# WS-Calendar Minimal PIM-Conformant Schema Version 1.0

## Committee Specification Draft 01 / Public Review Draft 01

## 18 December 2015

### Specification URIs

**This version:**

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

**Previous version:**

N/A

**Latest version:**

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

**Technical Committee:**

OASIS Web Services Calendar (WS-Calendar) TC

**Chair:**

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

**Editors:**

Toby Considine (toby.considine@unc.edu), University of North Carolina
William T. Cox (wtcox@coxsoftwarearchitects.com), Individual Member

**Additional artifacts:**

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

- XML schema: http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/csprd01/schema/ws-calendar-min-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.html
- *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.html

**Declared XML namespaces:**

- http://docs.oasis-open.org/ws-calendar/ns/min-xcal/2015/12

**Abstract:**

The WS-Calendar MIN is a WS-Calendar conformant schema optimized for use in machine-to-machine (M2M) schedule negotiations.

iCalendar (RFC5545) and its peer specification XCAL (also in WS-Calendar 1.0) is a well-known and long used means to convey schedule-related information. iCalendar makes extensive use of extension and recursion. The WS-Calendar Platform Independent Model (PIM) constrains iCalendar and defines a simpler information model which shares iCalendar semantics and can be used to create as the common basis for any number of Platform Specific Models (PSMs).

Because an information model is abstract, it can apply to many transmission and serialization schemas. The PIM itself does not include a transmission and serialization schemas. Through transitive conformance such PSMs themselves conform to WS-Calendar.

The Minimal PIM-Conformant (MIN) schema defines an XML Schema that conforms with the PIM. MIN can be used by itself or as a seed-schema for other specifications.

**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:

**[WS-Calendar-Min-v1.0]**

*WS-Calendar Minimal PIM-Conformant Schema Version 1.0*. Edited by Toby Considine and William T. Cox. 18 December 2015. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/csprd01/ws-calendar-min-v1.0-csprd01.html. Latest version: http://docs.oasis-open.org/ws-calendar/ws-calendar-min/v1.0/ws-calendar-min-v1.0.html.

  

# Notices

# Table of Contents



# Table of Figures



# Table of Tables

# 1 Introduction

[All text is normative unless otherwise labeled]

This specification addresses the need for a tightly conformable seed specification for use of **[WS-Calendar]**-compatible in rapid-processing and light-weight environments. This specifications conforms with the WS-Calendar Platform Independent Model **[WS-Calendar PIM]** and thereby transitively conforms with **[WS-Calendar]**.

iCalendar (RFC5545) and its peer specification XCAL (also in WS-Calendar 1.0) is a well-known and long used means to convey schedule-related information. iCalendar makes extensive use of extension and recursion. The WS-Calendar Platform Independent Model (PIM) constrains iCalendar and defines a simpler information model which shares iCalendar semantics and can be used to create as the common basis for any number of Platform Specific Models (PSMs).

A key concern for the original **[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 defining only specific extensions and limiting recursion. This approach redefined WS-Calendar as what Model Driven Architecture calls a Platform Specific Model (PSM) which conforms to **[WS-Calendar PIM]**

**[WS-Calendar PIM]** is a general specification and makes no assumptions about how its information model is used. **[WS-Calendar PIM]** has specific rules which define Inheritance as a means to reduce the conveyance of repetitive information. As this specification anticipates 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 within each interval.

The **[WS-Calendar PIM]** itself does not include a transmission and serialization schemas, i.e. it is an information model that does not define a normative structure for the information conveyed. Because an information model is abstract, it can apply to many transmission and serialization schemas.

High speed transaction processing requires more predictable means to convey structured information concerning time-based events, states, and transactions. Even valid and conformant conveyances of **[WS-Calendar]** information may fail to meet the requirements for basic interoperability requirements **[WSI-Basic].**

This specification defines a normative structure for conveying time series of information that is conformant with **[WS-Calendar PIM]**. It is the intent of the TC meet the requirements of **[WSI-Basic]**. The Minimal PIM-Conformant **[MIN]** schecification defines an XML Schema that conforms just with the PIM. **[MIN]** can be used by itself or as a seed-schema for other specifications,

## 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]** Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

**[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.

**[vAvailability]** **C. Daboo, M. Douglas: *Calendar Availability*,** https://tools.ietf.org/html/draft-ietf-calext-availability-01 Internet Draft, November 2015.

