Time Zones and WS-Calendar 1.0

Working Draft 01

22 September 2011

Technical Committee:

OASIS Web Services Calendar (WS-Calendar) TC

Chair:

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

Editors:

Mike Douglass (douglm@rpi.edu), Rensselaer Polytechnic Institute

Related work:

This Work Product is related to:

- OASIS WS-Calendar 1.0 (http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.html)

Abstract:

Summary of the purpose of the document

Introduction to Time Zones, TZID, and Time Zone Servers

Status:

This Working Draft (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or approved as a Committee Note Draft. The OASIS document Approval Process begins officially with a TC vote to approve a WD as a Committee Note Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Note Draft.

This document is intended to become a Non-Standards Track Work Product. The patent provisions of the OASIS IPR Policy do not apply.

Table of Contents

1 Introduction

Timezones, …

1.1 References

[RFC 5545]
Internet Calendaring and Scheduling Core Object Specification (iCalendar). September 2009. IETF RFC. http://tools.ietf.org/pdf/rfc5545

[Timezone-Service]
Timezone Service Protocol-draft 2. 17 August 2011. IETF Draft RFC. https://datatracker.ietf.org/doc/draft-douglass-timezone-service/

1.2 Terminology

Term	Definition
DST	Daylight Saving Time: Modifying the UTC offset for part of the year in order to manipulate the local time with respect to the position of the sun.
DST transition	The point at which local time is adjusted with respect to UTC
Local time	The time displayed on clocks.
Standard time	That part of the year which is not within DST
Timezone	Defines an area of the world which has the same local time and the same rules for transitions into and out of DST
TZID	A name given to a set of rules which determine the UTC offset at a given time and date.
UTC	Coordinated Universal Time a time standard based in International Atomic Time modified by the addition or removal of leap seconds to synchronize it with the earth’s rotation.
UTC Offset	The (possibly negative) number of minutes added to UTC to obtain local time.

2 Timezones

Before the use of timezones it was generally the practice to coordinate time only over very small areas such as towns or cities. Timezones allow us to establish a local or clock time within much larger areas which correspond to some extent to the natural cycle of daily personal or business activities. Currently there are around 85 to 90 of these timezones in active use.

It is often assumed that UTC is the constant and that local time is a derivative. In reality the reverse is the case. As humans we stick to a cycle which is driven by local or clock time. This has economic and practical impacts.

Countries often choose to use a timezone that is not well coordinated with the sun in order to facilitate business transactions. Power usage depends to a large extent on local time.

2.1 Daylight Saving Time (DST)

Without DST timezones would be relatively simple. A given area of the planet would choose a timezone and the offset fromUTC would be constant throughout the year. However, DST introduces a complication in that the offset from UTC depends not only on position but on the date.

Contrary to popular belief, it was not the agricultural community that pushed for the adoption of UTC. In fact UTC was adopted as the result of pressure from large cities, especially those involved in international finance. Governments occasionally choose to alter these DST transitions, often because it is claimed that there will be some financial or safety benefit, and sometimes because of some special event taking place. These irregularities in the DST transitions require a complex set of rules to allow UTC offsets to be calculated for any point in time.

2.2 Timezone specifications.

Timezone specifications are the rules by which we determine what offset must be applied to UTC at specific times in order to determine local time. The [RFC 5545] VTIMEZONE defines a particular form of these rules based on [RFC 5545] recurrence rules and RDATES (a specific date and time for a transition). These VTIMEZONE objects are often, but not exclusively, derived from such data as the Olson database.

Another form of specification can be more like the zoneinfo information used by operating systems and some libraries. This takes the form of an array of pairs, onset + offset, which define the time at which (the onset) a given offset must be applied. This form is relatively easy to use – just find which 'slot' the local time is within and that provides the offset from UTC.

2.3 Timezone identifier (TZID)

The tzid is a name by which we refer to a timezone specification. An example of an Olson name is "America/New_York". The iCalendar specification doesn't say much about the form of tzids other than:

```
This is the label by which a time zone calendar
component is referenced by any iCalendar properties whose value
```



```
type is either DATE-TIME or TIME and not intended to specify a UTC
or a "floating" time.  The presence of the SOLIDUS character as a
prefix, indicates that this "TZID" represents an unique ID in a
globally defined time zone registry (when such registry is
defined).

Note: This document does not define a naming convention for
time zone identifiers.  Implementers may want to use the naming
conventions defined in existing time zone specifications such
as the public-domain TZ database [TZDB].  The specification of
globally unique time zone identifiers is not addressed by this
document and is left for future study.
```

To paraphrase, if it starts with "/" it is a global identifier but no global identifiers have been defined. Most of the zoned date/times passed around are using the Olson set of names. Internally other system are used, in particular, MicroSoft has its own set of timezone rules and names.

