Schedule Signals and Streams Version 1.0

Committee Specification Draft 03

03 June 2016

Specification URIs

This version:

http://docs.oasis-open.org/ws-calendar/streams/v1.0/csd03/streams-v1.0-csd03.pdf (Authoritative)
http://docs.oasis-open.org/ws-calendar/streams/v1.0/csd03/streams-v1.0-csd03.html
http://docs.oasis-open.org/ws-calendar/streams/v1.0/csd03/streams-v1.0-csd03.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), University of North Carolina at Chapel Hill

Editors:

Toby Considine (toby.considine@unc.edu), University of North Carolina at Chapel Hill
William T. Cox (wtcox@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/csd03/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.

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. 03 June 2016. OASIS Committee Specification Draft 03. http://docs.oasis-open.org/ws-calendar/streams/v1.0/csd03/streams-v1.0-csd03.html. Latest version: http://docs.oasis-open.org/ws-calendar/streams/v1.0/streams-v1.0.html.

Notices

Table of Contents

Table of Figures

Table of Tables

1 Introduction

All text is normative unless otherwise labeled

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

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

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

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

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

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

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

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

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

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

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

ISO8601 ISO (International Organization for Standardization). *Representations of dates and times, third edition*, December 2004, (ISO 8601:2004)

MIN ***WS-Calendar Minimal PIM-Conformant Schema Version 1.0*. Edited by Toby Considine and William Cox. 18 December 2015.** OASIS Committee Specification Draft 01. December 2015, http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.pdf

RFC2119 S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

RFC5545 B. Desruisseaux *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, http://www.ietf.org/rfc/rfc5545.txt, IETF RFC5545, proposed standard, September 2009

WS-Calendar PIM *“WS-Calendar Platform Independent Model (PIM) Version 1.0”.* Edited by William T. Cox and Toby Considine. 21 August, 2015. OASIS Committee Specification 02. http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.pdf

XML NAMES **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**

XSD PV Biron, A Malhotra, XML Schema Part 2: Datatypes Second Edition, http://www.w3.org/TR/xmlschema-2/ October 2004.

1.3 Non-Normative References

SOA-RM ***Reference Model for Service Oriented Architecture 1.0.*** SOA-RM OASIS Standard, Edited by C. Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown, Rebekah Metz. 12 October 2006. OASIS Standard. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

WSI-BASIC R Chumbley, J Durand, G Pilz, T Rutt , *Basic Profile Version 2.0,* http://ws-i.org/profiles/BasicProfile-2.0-2010-11-09.html, The Web Services-Interoperability Organization, November 2010

WS-Calendar ***WS-Calendar Version 1.0.* Edited by** Toby Considine and Mike Douglas. 30 July 2011. OASIS Committee Specification 01. http://docs.oasis-open.org/ws-calendar/ws-calendar/v1.0/ws-calendar-1.0-spec.pdf

RFC6321 C. Daboo, M Douglass, S Lees *xCal: The XML format for iCalendar*, http://tools.ietf.org/html/rfc6321, IETF Proposed Standard, August 2011.

1.4 Namespace

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

```
http://docs.oasis-open.org/ws-calendar/ns/streams/201602
```

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

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

Table 1-1: Namespaces Used in this Specification

Prefix	Namespace
xs	http://www.w3.org/2001/XMLSchema
min	http://docs.oasis-open.org/ws-calendar/ns/min-xcal/2015/12
strm	http://docs.oasis-open.org/ws-calendar/ns/streams/201606

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

1.5 Naming Conventions

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

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

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

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

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

For the names of intents, the names follow the lowerCamelCase 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.

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

1.6 Editing Conventions

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.

All elements in the tables not marked as “optional” are mandatory.

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

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

2 WS-Calendar in Streams

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

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

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

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

2.1 When: Start, End and Duration

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

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

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

2.2 Semantics of Inheritance

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

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

2.3 Semantics from MIN

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

3 Streams

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

[WS-Calendar] specifies that Sequences that describe a Service be expressed as Duration within each Interval, Temporal Relations between those Intervals, and a single Start or End time for the Sequence. **[WS-Calendar]** specifies that each Interval have a unique identifier (UID) that can be externally referenced. **[WS-Calendar]** further specifies that each Interval include a Temporal Relation, either direct or transitive, with all other Intervals in a Sequence. A Temporal Relation consists of the Relationship, the UID of the related Interval, and the optional Gap between Intervals.

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

3.1 New Semantic Elements in Streams

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

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

Streams Term	Description
Payload Base	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.
Relationship	In **[WS-Calendar PIM],** 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.
Stream Base	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.
UID	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.

