#technical).

TC members should send comments on this specification to the TC’s email list. Others should send comments to the TC’s public comment list, after subscribing to it by following the instructions at the “[Send A Comment](#)” button on the TC’s web page at <https://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 TC’s web page (<https://www.oasis-open.org/committees/ws-calendar/ipr.php>).

**Citation format:**

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

**[streams-v1.0]**

*Schedule Signals and Streams Version 1.0*. Edited by Toby Considine and William T. Cox. 18 September 2016. OASIS Committee Specification 01. <http://docs.oasis-open.org/ws-calendar/streams/v1.0/cs01/streams-v1.0-cs01.html>. Latest version: <http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html>.

---

## Notices

Copyright © OASIS Open 2016. 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 name "OASIS" is a trademark 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 <https://www.oasis-open.org/policies-guidelines/trademark> for above guidance.

---

# Table of Contents

|             |   |    |
|-------------|---|----|
| 1           | Introduction.....   | 7  |
| 1.1         | Terminology.....  | 8  |
| 1.2         | Normative References.....                                       | 8  |
| 1.3         | Non-Normative References.....                                   | 8  |
| 1.4         | Namespace.....  | 8  |
| 1.5         | Naming Conventions.....   | 9  |
| 1.6         | Editing Conventions.....  | 9  |
| 2           | WS-Calendar in Streams.....                                     | 10 |
| 2.1         | When: Start, End and Duration.....                              | 10 |
| 2.2         | Semantics of Inheritance.....                                   | 10 |
| 2.3         | Semantics from MIN.....   | 10 |
| 3           | Streams.....  | 11 |
| 3.1         | New Semantic Elements in Streams.....                           | 11 |
| 3.2         | Intervals and Unique Identifiers.....                           | 12 |
| 3.3         | Streams: a Restricted Profile for Sequences and Intervals.....  | 13 |
| 3.4         | Observational Data expressed as Streams.....                    | 13 |
| 3.5         | Payload Optimization in Streams.....                            | 14 |
| 3.6         | Extending Stream Payloads.....                                  | 14 |
| 4           | Conformance.....  | 15 |
| 4.1         | Conformance Points.....   | 15 |
| 4.2         | Conformance of Streams to WS-Calendar-PIM.....                  | 15 |
| 4.3         | Inheritance within Streams.....                                 | 15 |
| 4.4         | Stream expression of Intervals expressed as Durations.....      | 15 |
| 4.5         | Conformance for Observational Data.....                         | 16 |
| 4.6         | Conformance for Stream Payloads and Optimizing Inheritance..... | 16 |
| Appendix A. | Acknowledgments.....  | 17 |
| Appendix B. | Revision History.....   | 18 |

---

## Table of Figures

|  |    |
|--|----|
| Figure 3-1: Stream as Gluon-Equivalent and Degenerate Sequence ..... | 12 |
| Figure 3-2: Interval, the components of a Sequence .....             | 12 |

---

## Table of Tables

|  |    |
|--|----|
| Table 1-1: Namespaces Used in this Specification .....                 | 9  |
| Table 3-1: Core Semantics and their derivations from WS-Calendar ..... | 11 |

---

# 1 Introduction

2 All text is normative unless otherwise labeled

3 There is a common need to communicate information linked to repetitive intervals of time, for history, for  
4 telemetry, for projections, and for bids. Such communications benefit from a common model for conveying  
5 these series of information.

6 The iCalendar model is almost infinitely malleable in the number and manner of intervals in time that it  
7 can communicate. Separate intervals exist as separate calendar information objects; a single  
8 communication can include any number of these objects. This model is verbose in that each of these  
9 calendar information objects **MUST** include all distinct information.

10 The **[WS-Calendar]** model adds to the underlying iCalendar model the notion of inheritance. Using  
11 inheritance, one or many of the calendar information objects can be “completed” by applying the inherited  
12 information to the information conveyed within the object. WS-Calendar specifies rules for how this  
13 inheritance is applied, and how to handle instances wherein the inherited information collides with  
14 information inside the calendar information object.

15 **[WS-Calendar]** and **[WS-Calendar PIM]** also define the Sequence, in which sets of time-related Intervals  
16 are handled as a single entity. WS-Calendar defines a special case of the Sequence, the Partition, for the  
17 special case wherein substantially all of the Intervals are of the same Duration. Sequences rely on  
18 Inheritance to convey the repetitive information in each Interval of a Sequence.

