Schedule Signals and Streams Version 1.0

Committee Specification Draft 02

24 May 2013

Specification URIs

This version:

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

Previous version:

http://docs.oasis-open.org/ws-calendar/streams/v1.0/csprd01/streams-v1.0-csprd01.pdf (Authoritative)
http://docs.oasis-open.org/ws-calendar/streams/v1.0/csprd01/streams-v1.0-csprd01.html
http://docs.oasis-open.org/ws-calendar/streams/v1.0/csprd01/streams-v1.0-csprd01.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

Editor:

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

Additional artifacts:

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

- XML schemas: http://docs.oasis-open.org/ws-calendar/streams/v1.0/csd02/xsd/

Related work:

This specification is related to:

- *WS-Calendar Version 1.0*. 30 July 2011. OASIS Committee Specification 01. http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.html

Declared XML namespaces:

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

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 WS-Calendar. 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.

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" button on the Technical Committee's web page at 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).

Citation format:

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

[streams-v1.0]

Schedule Signals and Streams Version 1.0. 24 May 2013. OASIS Committee Specification Draft 02. http://docs.oasis-open.org/ws-calendar/streams/v1.0/csd02/streams-v1.0-csd02.html.

Notices

Table of Contents

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, 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 also defines the Sequence, in which a set of temporally related calendar information objects, known as 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.

[WS-Calendar] is a general specification and makes no assumptions about how its information model is used. **[WS-Calendar]** has specific rules which 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.

Even so, WS-Calendar does not define a normative structure for the information conveyed. WS-Calendar is primarily 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. Even legal and conformant conveyances of calendar information may fail to meet the requirements for basic interoperability requirements **[WSI-Basic].**

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]**.

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

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)

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

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

SOA-RM SOA-RM OASIS Standard, *OASIS Reference Model for Service Oriented Architecture 1.0*, October 2006 http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

WS-Calendar WS-Calendar OASIS Committee Specification, WS-Calendar Version 1.0, July 2011, http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.pdf

WS-Calendar PIM WS-Calendar OASIS Committee Working Draft, *"WS-Calendar Platform Independent Model (PIM) Version 1.0 WD05"*, https://www.oasis-open.org/committees/download.php/48936/ws-calendar-pim-v1.0-wd05.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

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

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-cd01.pdf October 2009.