2.4 Zoned date/times & UTC

Zoned date/time values are dates and times which specify the timezone in which they occur. UTC is the time based on an atomic standard so would seem the most dependable. However, what matters most of the time is actually the local time in some timezone. If we are trying to set up a meeting, bid for power at some time or set an alarm what is wanted is for that event to take place at a given local time. We may derive the UTC from that with the aid of timezone rules but when those rules change it's the UTC we alter – not the local time – that is, we recalculate the value with the new rules and renegotiate.

2.5 Storing Zoned Times

The assertion is often made that the best practice is to store UTC. Another frequent statement is that timezones are merely a UI issue, that we only need them when displaying times to humans. This practice has led to many unpleasant issues over time and no doubt will continue to do so. UTC is a useful value for indexing and searching but without the local time and tzid to accompany it we have no way of validating that UTC value in the face of changes to the specifications.

When we try to negotiate a meeting, bid for power at some time etc, the UTC value allows us to assure ourselves that we are indeed talking about the same time. Storing that value (or the UTC offset) along with the local time and the tzid on which it is based allow us to revalidate that value when rules change.

2.6 A meeting example.

Imagine two parties, Bill and Gershon who are trying to set up a meeting. After much discussion, they both have full calendars, they agree on a time 10:00 for Bill in the US and 16:00 for Gershon in Brussels which correspond to a UTC time of 14:00.

Before they can have their meeting the US government decides to change the rules again. 10:00 in the Eastern US now corresponds to 15:00 UTC meaning Bill would have to call Gershon at 09:00 locally or Gershon would have to wait for the call till 17:00 in Brussels. However, both have full calendars so they have to abandon the meeting and renegotiate.

Such a scenario seems moderately easy to handle, other than that setting up a meeting takes a lot of work. However, this scenario was repeated many thousands of times in a matter of 1 to 2 weeks with the calendaring system unable to provide any useful assistance to their users. They simply didn't know which meetings or events would have to be renegotiated. This problem affects even simple reminders or public events as moving them can cause conflicts with other events and meetings they may have stored in other systems.

2.7 Recurrences and durations.

Recurring events are much used in calendaring and rely on the use of timezone specifications to work correctly. For example, we may specify "Every Monday at 10:30am". The rule to do this is very simple but it does not translate to a constant UTC time. Rather the UTC time changes as we go through the year. However, the clock time is always the same.

In durations, when we specify a time such as a day or a week the number of hours may vary according to the date. One day may be 23, 24 or 25 hours in some regions.

3 Timezone Specifications

Timezone specifications come mostly from one of two places.

3.1 The Olson database

This is a long standing repository of timezone information used by almost all systems that handle zoned times. This data is collected by a large number of individuals who post to a mailing list. Up to the present this data was checked by 2 people who use it to update the zone database. This is distributed on a regular basis and forms the base for all timezone information for non-Microsoft operating systems, the Java runtime and many other language frameworks and calendar systems.

The Olson data attempts to be an historical record of all timezone information and has around 400 specifications.

The IETF and IANA are in the process of taking over management of the data and the mailing list so as to put it on a more formal footing.

Note that the maintainers and holders of the Olson database are currently the targets of a law suit and the database is currently shut down.

3.2 Microsoft timezones

These are collected in much the same way by a different set of individuals who feed the information back to Microsoft. Unlike Olson, the Microsoft data does not attempt to be a historical record.

3.3 Timezone service

Work is currently underway to design and implement a timezone service which will allow applications, services and systems to fetch timezone information from a number of servers arranged in a hierarchical system. In this system local or secondary servers will populate themselves from primary servers which maintain copies of the data for redistribution.

This specification [Timezone-Service] is currently in draft stage but there are a small number of implementations some already in production. In traditional calendaring the specification MUST be sent with the events. In WS-Calendar and probably in the future for [RFC 5545], the timezone is simply referenced by its TZID and recipients of zoned data will fetch the timezone information for themselves from their local server.

Calendaring services will advertise which timezone server they used.

4 Leap Seconds

Currently leap seconds play no part in timezone calculations. Conversions are based on UTC which is adjusted by the underlying system to take account of leap seconds as needed. At some point, perhaps sometime in 2014 UTC will no longer be adjusted to take accounts of leap seconds.

This is a change requested by a number of parties. What will be the final outcome of the current negotiations is still unknown. One possibility is that UTC will be allowed to drift with relation to the sun.

It is possible a new specification will provide a timezone which effectively provides a leap-second adjusted time.

Appendix A. Acknowledgments

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

Participants:

[Participant Name, Affiliation | Individual Member]
[Participant Name, Affiliation | Individual Member]

Appendix B. Some Section

Appendix C. Revision History

Revision	Date	Editor	Changes Made
[Rev number]	[Rev Date]	[Modified By]	[Summary of Changes]