19 A key concern for **[WS-Calendar]** was direct compatibility with **[xCal]**, the XML Format for iCalendar  
20 defined in **[RFC6321]**. While this format is flexible, it can offer too much optionality to be easily analyzed.  
21 To this end, the TC developed a Platform Independent Model **[WS-Calendar PIM]**, which supports all the  
22 functions and messages from WS-Calendar, while restricting extension so that the models can be  
23 analyzed and validated. This approach redefined WS-Calendar as what Model Driven Architecture calls a  
24 Platform Specific Model (PSM) that conforms to **[WS-Calendar PIM]**

25 The Platform Independent Model **[WS-Calendar PIM]** describes how to make use of the general model  
26 and semantics defined in **[WS-Calendar]** when defining information exchanges subject to specific  
27 constraints. Artifacts that are conformant with **[WS-Calendar PIM]** can be transformed into a form that is  
28 conformant to **[WS-Calendar]**, even while their expression may not support the general purpose  
29 expression required for **[WS-Calendar]**.

30 **[WS-Calendar PIM]** is a general specification and makes no assumptions about how its information  
31 model is used. **[WS-Calendar PIM]** has specific rules that define Inheritance as a means to reduce the  
32 conveyance of repetitive information. As this specification constrains schedule communications to specific  
33 business interactions, these inheritance rules are extended to embrace rules of interaction and rules of  
34 process that further reduce the information that **MUST** be expressed in each Interval.

35 **[WS-Calendar PIM]** does not define a normative structure for the information conveyed. **[WS-Calendar  
36 PIM]** is an information model, and information models can be conveyed in a number of ways. High speed  
37 transaction processing requires more predictable means to convey structured information concerning  
38 time-based events, states, and transactions. Even legal and conformant conveyances of calendar  
39 information may fail to meet the requirements for basic interoperability requirements **[WSI-Basic]**.

40 The document defines a normative structure for conveying time series of information that is conformant  
41 with **[WS-Calendar PIM]**. We term these conveyances “Streams”.

42 Streams specifies a PSM that conforms to **[WS-Calendar PIM]**. Model driven architecture considers that  
43 any PSM conformant to a PIM can be transformed into an expression conformant with any other PSM,  
44 and thus transitively conforms to that other PSM. In this way, Streams is conformant not only with **[WS-  
45 Calendar PIM]** but with **[WS-Calendar]**.

## 46 1.1 Terminology

47 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD  
48 NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described  
49 in RFC2119.

## 50 1.2 Normative References

- 51 **ISO8601** ISO (International Organization for Standardization). *Representations of dates*  
52 *and times, third edition*, December 2004, (ISO 8601:2004)
- 53 **MIN** **WS-Calendar Minimal PIM-Conformant Schema Version 1.0. Edited by Toby**  
54 **Considine and William Cox. 18 December 2015.** OASIS Committee  
55 Specification Draft 01. December 2015, [http://docs.oasis-open.org/ws-](http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.pdf)  
56 [calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.pdf](http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.pdf)
- 57 **RFC2119** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,  
58 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 59 **RFC5545** B. Desruisseaux *Internet Calendaring and Scheduling Core Object Specification*  
60 *(iCalendar)*, <http://www.ietf.org/rfc/rfc5545.txt>, IETF RFC5545, proposed  
61 standard, September 2009
- 62 **WS-Calendar PIM** “*WS-Calendar Platform Independent Model (PIM) Version 1.0*”. Edited by  
63 William T. Cox and Toby Considine. 21 August, 2015. OASIS Committee  
64 Specification 02. [http://docs.oasis-open.org/ws-calendar/ws-calendar-](http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.pdf)  
65 [pim/v1.0/ws-calendar-pim-v1.0.pdf](http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.pdf)
- 66 **XML NAMES** **T Bray, D Hollander, A Layman, R Tobin, HS Thompson “Namespaces in**  
67 **XML 1.0 (Third Edition)”** <http://www.w3.org/TR/xml-names/> **W3C**  
68 **Recommendation, December 2009**
- 69 **XSD** PV Biron, A Malhotra, *XML Schema Part 2: Datatypes Second Edition*,  
70 <http://www.w3.org/TR/xmlschema-2/> October 2004.