**[WS-Calendar PIM]** *WS-Calendar Platform Independent Model (PIM) Version 1.0*. Edited by William Cox and Toby Considine. 21 August 2015. OASIS Committee Specification 02. http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/cs02/ws-calendar-pim-v1.0-cs02.html. Latest version: http://docs.oasis-open.org/ws-calendar/ws-calendar-pim/v1.0/ws-calendar-pim-v1.0.html

**[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.

## 1.3 Non-Normative References

**[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

**[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 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

**[xCal]** C. Daboo, M Douglass, S Lees xCal: The XML format for iCalendar, http://tools.ietf.org/html/rfc6321, IETF Proposed Standard, April 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/ min-xcal/2015/12
```

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 |

The normative schemas for WS-Calendar MIN 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="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="ComponentType">
```

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.

## 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 Specification Based on WS-Calendar PIM

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).

Because this specification is A PSM conformant with **[WS-Calendar PIM]**, it transitively conforms to **[WS-Calendar**].

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

## 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”.

An application or specification MAY choose to specify the Duration and the End of an event, if this is simpler for its domain. Such a specification MUST make this expectation clear, as allowing a mix of Start and End based requests makes programming and conformance more difficult. For simplicity, in this document, all scheduling is described refining an Interval with a Duration and adding a Start.

A service request MAY specify both. For example, a Sequence may be advertised with no fixed duration, and a service request MAY specify both the Duration and the Start.

The use of the Start and the End without a definition is discouraged because it reduces flexibility while increasing required computation.

The complete normative discussion of these issues can be found in **[WS-Calendar PIM]**.

### 2.1.1 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].**

  

# 3 Core Components: Intervals, Sequences, and Gluons

The types in this section are each defined in **[WS-Calendar PIM]**. As the PIM is an information model rather than a message format, they are restated here and in the associated schema.

## 3.1 Intervals

The Interval is the core artifact of calendar and schedule. It conveys when something happens and for how long.

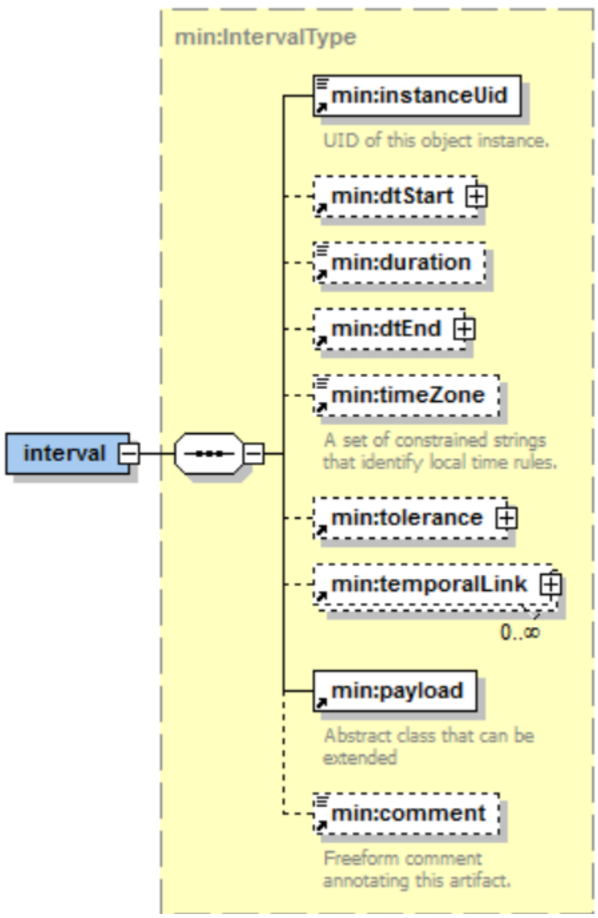


*Figure 3-1 The Interval*

Everything is calendar related except for the payload. The payload is an abstract type to be extended by specifications using this specification. Specifications incorporating this specification Shall define how inheritance applies to the Payload

## 3.2 Temporal Links and Sequences

Temporal Links convey the relations between Intervals in a Sequence.

Each Interval can be considered as a distinct activity for a period of time. A Sequence is a set of such activities. These activities may follow one after another. There may be mandatory gaps, as in paint drying for at least six hours before the next step. It may be a requirement that two Intervals finish at the same time.

If a Sequence describes a ramp-time of activities prior to the Inherited dtStart, then the ramp activities must complete prior to the start time. Similarly, a system MAY need to ramp down at the end of a requested Duration of activity.

There is a special case of Sequence in which all Intervals proceed linearly without pause, and all Intervals share a common Duration. A Sequence of this Type is referred to as a Partition.