1.3 Non-Normative References

[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

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
```

Dereferencing the above URI will produce the Resource Directory Description Language [RDDL 2.0] 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
xcal	urn:ietf:params:xml:ns:icalendar-2.0
strm	http://docs.oasis-open.org/ws-calendar/ns/streams

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

[WS-Calendar] defines how to use the semantics of the enterprise calendar communications within service communications. **[WS-Calendar PIM]** defines how conformance to [**WS-Calendar**] is to be achieved on platforms that cannot themselves interact directly with traditional calendar servers.

Streams are conformant with the **[WS-Calendar PIM]**, the platform independent model (PIM) for **[WS-Calendar]**. 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]**.

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

2.1 Schedule Semantics from WS-Calendar PIM (Non-Normative)

Without an understanding of certain terms defined in **[WS-Calendar PIM]**, the reader may have difficulty achieving complete understanding of their use in this standard. The table below provides summary descriptions of certain key terms from that specification. This specification does not redefine these terms; they are listed here solely as a convenience to the reader.

Table 2-1: Core Semantics from WS-Calendar

WS-Calendar Term	Description
Artifact	The placeholder in an Component that holds that thing that occurs during an Interval. **[**EMIX Product Descriptions populate Schedules as Artifacts inside Intervals. In Streams, this specification refers to the Payload conveyed by an Interval.
Availability	Availability in this specification refers to the Vavailability Component, itself a collection of recurring Availability parameters each of which expresses set of Availability Windows. In this specification, these Windows may indicate when an Interval or Sequence can be Scheduled, or when a partner can be notified, or even when it cannot be Scheduled.
Component	In **[iCalendar]**, the primary information structure is a Component, also referred to as a “vcomponent.” A Component is refined by Parameters and can itself contain Components. Several RFCs have extended iCalendar by defining new Components using the common semantics defined in that specification. In the list below, Interval, Gluon, and Availability are Components. Duration, Link, and Relationship are Parameters. A Sequence is set of Components, primarily Intervals and Gluons, but is not itself a Type.
Duration	Duration is the length of time for an event scheduled using iCalendar or any of its derivatives. The XCAL [**RFC 6321]** duration is a data type using the string representation defined in the iCalendar (**[RFC5545]**) Duration.
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.
Interval	The Interval is a single discrete segment, an element of a Sequence, and expressed with a Duration. The Interval is derived from the common calendar Components. An Interval is part of a Sequence.

WS-Calendar Term	Description
Link	A reference to an internal object within the same calendar, or an external object in a remote system. The Link is used by one **[WS-Calendar]** Component to reference another.
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.
Relation Link	Links between Components.
Sequence	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, and can be scheduled by scheduling that single Interval in that Sequence.

Normative descriptions of the terms in the table above are in **[WS-Calendar]**.

2.2 Schedules and Inheritance