## 71 1.3 Non-Normative References

- 72 **SOA-RM** **Reference Model for Service Oriented Architecture 1.0.** SOA-RM OASIS  
73 Standard, Edited by C. Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter  
74 F Brown, Rebekah Metz. 12 October 2006. OASIS Standard. [http://docs.oasis-](http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf)  
75 [open.org/soa-rm/v1.0/soa-rm.pdf](http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf)
- 76 **WSI-BASIC** R Chumbley, J Durand, G Pilz, T Rutt , *Basic Profile Version 2.0*,  
77 <http://ws-i.org/profiles/BasicProfile-2.0-2010-11-09.html>,  
78 The Web Services-Interoperability Organization, November 2010
- 79 **WS-Calendar** **WS-Calendar Version 1.0. Edited by Toby Considine and Mike Douglas.** 30  
80 July 2011. OASIS Committee Specification 01. [http://docs.oasis-open.org/ws-](http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.pdf)  
81 [calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.pdf](http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.pdf)
- 82 **RFC6321** C. Daboo, M Douglass, S Lees *xCal: The XML format for iCalendar*,  
83 <http://tools.ietf.org/html/rfc6321>, IETF Proposed Standard, August 2011.

## 84 1.4 Namespace

85 The XML namespace [XML-ns] URI that MUST be used by implementations of this specification is:

86 <http://docs.oasis-open.org/ws-calendar/ns/streams/201602>

87 Dereferencing the above URI will produce the HTML document that describes this namespace.

88 Table 1 lists the XML namespaces that are used in this specification. The choice of any namespace prefix  
89 is arbitrary and not semantically significant.

90



91 Table 1-1: Namespaces Used in this Specification

| Prefix | Namespace   |
|--------|---|
| xs     | <a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>   |
| min    | <a href="http://docs.oasis-open.org/ws-calendar/ns/min-xcal/2015/12">http://docs.oasis-open.org/ws-calendar/ns/min-xcal/2015/12</a> |
| strm   | <a href="http://docs.oasis-open.org/ws-calendar/ns/streams/201606">http://docs.oasis-open.org/ws-calendar/ns/streams/201606</a>     |

92 The normative schemas for Streams can be found linked from the namespace document that is located at  
93 the namespace URI specified above.

## 94 1.5 Naming Conventions

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

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

98 

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

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

101 

```
<complexType name="ComponentServiceType">
```

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

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

## 106 1.6 Editing Conventions

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

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

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

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

---

## 113 2 WS-Calendar in Streams

114 Without an understanding of certain terms and conventions based in **[WS-Calendar PIM]**, the reader may  
115 have difficulty achieving complete understanding of their use in this standard. **[WS-Calendar PIM]**  
116 defines a Platform Independent Model and re-defined **[WS-Calendar]** as a semantically richer and more  
117 variable conformant Platform Specific Model (PSM). The terms PIM and PSM are used as defined in  
118 model driven architecture.

119 Streams are a Platform Specific Model conformant with the **[WS-Calendar PIM]**. Through conformance  
120 with the PIM, Streams are conformant with **[WS-Calendar]** specification for communicating duration and  
121 time to define a Schedule. **[WS-Calendar]** itself extends the well-known semantics of **[RFC5545]**.

122 In particular, the reader should take care to understand the logic of time specification and the language of  
123 inheritance as described in **[WS-Calendar PIM]**.

124 This entire section is informative, to assist the reader in understanding later sections.

### 125 2.1 When: Start, End and Duration

126 Any Interval can be fully defined by two out of these three elements: when it begins, how long it lasts, and  
127 when it ends. With any two, you can compute the third.

128 This specification assigns predominance to how long it lasts, the Duration. This approach is commonly  
129 used to request human scheduling, i.e., “Find a time when the three of us can meet for an hour.” Activities  
130 are then normally scheduled by Start Time, again to reflect human usage: “We will meet for lunch at  
131 Noon”.

132 Streams addresses the special case of consecutive Intervals, each of the same Duration, and each with  
133 an identical Payload, when adjusted for time. All Durations are known, and the Start Time for all Intervals  
134 after the first can be computed by its precedent.