### 3.2.1 Temporal Links

Temporal Links are so named because they convey how Intervals are related in Time. A Temporal Link consists of a reference to an Interval, a type of Temporal Relationship, and the Duration of the Relationship.

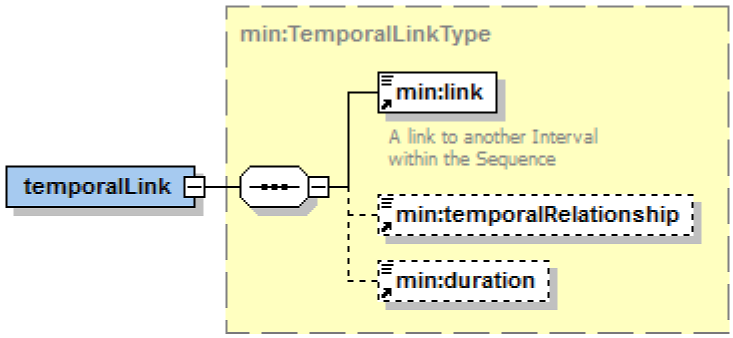


*Figure 3-2: The Temporal Link*

As defined in the PIM, there are four types of Temporal Relationship. Temporal Relationships combine with the Duration to describe a sequence; a Sequence is a set of temporally linked Intervals. A missing or empty Duration is considered a zero length Duration.

If a specification that claims conformance this specification permits a missing Temporal Relationship, then that specification MUST state which Temporal Relationship is implied. A conforming specification MAY disallow a missing Temporal Relationship.

### 3.2.2 Sequences

Sequences are collections of Intervals connected by Temporal Relationships. There is no Sequence structure per-se. A Sequence is referenced by referencing the InstanceUID of one Interval in the Sequence. That Interval is referred to as the Designated Interval. The Designated Interval has special rules for Inheritance. For example, when a Gluon Bequeaths a dtStart to a Sequence, is it the Designated Interval that starts at that time.

Inheritance within a Sequence is specified in **[WS-Calendar PIM]**.

## 3.3 The Gluon

The Gluon links a Sequence to a service interaction. The Gluon can be considered a degenerate Interval that cannot itself be executed. It does, however provide missing information to fully bind each Interval in the Sequence.

Another perspective describes the Gluon as the service entry point for an activity defined by a Sequence. Sequence execution is launched by providing a dtStart though a Gluon. A service request acting as a Gluon bequeaths missing information that is inherited by the entry point Gluon to bind the Sequence.

The Gluon Type is shown in *Figure 3-3: The Gluon.*

  