Nearly every response, every event, and every interaction can have payloads with values that vary over time, i.e., it a set of intervals can be using a Sequence of Intervals. Many market communications involve information about or a request for power delivered over a single interval of time. Simplicity and parsimony of expression must coexist with complexity and syntactical richness.

Consider a request to reduce power consumption in response to market conditions on a smart grid (Demand Response). The simplest demand response is to reduce power for a set interval.

Units:	KW	Quantity	10

Figure 2-1: Basic Power Object from EMIX

At its simplest, though, WS-Calendar expresses repeating intervals of the same duration, one after the other, and something that changes over the course of the schedule

Start:	8:00	Duration:	1Hour		
		Duration:	1Hour		
		Duration:	1Hour		
		Duration:	1Hour		
		Duration:	1Hour		

Figure 2-2: WS-Calendar Partition, a simple sequence of 5 intervals

The WS-Calendar specification defines how to spread an object like the first over the schedule. The information that is true for every interval is expressed once only. The information that changes during each interval, is expressed as part of each interval.*

Units	KW	Start:	8:00	Duration:	1Hour	Quantity	10
				Duration:	1Hour	Quantity	10
				Duration:	1Hour	Quantity	15
				Duration:	1Hour	Quantity	25
				Duration:	1Hour	Quantity	10 *

Figure 2-3: Applying Basic Power to a Sequence

Many communications communicate requirements for a single interval. When expressing market information about a single interval, the market object (Power) and the single interval collapse to a simple model:

Units	KW	Start:	8:00	Duration:	1Hour	Quantity	10

Figure 2-4: Simplifying back to Power in a Single Interval

WS-Calendar calls this pattern Inheritance and specifies a number of rules that govern Inheritance. Table 2-2 summarizes those terms defined in WS-Calendar to describe Inheritance that are used in this specification as well. This specification does not redefine these terms; they are listed here solely as a convenience to the reader.

Table 2-2: WS-Calendar Semantics: Inheritance (non-normative)

Streams Term	Definition
Lineage	The ordered set of Parents that results in a given inheritance or execution context for a Sequence.
Inherit	A Child Inherits attributes (Inheritance) from its Parent.
Inheritance	A pattern by which information in Sequence is completed or modified by information from a Gluon. Information specified in one informational object is considered present in another that is itself lacking expression of that information.
Bequeath	A Parent Bequeaths attributes (Inheritance) to its Children.

Normative descriptions of the terms in the table above are in **[WS-Calendar]**.

This specification extends the use of Inheritance as defined in WS-Calendar. Each Interval in a Stream contains an information payload. Each of these payloads is completed through inheriting information from the Stream as if from a Gluon. The Stream itself inherits information from the context of the interaction or information, as if from Gluon.

A higher-level object Bequeaths essential information to a Stream, which in turn its information to each Interval in the Stream. This specification uses this pattern of expression throughout.

3 Streams

Streams use WS-Calendar 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; event signals and reports are each instances of Streams.

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]** by conforming to **[WS-Calendar PIM].**

3.1 New Semantic Elements in Streams

Streams may contain Intervals, each containing an informational payload. .Intervals MAY contain any property defined in WS-Calendar. Streams also 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.
Stream Base	The Stream Base is an abstract element that contains the “header” information for a Stream. The Stream Base specifies recurring information that applies to each Interval in the Stream. A Stream Base may be related to a context from which the recurring 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 and the Uids of any artifiacts acts as a Gluon, including that of the Stream Base.