### 135 2.2 Semantics of Inheritance

136 **[WS-Calendar PIM]** enables parsimony and artifact reuse through defined rules of inheritance. At its  
137 simplest, a Sequence can be relocated or replicated from one day to another, each time inheriting the  
138 start date, without being re-crafted. Similarly a start time for a single Interval can affect the start times of  
139 the other Intervals in the Sequence. Depending upon Inheritance, an Interval may become Fully Bound,  
140 i.e., defined sufficiently for execution.

141 The terms Inherit, Inheritance, and Bequeath are as defined within **[WS-Calendar PIM]**.

### 142 2.3 Semantics from MIN

143 Because **[WS-Calendar PIM]** is an information model, it does not define any particular serialization or  
144 XML elements. The platform specific model described in “WS-Calendar Minimal PIM-Conformant Schema  
145 Version 1.0 (**[MIN]**) defines the essential semantic elements in the PIM. The schema definition artifacts  
146 (**[XSD]**) from PIM are referenced by the Streams schema to define these elements.

### 147 3 Streams

148 Streams use Sequences to convey a time sequence of prices, usage, demand, response, or anything  
149 else that varies over time. Streams are used both for projections of the future and for reports about the  
150 past.

151 **[WS-Calendar]** specifies that Sequences that describe a Service be expressed as Duration within each  
152 Interval, Temporal Relations between those Intervals, and a single Start or End time for the Sequence.

153 **[WS-Calendar]** specifies that each Interval have a unique identifier (UID) that can be externally  
154 referenced. **[WS-Calendar]** further specifies that each Interval include a Temporal Relation, either direct  
155 or transitive, with all other Intervals in a Sequence. A Temporal Relation consists of the Relationship, the  
156 UID of the related Interval, and the optional Gap between Intervals.

157 **[WS-Calendar]** defines a Partition as a Sequence of consecutive Intervals. Streams are a parsimonious  
158 expression of a Partition that conforms to **[WS-Calendar]** indirectly by conforming to **[WS-Calendar PIM]**.  
159 Streams also specifies means to define *de facto* UIDs from Stream Contexts and Interval UIDs to achieve  
160 additional parsimony.

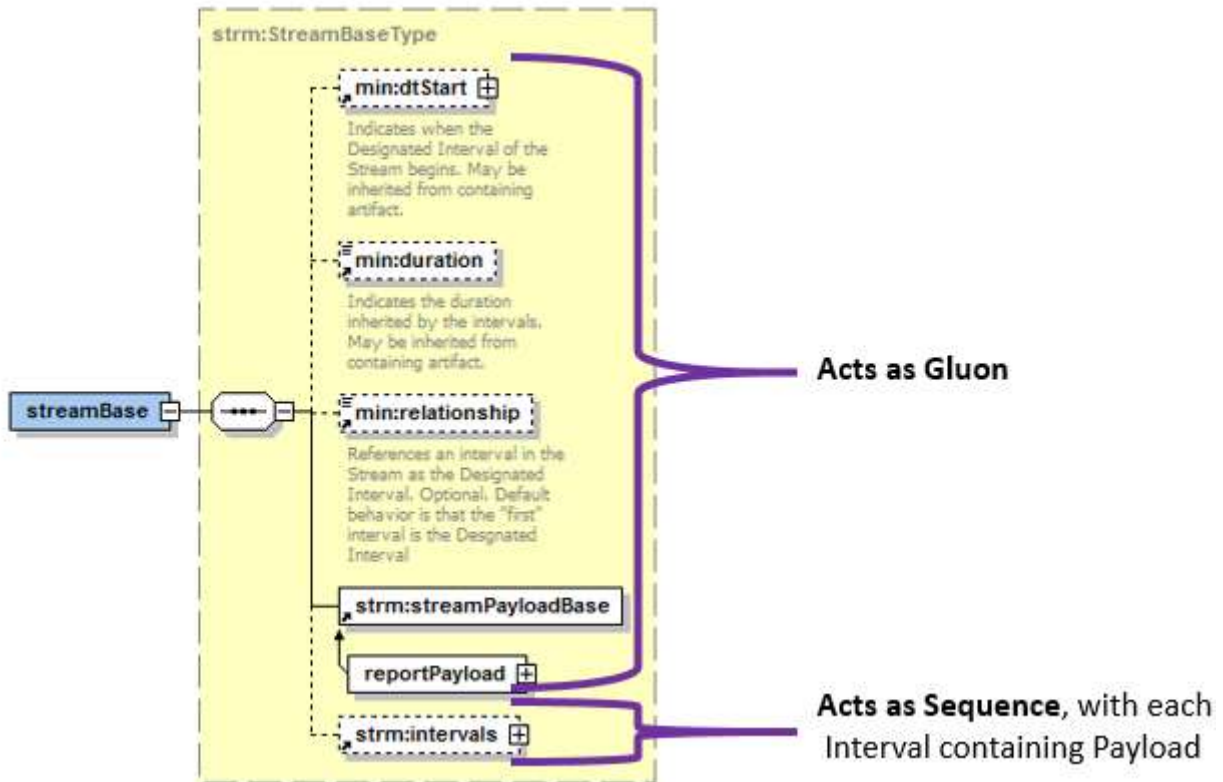
#### 161 3.1 New Semantic Elements in Streams