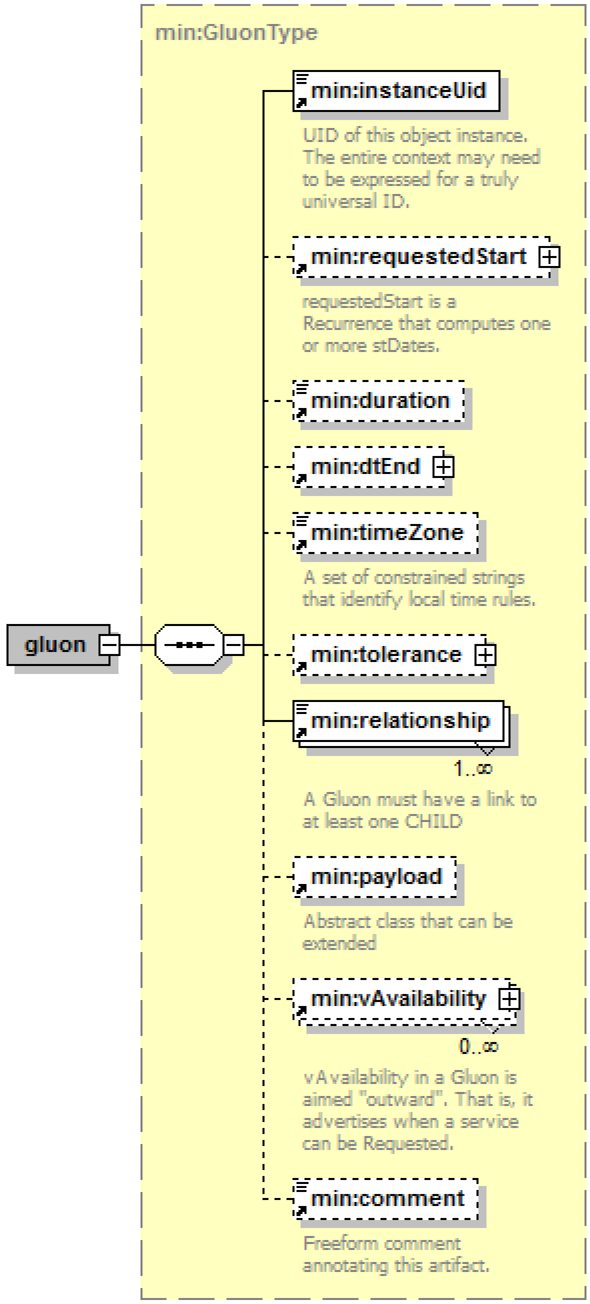


*Figure 3-3: The Gluon*

Notice that the Gluon is nearly identical to the Interval. A Requested Start replaces the dtStart. Requested Start is of type Recurrence. Recurrence describes how to compute a collection of dtStarts. Recurrence is discussed in below in Section 4.

The significant difference between Gluon and Interval are as follows:

1) The Gluon has no Temporal Links. It cannot be part of a Sequence, so it maintains no Temporal Relations with other Components.
2) A Gluon must have at least one Relationship, and it can have many. The Relationship connects a Gluon to a Sequence, to establish Inheritance. A Relationship MAY connect a Gluon to another Gluon, establishing a Lineage that eventually binds a Sequence.

  

3) A Gluon may convey multiple dtStarts. This collection is computed in RequestedStart, which is of type Recurrence. A recurrence is a structure to convey or compute a collection of starting dates and times. These act as if there were multiple Gluons, each conveying a single dtStart.
4) vAvaialbility. VAvailability is an outward looking element that conveys information about potential schedules for the underlying Sequence.

  

# 4 Service Advertising and Request: Recurrence and Availability

Up until this section, dates and times were specific. This section introduces Recurrence and types that enable patterns of dates and schedules to be computed. When a specific term is not defined within this specification, it is as defined in **[WS-Calendar PIM]**.

There may be good reasons for a specification that claims conformance with this specification to forbid e Recurrence. Requiring each service invocation to require its own message that acts as a Gluon MAY simplify the system. A conforming specification MUST state of the use of these components is forbidden.

## 4.1 Recurrence Rules

Recurrence Rules are used in both Recurrence and in Availability to compute patterns of schedules and dates. Each Rule consists of a Rule Part, which names a type of Rule, and Rule Values, constrained lists which operate within the Rule Part.

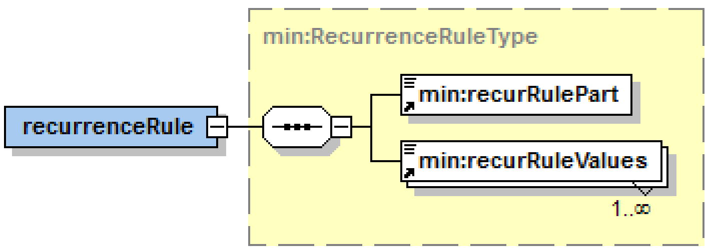


*Figure 4-1: The Recurrence Rule*

Representative recurRuleParts indicate that a Rule is hourly, or at a fixed frequency, or on certain days of the month. Rule Values are constrained depending on the RulePart, to indicate days of the week, every three hours, and so on. Recurrence Rules are normatively described in **[RFC5545]** section 3.3.10. Many web-sites and open source libraries discuss these rules; no efforts will be made in this specification to re-state these rules.

## 4.2 Recurrence

Recurrence is conveys a mechanism to compute a collection of starting date-times. At its simplest, it is a dtStart, just as in the Interval. Recurrence Rules then describe how to compute additional starting dates and times using the dtStart as a seed. rDates add additional starting dates to the collection. xDates then block out dates,that is, remove specific date-times from the collection.