All Streams follow the Gluon-Sequence pattern from WS-Calendar, i.e., the Stream Base acts a Gluon that optionally contains a degenerate Sequence. Information valid for 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.

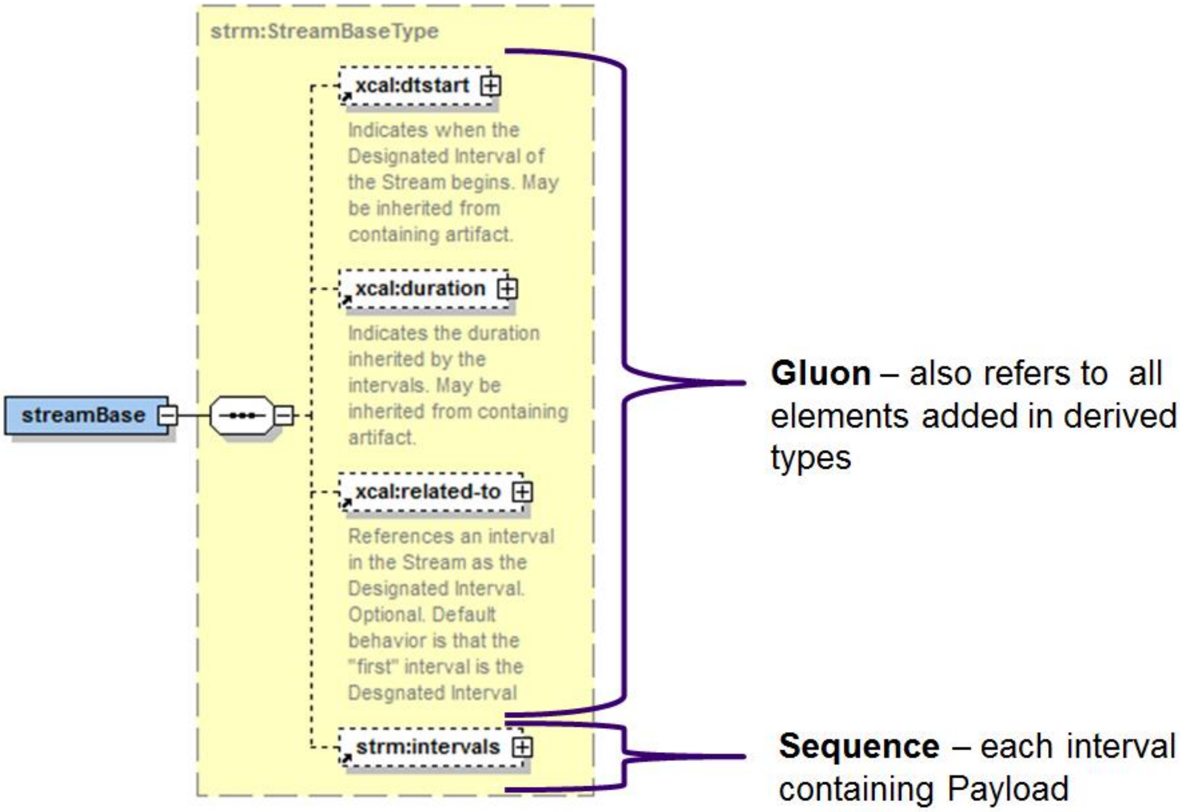

Figure 3-1: Stream as Gluon and Sequence

For example, an associated transaction or even a service definition MAY establish a context, which context acts as a Gluon 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 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 intervals for proper interpretation. To this end, each Interval in a Stream contains a Uid.

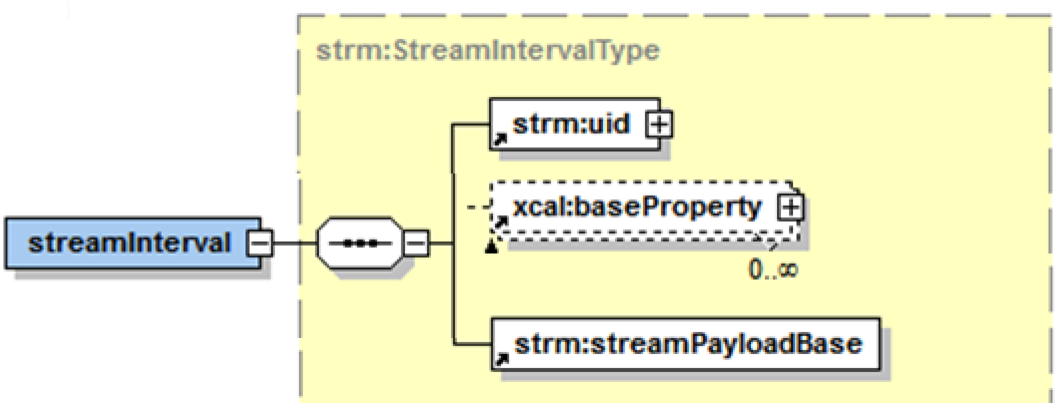

Figure 3-2: Interval, the components of a Sequence

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 Uid for each interval could be considerable. Stream UIDs must only be unique within the Stream, each intervals is uniquely identified by a Stream UID.

Streams augment the inheritance pattern of **[WS-Calendar]** 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 by concatenation with inherited UIDs.

If it is necessary to instantiate an Interval in the Sequence as a WS-Calendar Interval, the UID for each Interval is derived by 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 concatenation of UIDs from each object acting as a Gluon.

Specifications claiming conformance with Streams MUST specify the mechanism of this concatenation, i.e., concatenation could be by either pre-pending or by appending.

3.3 UML Diagram of Stream

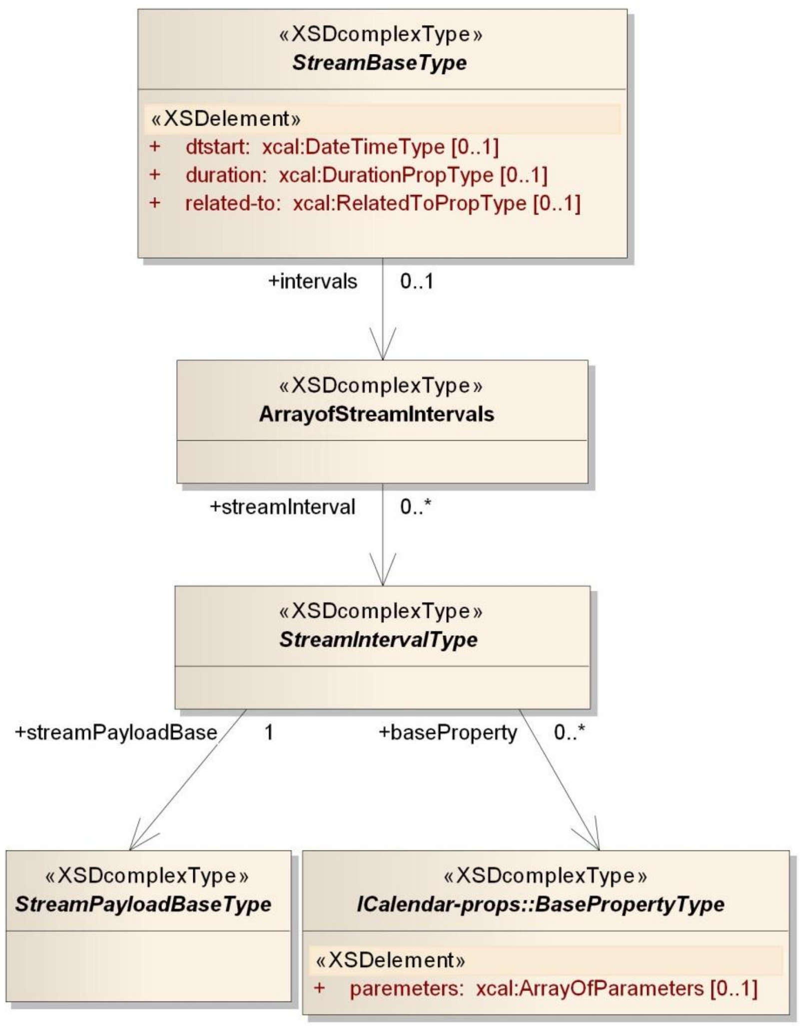

Figure 3-3: UML Class Diagram of abstract StreamBase class

3.4 Stream expression of Intervals expressed as Durations

While conformant specifications can include anything expressible in [WS-Calendar], 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 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 (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 Event. Signals, Reports, and many other messages use this pattern of expression. For example, the Active Period of an Event Bequeaths its start date and time to an Event Signal which Bequeaths that to the Designated Interval in the sequence. These terms are defined below.

3.4.1 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 that can be lost in processing. Low-end sensor systems may not update instantly. For example, a reading taken at 4:30 may be known to actually have been recorded at 4:27. Streams expressing a series of observations MAY use the date and times rather than the duration as their primary temporal element.

When the boundaries of Intervals 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, 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, the and a unique ID (which may be the service 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, each Interval in a Sequence potentially contains any 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.

All streams in this specification share a common Payload base. This commonality is derived from the commonality of a request for performance (Signal), a report of performance (Report and Delivery), projections of performance (Projection), and a baseline of performance (Baseline).

3.6 Other elements in Stream Payloads

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. To support these uncertainties different payloads are defined for different services.

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

4 Conformance

4.1 Conformance with the Semantic Models of WS-Calendar-PIM

This section specifies conformance with the semantic models of **[WS-Calendar-PIM]**. This specification requires that specifications claiming conformance also conform to the specific conformance requirements of **[WS-Calendar-PIM]** are described in section 5.3 of that specification, *“Conformance Rules for WS-Calendar PIM”.*