162 Streams may contain Intervals, each containing an informational Payload. Streams introduce their own  
163 semantic elements.

164 *Table 3-1: Core Semantics and their derivations from WS-Calendar*

| Streams Term        | Description  |
|---------------------|--|
| <b>Payload Base</b> | Payload Base is an abstract class that acts as the Artifact in each Interval. A Specification that conforms to Streams MUST specify both the Payload and inheritance rules for the Payload.  |
| <b>Relationship</b> | In <b>[WS-Calendar PIM]</b> , Relationships are defined by Relation Links and define how Intervals are connected for Binding. In Streams, there is always an implied Relationship binding the Stream Base to the first Interval in each Sequence. That interval is the Designated Interval.  |
| <b>Stream Base</b>  | The Stream Base is an abstract element that contains the “header” information (or context) for a Stream. The Stream Base specifies recurring information that applies to each Interval in the Stream. A Stream Base MAY be derived from a non-calendar application-specific context from which the information is inherited as if the context were a Gluon.  |
| <b>UID</b>          | In WS-Calendar, each Interval MUST be uniquely addressable by the UID, to support reference by an external system. In Streams, the UID is degenerate, requiring only enough uniqueness to indicate processing order between Intervals. If it is necessary to reference a particular Interval in a Stream, a unique reference is created by concatenating the Stream UID with the UID of the Stream Base. |

165 All Streams follow the Gluon-Sequence pattern from **[WS-Calendar PIM]**, i.e., the Stream Base acts a  
166 Gluon that optionally contains a degenerate Sequence. Information applied to the entire Stream is  
167 indicated in the Gluon, i.e., external to the Intervals of the Sequence. Only information that changes over  
168 time is contained within each Interval. This changing information is referred to herein as the Payload.

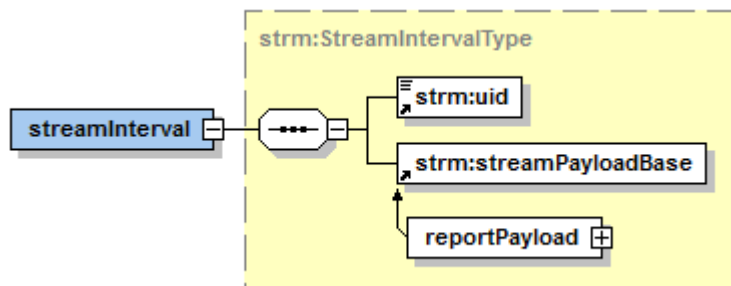


169  
170 *Figure 3-1: Stream as Gluon-Equivalent and Degenerate Sequence*

171 For example, an associated transaction, a request for telemetry, or even a service definition MAY  
172 establish a context, and that context acts as a Gluon with respect to the Stream Base. The Stream Base  
173 MAY inherit information in the Context. Each Interval in the Stream inherits information from the Stream  
174 Base. WS-Calendar PIM calls this the *Lineage* of the information.

### 175 3.2 Intervals and Unique Identifiers

176 XML processing rules do not require that order is preserved when a collection is processed. For a  
177 Stream, it is necessary that the receiver be able to order the de-serialized Intervals for proper  
178 interpretation. To this end, each Interval in a Stream contains a UID.