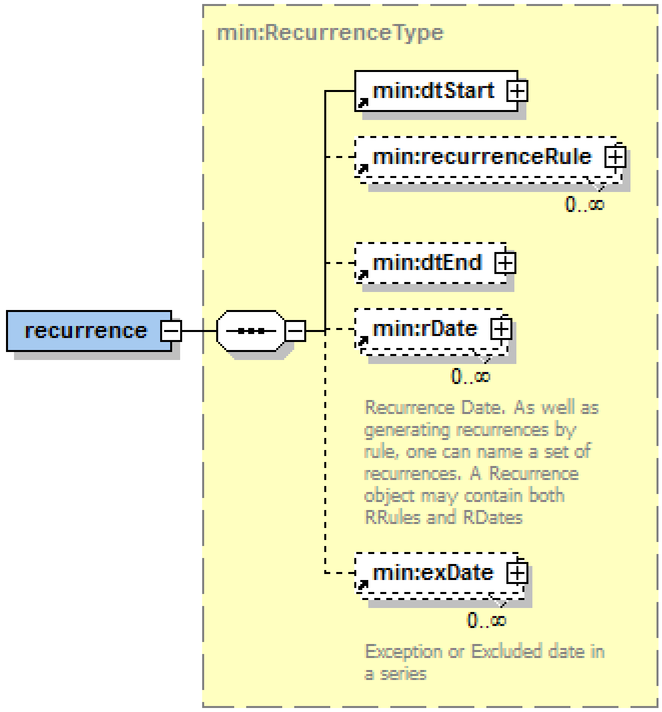


*Figure 4-2: Recurrence*

The Requested Start in the Gluon is of type Recurrence.

## 4.3 Availability and VAvailability

VAvailability is the sum of one or more patterns (Availability) that together express when a Service can be invoked.

As a non-normative illustration, the well-known pattern of “During Business Hours” can be described as the hours from 9:00 AM to 5:00 PM repeated weekly on Monday, Tuesday, Wednesday, Thursday, and Friday. Alternately is might be the sum of two patterns, 8:00 AM until noon, Monday, Tuesday, Wednesday, Thursday, and Friday and 1:00 until 5:00 on Monday, Tuesday, Wednesday, Thursday, and Friday. An additional pattern of 9:00 AM until 1:00 PM might be added each Saturday. The smaller patterns are named “Availability” and the top level summary is named VAvailability.

Note that a Gluon may have an array of Vavailability components. These components MAY be both Available and Unavailable in the same set. There are specific rules for overlaying vAvailability components which the practitioner should be aware of. These rules are described in **[vAvailability]**.

  

### 4.3.1 Availability

The Availability type uses the same computational rules as Reccurrence and applies then to a seed Interval, that is a Duration and dtStart. The DateTime and the Duration are known as the Availability Interval.

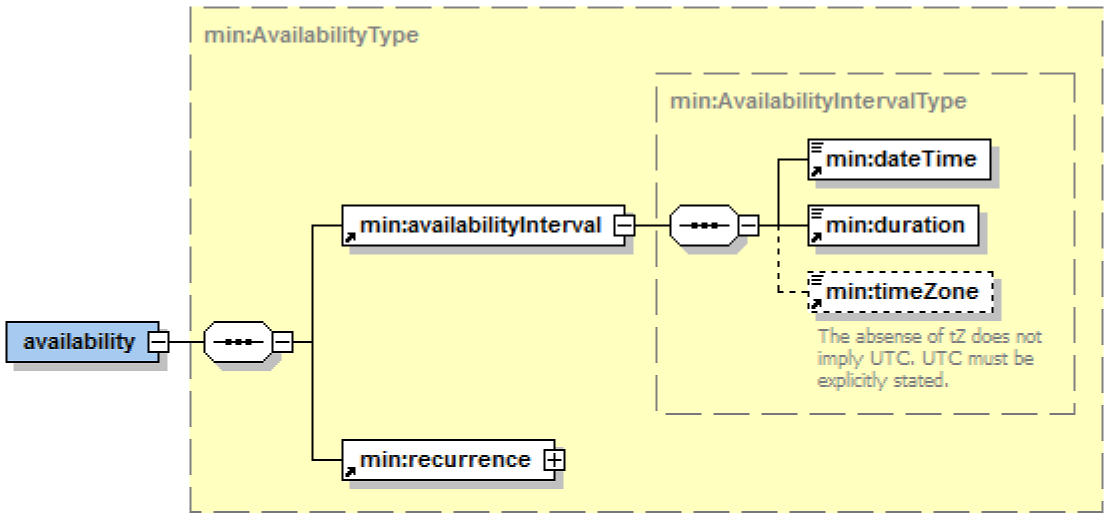


*Figure 4-3: Availability*

Availability applies the Recurrence Rules (RRules) defined in **[RFC5545]** to the availability interval.

## 4.3.2 VAvailability

VAvailability represents the sum of a collection of Availability types applied within the bounds of a defined Time Range.