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

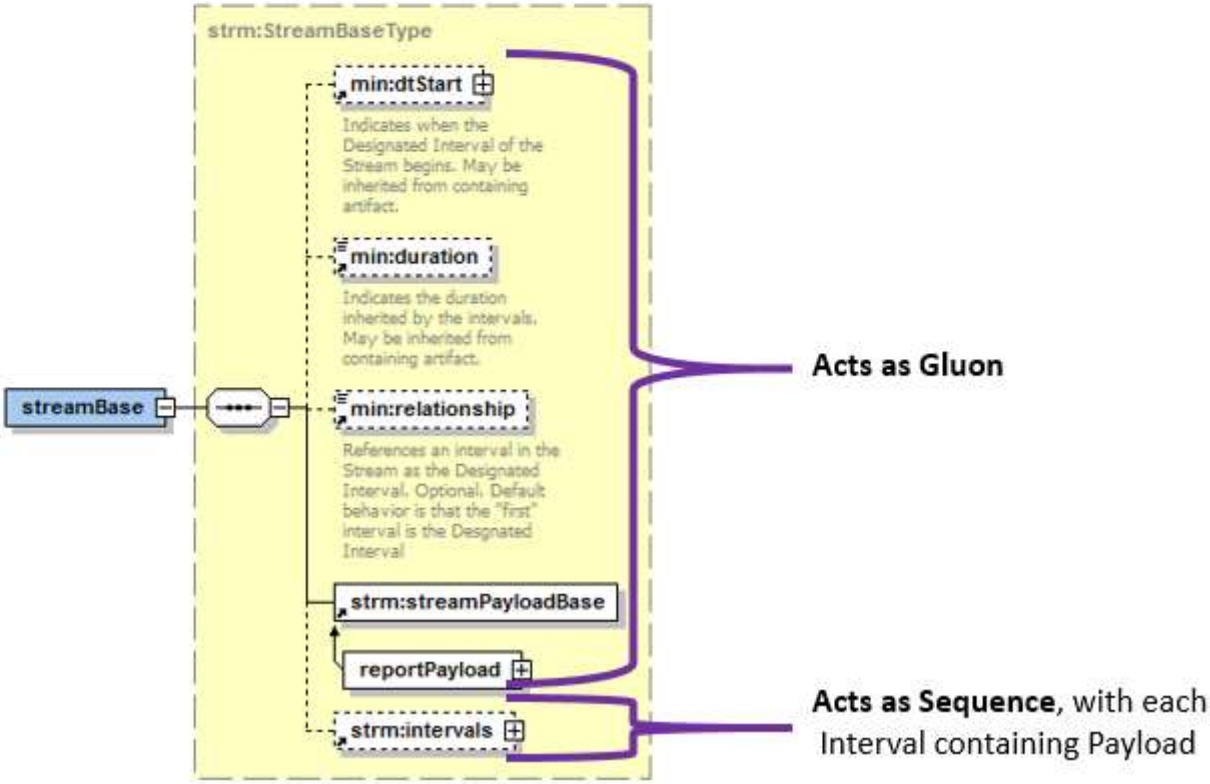

Figure 3-1: Stream as Gluon-Equivalent and Degenerate Sequence

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

3.2 Intervals and Unique Identifiers

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

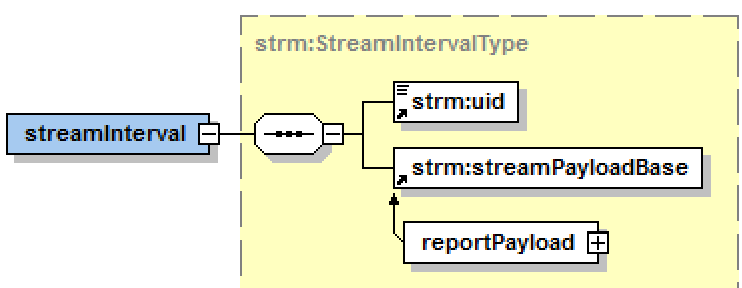

Figure 3-2: Interval, the components of a Sequence

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

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

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

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

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

3.3 Streams: a Restricted Profile for Sequences and Intervals

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

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

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

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

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

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

3.4 Observational Data expressed as Streams

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

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

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

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

3.5 Payload Optimization in Streams

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

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

3.6 Extending Stream Payloads

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

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

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

4 Conformance

4.1 Conformance Points

We define two conformance points for WS-Calendar Streams:

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

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

4.2 Conformance of Streams to WS-Calendar-PIM

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

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

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

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

All Sequence information MAY remain internal to that Gluon.

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

4.3 Inheritance within Streams

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

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

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

4.4 Stream expression of Intervals expressed as Durations

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

---

[1] Within a Stream, this UID can be expressed within each Interval by the sequence number alone.

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

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

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

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

4.5 Conformance for Observational Data

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

4.6 Conformance for Stream Payloads and Optimizing Inheritance

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

Appendix A. Acknowledgments

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

Participants:

David Thewlis, CalConnect
William Cox, Individual
Gershon Janssen, Individual
Benoit Lepeuple, LonMark International
Michael Douglass, Rensselaer Polytechnic Institute
Toby Considine, University of North Carolina at Chapel Hill
Chris Bogen, US Department of Defense (DoD)

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

Appendix B. Revision History

Revision	Date	Editor	Changes Made
WD01	8-November-2012	Toby Considine	Initial Draft
WD02	27-March-2013	Toby Considine	Editing issues per comments Removed spurious references to Energy Interoperation
WD03	13-May 2013	Toby Considine	Added references to WS-Calendar PIM Re-wrote conformance to rely on PIM 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, Toby Considine	Conformance clarification. More consistent model description and terminology
Wd11	2-June-2016	Toby Considine	Updated Fields and References