179  
180  
181 *Figure 3-2: Interval, the components of a Sequence*

182 The Stream UID is a sortable element that can be used to order the Intervals after processing. The  
183 unique identifiers (UID) mandated by [WS-Calendar] can be verbose; as Streams may contain hundreds  
184 or even thousands of Intervals, the overhead for expressing a [WS-Calendar] UID for each Interval could  
185 be considerable. [WS-Calendar PIM] is less specific as to how identifiers are constructed. Stream UIDs

186 MUST only be unique within the Stream: each Interval is uniquely identified by a Stream UID within the  
187 Stream.

188 Streams augment the inheritance pattern of **[WS-Calendar PIM]** by extending it to the UID. Where each  
189 Interval in **[WS-Calendar]** MUST have a uniquely addressable UID, in Streams, an addressable UID MAY  
190 be constructed through concatenation of the Interval ID with UIDs inherited from the Stream.

191 If it is necessary to instantiate an Interval in the Sequence as a **[WS-Calendar PIM]** conformant Interval,  
192 the GUID for each Interval MAY be derived by (e.g.) appending the Sequence ID to the Stream's UID. If it  
193 is necessary to further differentiate the UID of a particular instance of a Stream, it MAY be concatenated  
194 with the UIDs of whatever references and context information is acting as a Gluon for that Stream. In this  
195 way, Unique Identifiers for each Interval in each instance of a Stream can be created by concatenation of  
196 UIDs from each object acting as a Gluon.

197 Specifications claiming conformance with Streams MUST specify the mechanism of this concatenation,  
198 i.e., appending the Stream Interval UID to the Stream UID.

### 199 **3.3 Streams: a Restricted Profile for Sequences and Intervals**

200 While this specification is conformant with **[WS-Calendar PIM]**, this specification further defines standard  
201 profiles of Sequences and Intervals for use in Streams.

202 Streams describe Partitions. Within a Stream expressed using Durations, a virtual UID for each Interval  
203 MAY be constructed by concatenating the Stream Identifier, which MAY include the identity of the source  
204 or recipient, and a sequence number. Within a Stream, this Stream Interval UID can be expressed within  
205 each Interval by the sequence number alone.

206 If the Designated Interval in a Sequence within a Stream omits a Temporal Relationship, then all Intervals  
207 in the Sequence MUST NOT include a Temporal Relation. Such Intervals are sorted by increasing  
208 sequence number (expressed in the UID), and each Interval is treated as if it contained an implied  
209 *FinishToStart* relation to the next Interval with a Gap of zero Duration.

210 Partitions expressed in this way consist of Intervals containing only a Sequence Number, the Duration of  
211 the Interval (if not inherited), and the Payload. The effect of this is that Stream Intervals are ordered as a  
212 Partition in order of increasing UID.

213 WS-Calendar inheritance defines a Lineage whereby Intervals inherit information from Gluons. In Energy  
214 Interoperation, Streams are contained in larger messages. A Stream MAY inherit information from its  
215 containing message as if from a Gluon. A Stream-derived Type MAY contain information external to the  
216 Sequence. This information inherits acts as if it were a Gluon, inheriting from the containing message,  
217 and Bequeathing information to the Designated Interval in the Sequence.

218 The first Interval in the Sequence conveyed by a Stream is the Designated Interval unless another  
219 Interval is explicitly so designated in the Stream Base. These terms are defined below.

### 220 **3.4 Observational Data expressed as Streams**

221 Observed information may be best communicated as raw data without interpretation. A single set of  
222 Observations may be re-purposed or re-processed for multiple uses. For example, a measurement  
223 recorded at 3:15 may be a point in both a 5-minute series and a 15-minute series. Observational data  
224 may have known errors. Low-end sensor systems may not update instantly. For example, a reading for  
225 4:30 P.M. may be known to actually have been recorded at 4:27 P.M. Streams expressing a series of  
226 observations MAY use date and time rather than the duration as their primary temporal element.  
227 Conforming applications and specifications SHALL describe how observational data is mapped to Stream  
228 Intervals.

229 When an Interval in a Stream are expressed with Date and Time, then all Intervals in that Sequence  
230 SHALL be expressed with a Date and Time and that boundary selected SHALL be the Same, i.e., all  
231 Intervals MAY be expressed with a Begin Date and Time OR with an End Date and Time. For  
232 observations, typical implementations use the End Date and Time.