4.2 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 claiming conformance SHALL use a similar pattern of inheritance, and MUST make explicit what the Gluon equivalent for their specification is, including defining 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, and are inherited by each Interval.

4.2.1 Conformance of Streams to WS-Calendar-PIM

If it is necessary to process a Stream through standard Calendar communications, the Stream’s GUID is the key and the Stream is processed as if 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 Interval, the GUID for each is derived by appending the Sequence ID to the Stream’s GUID.

4.2.1.1 Stream expression of Intervals expressed as Durations

While conformant communications can include anything expressible in [WS-Calendar], 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 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 MAY 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 Market Signal 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.

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 Event. Signals, Reports, and many other messages use this pattern of expression. For example, the Active Period of an Event

Bequeaths its start date and time to an Event Signal which Bequeaths that to the Designated Interval in the sequence. These terms are defined below.

4.2.1.2 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 that can be lost in processing. Low-end sensor systems may not update instantly. For example, a reading taken at 4:30 may be known to actually have been recorded at 4:27. Streams expressing a series of observations MAY use the date and times rather than the duration as their primary temporal element.

When the boundaries of Intervals 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, 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.

4.2.2 Conformance of Streams to WS-Calendar

Specifications that conform to **[WS-Calendar-PIM]** also conform to **[WS-Calendar]** as described in Section 5.1 “Relationship to WS-Calendar” of **[WS-Calendar-PIM]**.

4.2.2.1 Specific Rule for Optimizing Inheritance

If the Designated Interval in a Series has a single element of the Payload only, all Intervals in the Sequence convey only that payload element.

4.3 Claiming Conformance to Streams

Specifications claiming conformance to Streams must specify inheritance rules.

4.3.1 Conformance to Lineage

A specification claiming conformance to Streams must specify what artifacts act as Gluons and specify any special rules of inheritance. For example, telemetry would tend to measure one thing again in each interval. That one thing MAY be specified in the Stream Base, enabling a Stream Artifact to be fully understood on its own. Alternately, there may be some artifact that describes the measured element, which acts as a Gluon to the Stream Base.

A specification claiming conformance to Streams must make explicit the inheritance rules the define the lineage, i.e., that disambiguate the payload in the Stream.

4.3.2 Construction of Referenceable Identifier

WS-Calendar requires that each interval be uniquely referenceable by an entity external to the system. Identifiers within Intervals of a Stream must only be unique within that sequence. A Stream may contain more than one Sequence. A Stream itself may only be identifiable within a specific context.

A specification claiming conformance to Streams MUST specify how a unique identifier can be constructed using the inheritance of each Sequence.

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 from sequence through Inheritance
WD04	20-May-2013	Toby Considine	Numerous consistency isses from TC comments