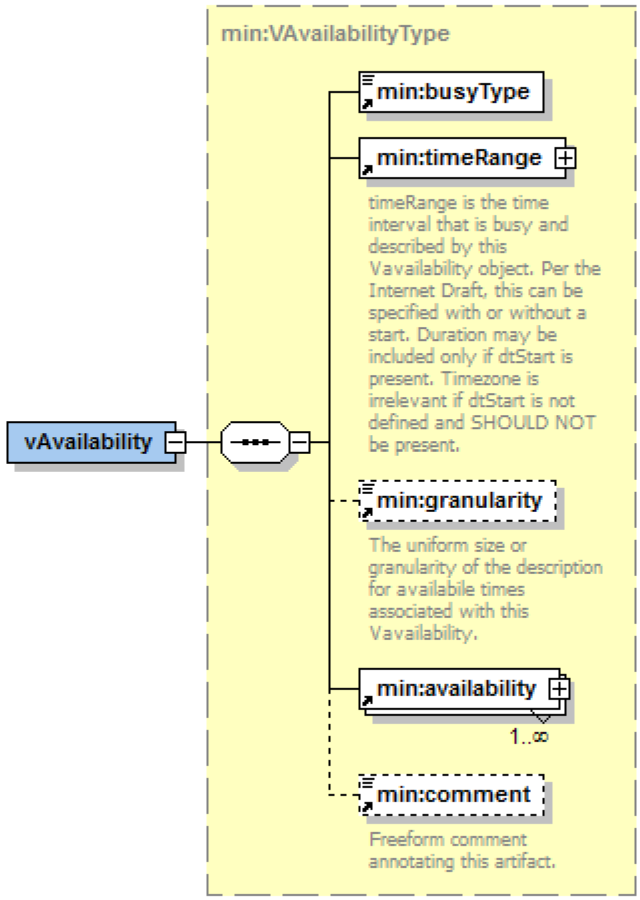


*Figure 4-4: VAvailability Type*

Note that Granularity, when applied to vAvailability has a special meaning. A three hour interval advertised with a granularity of 15 minutes may only be invoked an the 15 minute interval. Forexample, the interval may be 9:00 until Noon, but the only dtStarts that may be requested are at 9:00, 9:15, 8:30, 9:25 and so on.

  

# 5 Conformance

## 5.1 Conformance to WS-Calendar MIN

Implementations and specifications claiming conformance to this specification SHALL implement all inheritance and semantic rules as described in **[WS-Calendar-PIM]** and in particular its Section 5.

Conformance rules in PIM Section 6 are applied to implementations and specifications claiming conformance to MIN.

Implementations and specifications claiming conformance to MIN SHALL implement the entire MIN schema. Extensions are permitted, but MUST be documented in the conforming implementation's conformance statement.

## 5.2 Detailed Conformance with the WS-Calendar-PIM

**[WS-Calendar-PIM]** requires that MIN and other conforming implementations and specification fully support the defined rules in Section 5 "Conformance Rules for WS-Calendar PIM".

**[WS-Calendar-PIM]** Section 6.1 "Conformance for Specifications Claiming Conformance to WS-Calendar PIM" details conformance rules for this specification.

| Section of WS-Calendar-PIM | Notes |
|---|---|
| 6.1 | MIN requires conformance to the referenced rules. |
| 6.2 | Non-normative; the precision is addressed in **[XSD].** Conforming applications should example PIM Section 6.2 and address those issues as deemed appropriate. |
| 6.3 | All conformance requirements in PIM Section 6.3 and its sub-sections meet the requirements of PIM 6.3. |
| 6.4 | These operational conformance requirements are applied to specifications and implementations claiming conformance to MIN in Section 5.1 above. |
| 6.5 | This non-normative section SHOULD be considered by conforming implementations and specifications |

# 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)

  

# Appendix B. Revision History

| Revision | Date | Editor | Changes Made |
|---|---|---|---|
| WD01 | 21 Jul 2015 | Toby Considine | Initial Draft |
| WD02 | 22 Jul 2015 | Toby Considine | Added section on Recurrence and Availability. Added recurrence to Gluons. |
| WD03 | 25 Oct 2015 | Toby Considine | Removed re-statement of PIM, keeping definitions and graphics for simplified models for serialization. |
| WD04 | 10 Dec 2015 | Toby Considine | Changed Relations and Temporal Relations to Relationships and Temporal Relationships. This avoids overloading “Relation” in the PIM. |
| WD05 | 14-Dec-2015 | William T Cox | Added conformance. Minor edits |
| WD06 | 31-Dec-2015 | Toby Considine | Migrated to official template as part of publishing for public review. Slight update of Abstract. Removed some spurious references to MPC. No substantive changes. |