233 Within a Stream expressed using Dates and Times, a virtual UID for each Interval MAY be constructed by  
234 concatenating the Signal Identifier, and an inherited context ID and the Date and Time. Within an  
235 Observational Stream, this UID can be expressed within each Interval by the End Date and Time alone.

236 Intervals in a Sequence expressed this way are treated as if each contains an implied FinishToStart  
237 relation to the next Interval with a Gap of zero duration. The Duration of each Interval can be computed  
238 by using the Date(s) and Time(s) of adjacent Intervals.

### 239 **3.5 Payload Optimization in Streams**

240 As defined in **[WS-Calendar PIM]**, each Interval in a Sequence potentially contains an Artifact that  
241 inherits/extends the WS-Calendar Artifact as a Payload. As used in Streams, this Artifact is expressed  
242 once or inherited from the service context. Each Interval in a Stream expresses only the common subset  
243 of facts that varies within the context of the Stream. For efficient communication and processing, Streams  
244 use these explicit processing rules:

- 245 1. Unless each Interval includes a full Payload, each Interval in a Stream expresses only the defined  
246 subset of the Payload that varies over time.
- 247 2. Each Interval in a Stream uses the same Payload subset as all other Intervals in that Stream.
- 248 3. All Streams in this specification share a common Payload Base. This commonality is derived from  
249 the commonality of a request for future performance, telemetry reporting performance, conveying  
250 baselines, and submitting projections.

### 251 **3.6 Extending Stream Payloads**

252 Streams does not limit the Payload, but only requires that the Payload be derived from the Payload Base.

253 It may be necessary to qualify information about Intervals in the future, i.e. indicate the probability of  
254 accuracy or some other information. This specification does not address this information requirement.

255 It may be necessary to qualify measurements delivered in a report. Devices have known accuracies.  
256 Several Measurements MAY be added together to create a single quantity. A particular reading among  
257 many may be estimated or interpolated. To support these uncertainties different Payloads would  
258 generally be defined for different services.



---

## 259 4 Conformance

### 260 4.1 Conformance Points

261 We define two conformance points for WS-Calendar Streams:

- 262 (1) Conformance of an application to Streams
- 263 (2) Conformance of a specification to Streams

264 Note that the term *implementation* may apply to both an application that uses Streams and a specification  
265 that extends or otherwise reuses Streams.

### 266 4.2 Conformance of Streams to WS-Calendar-PIM

267 Applications and specifications claiming conformance SHALL implement all inheritance and semantic  
268 rules as described in **[WS-Calendar-PIM]** Section 5, treating the Stream Base behavior as that of a Gluon

269 Applications and specifications claiming conformance to Streams SHALL conform to PIM Section 6  
270 subsections 6.1, 6.3, and 6.4.

271 Applications and specifications claiming conformance SHALL include all functions and schema  
272 representations of Stream. Extensions are permitted, but all extensions MUST be documented in the  
273 conforming application or specification conformance statement(s).

274 If it is necessary to process a Stream through standard Calendar communications, a Stream SHALL be  
275 processed as if it were a Gluon.

276 All Sequence information MAY remain internal to that Gluon.

277 If it is necessary to instantiate Interval in the Sequence as a WS-Calendar or PIM Interval, the UID for  
278 each instantiated Interval MAY be derived by concatenating the Stream Interval UID to the Stream UID.  
279 Conforming applications or specifications SHALL define that concatenation.

### 280 4.3 Inheritance within Streams

281 Streams are a means of conveying informational payloads that vary over time, optimized for concise  
282 expression. It may be desirable for those payloads themselves to be optimized by reducing the  
283 expression of redundant information.

284 Specifications and applications claiming conformance SHALL use the **[WS-Calendar PIM]** pattern of  
285 inheritance, and MUST explicitly define the Gluon equivalent(s) for their specification or application,  
286 including describing the inheritance rules for the payloads.

287 Conforming Streams MAY inherit from structures external to any particular Streams instance, so long as  
288 the specification requires that the information be conveyed by a discoverable artifact or chain of artifacts  
289 acting as Gluons. Such Gluons are considered to enter the Lineage of the Stream for purposes of **[WS-**  
290 **Calendar PIM]** conformance, and are inherited by each Interval.

### 291 4.4 Stream expression of Intervals expressed as Durations

292 Streams describe Partitions. Within a Stream expressed using Durations, a UID for each such Interval  
293 MAY be constructed by concatenating the Stream UID (which may include the identity of the source or  
294 recipient) and the Stream Interval UID, which MAY be as simple as a sequence number.<sup>1</sup> Conforming  
295 applications and specifications SHALL describe that concatenation and construction of Stream Interval  
296 UIDs.

---

<sup>1</sup> Within a Stream, this UID can be expressed within each Interval by the sequence number alone.

297 If the Designated Interval in a Sequence within a Stream omits a Temporal Relationship, then Intervals in  
298 that Sequence MAY NOT include a Temporal Relation. Such Intervals are sorted by increasing Stream  
299 Interval UID and each Interval is treated as if it contained an implied FinishToStart relation to the next  
300 Interval with a Gap of zero Duration.

301 Partitions expressed in this way consist of Intervals containing only a Sequence Number, the Duration of  
302 the Interval (if not inherited), and the Payload. The effect of this is that Stream Intervals are ordered as a  
303 Partition in order of increasing UID.

304 **[WS-Calendar-PIM]** inheritance defines a Lineage whereby Intervals inherit information from Gluons. In  
305 Energy Interoperation, Streams are contained in larger messages. A Stream MAY inherit information from  
306 its containing message as if from a Gluon. A Stream-derived Type may contain information external to the  
307 Sequence. This information inherits acts as if it were a Gluon, inheriting from the containing message,  
308 and Bequeathing information to the Designated Interval in the Sequence. Conforming applications and  
309 specifications SHALL describe how to determine the values associated with any Stream Interval.

310 The first (in time and in sequence number) Interval in the Sequence in a Stream is the Designated  
311 Interval unless another Interval is explicitly so designated in the Stream Base or other artifact acting as a  
312 Gluon. Conforming applications or specifications SHALL describe how to determine the Designated  
313 Interval.

## 314 **4.5 Conformance for Observational Data**

315 A conforming application or specification SHALL apply all mandatory statements in Section 3.4. If optional  
316 or extended behavior is supported the conforming application or specification SHALL specify all optional  
317 or extended behavior.

## 318 **4.6 Conformance for Stream Payloads and Optimizing Inheritance**

319 If the Designated Interval in a Series has a single element consisting of the Payload only, all Intervals in  
320 the Sequence MUST include only a payload element. Conforming applications and specifications SHALL  
321 describe any constraints on Stream Payloads.



---

## 322 Appendix A. Acknowledgments

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

325 **Participants:**

326 David Thewlis, CalConnect

327 William Cox, Individual

328 Gershon Janssen, Individual

329 Benoit Lepeuple, LonMark International

330 Michael Douglass, Rensselaer Polytechnic Institute

331 Toby Considine, University of North Carolina at Chapel Hill

332 Chris Bogen, US Department of Defense (DoD)

333

334 Streams were originally developed in the OASIS Energy Interoperation. We are grateful for their  
335 contribution to WS-Calendar.

336

## Appendix B. Revision History

337

| Revision | Date             | Editor                         | Changes Made  |
|----------|------------------|--------------------------------|---|
| WD01     | 8-November-2012  | Toby Considine                 | Initial Draft   |
| WD02     | 27-March-2013    | Toby Considine                 | Editing issues per comments<br>Removed spurious references to Energy Interoperation   |
| WD03     | 13-May 2013      | Toby Considine                 | Added references to WS-Calendar PIM<br>Re-wrote conformance to rely on PIM<br>Clarified issues with building GUIDs [UIDs] from sequence through Inheritance |
| WD04     | 20-May-2013      | Toby Considine                 | Numerous consistency issues from TC comments  |
| WD05     | 29-December-2014 | Toby Considine                 | Re-targeted conformance toward the recently completed WS-Calendar PIM   |
| WD06     | 8-December-2015  | Toby Considine                 | Removed MIN etc.  |
| WD07     | 7-February-2016  | Toby Considine                 | First full re-draft after transition from MPC to MIN, first PR of MIN   |
| WD08     | 25-March-2016    | Toby Considine                 | Fixed references, minor comments from last review   |
| WD09     | 30-May-2016      | Toby Considine                 | Addressed additional comments from last review  |
| WD10     | 2-June-2016      | William Cox,<br>Toby Considine | Conformance clarification. More consistent model description and terminology  |
| Wd11     | 2-June-2016      | Toby Considine                 | Updated Fields and References   |

338