

An Introduction to Internet Calendaring and Scheduling

Working Draft 01

23 October 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:

Dave Thewlis (Dave.Thewlis@calconnect.org) ([The Calendaring and Scheduling Consortium](#))

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

An introduction to the primary standards and protocols associated with Internet Calendaring and Scheduling. The contents of this document are copied in whole and with permission from An Introduction to Internet Calendaring and Scheduling V1.1 published 20 October 2011 and copyright by The Calendaring and Scheduling Consortium, licensed under a Creative Commons V3.0 Unported License.

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.

Copyright © OASIS Open 2011. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

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

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

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of Contents

1	Introduction.....	3
1.1	References	3
1.2	Section Level 2	Error! Bookmark not defined.
1.2.1	Section Level 3	Error! Bookmark not defined.
1.3	Section Level 2	Error! Bookmark not defined.
2	Heading.....	Error! Bookmark not defined.
Appendix A.	Acknowledgments.....	7
Appendix B.	Some Section.....	Error! Bookmark not defined.
Appendix C.	Revision History	9

1 Introduction

This document provides an introduction and overview of the major Calendaring and Scheduling standards and protocols. The majority of these standards and protocols are published by the IETF (Internet Engineering Task Force) and developed by an IETF working group or submitted as an Internet Draft by one or more individuals, including material submitted by The Calendaring and Scheduling Consortium.

1.1 References

- [1] <https://datatracker.ietf.org/doc/rfc2445/> : 'Internet Calendaring and Scheduling Core Object Specification'
- [2] <https://datatracker.ietf.org/doc/rfc2446/> : 'iCalendar Transport-Independent Interoperability Protocol'
- [3] <https://datatracker.ietf.org/doc/rfc2447/> : 'iCalendar Message-Based Interoperability Protocol'
- [4] <https://datatracker.ietf.org/doc/rfc5545/> : 'Internet Calendaring and Scheduling Core Object Specification'
- [5] <https://datatracker.ietf.org/doc/rfc5546/> : 'iCalendar Transport-Independent Interoperability Protocol'
- [6] <https://datatracker.ietf.org/doc/rfc4791/> : 'Calendaring Extensions to WebDAV'
- [7] <https://datatracker.ietf.org/doc/rfc6321/> : 'xCal: The XML format for iCalendar'
- [8] <http://msdn.microsoft.com/en-us/library/aa913903.aspx> : ActiveSync
- [9] <http://www.openmobilealliance.org/tech/affiliates/syncml/syncmlindex.html> : SyncML
- [10] <http://www.calconnect.org/pubdocs/CD1011%20CalWS-Rest%20Restful%20Web%20Services%20Protocol%20for%20Calendaring.pdf> : 'Restful Web Services Protocol for Calendaring'
- [11] <http://tools.ietf.org/id/draft-desruisseaux-ischedule-01.txt> : 'Internet Calendar Scheduling Protocol'
- [12] <https://datatracker.ietf.org/doc/rfc6047/> : 'iCalendar Message-Based Interoperability Protocol'
- [13] <http://docs.oasis-open.org/ws-calendar/ws-calendar-spec/v1.0/cs01/ws-calendar-spec-v1.0-cs01.html> : 'OASIS WS-Calendar 1.0'

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

[14] <http://msdn.microsoft.com/en-us/library/dd877045%28v=exchg.140%29.aspx> : 'Exchange Web Services'

2 iCALENDAR

2.1 History

The iCalendar specification was first produced by the IETF in 1998 as RFC 2445 [1]. Since then it has become the dominant standard for calendar data interchange on the internet and between devices (desktop computers, mobile phones etc). The specification was revised in 2009 as RFC 5545 [4].

Alongside iCalendar is the iTIP specification (RFC 2446 [2] and revised as RFC 5546 [5]) that defines how iCalendar is used to carry out scheduling operations (for example, how an organizer can invite attendees to a meeting and receive their replies). This forms the basis for email-based scheduling using iMIP (the specification that describes how to use iTIP with email) (RFC 2447 [3]) and revised as RFC 6047 [12]). iCalendar itself is a text-based data format. However, an XML format is also available, providing a one-to-one mapping to the text format (RFC 6321 [7]).

iCalendar data files typically have a .ics file name extension. Most desktop calendar clients can import or export iCalendar data, or directly access such data over the Internet using a variety of protocols.

2.2 Data Model

The iCalendar data format has a well defined data model. "iCalendar objects" encompass a set of "iCalendar components" each of which contains a set of "iCalendar properties" and possibly other sub-components. An iCalendar property consists of a name, a set of optional parameters (specified as "key-value" pairs) and a value.

iCalendar components include:

- "VEVENT" which represents an event
- "VTODO" which represents a task or to-do
- "VJOURNAL" which represents a journal entry
- "VFREEBUSY" which represents periods of free or busy time information
- "VTIMEZONE" which represents a timezone definition (timezone offset and daylight saving rules)
- "STANDARD" and "DAYLIGHT" are sub-components of "VTIMEZONE" and provide the onset and offset information
- "VALARM" is currently the only defined sub-component for events and tasks and is used to set alarms or reminders

Properties include:

- "DTSTART" which represents a start time for a component
- "DTEND" which represents an end time for a component
- "SUMMARY" which represents a title or summary for a component
- "RRULE" which can specify rules for repeating events or tasks (for example, every day, every week on Tuesdays, etc.)

- "ORGANIZER" which represents the calendar user who is organizing an event or assigning a task
- "ATTENDEE" which represents calendar users attending an event or assigned a task

In addition to this data model and the pre-defined properties, the specification defines how all those are used together to define the semantics of calendar objects and scheduling. The semantics are basically a set of rules stating how all the components and properties are used together to ensure that all iCalendar products can work together to achieve good interoperability. For example, a rule requires that all events must have one and only one "DTSTART" property. The most important part of the iCalendar specification is the semantics of the calendaring model that it represents. The use of text or XML to encode those is secondary.

2.2 Scheduling

The iTIP specification defines how iCalendar objects are exchanged in order to accomplish the key task needed to schedule events or tasks. An example of a simple workflow is as follows:

- To schedule an event, an organizer creates the iCalendar object representing the event and adds calendar users as attendees. The organizer then sends an iTIP "REQUEST" message to all the attendees.
- Upon receipt of the scheduling message, each attendee can decide whether they want to attend the meeting or not.
- Each attendee can then respond back to the organizer using an iTIP "REPLY" message indicating their own attendance status.

iTIP supports other types of scheduling messages, for example, to cancel meetings, add new instances to a repeating meeting, etc.

2.3 Extensibility

iCalendar was designed to be extensible, allowing for new components, properties and parameters to be defined as needed. A registry exists to maintain the list of standard extensions with references to their definitions to ensure anyone can use them and work well with others.

2.4 WS-Calendar

WS-Calendar [13], jointly developed by The Calendaring and Scheduling Consortium and OASIS, is one such extension to iCalendar. It is intended to provide a data model sufficient for the needs of the NIST Smart Grid Interoperability Standards. Many of these extensions to iCalendar also appear to be of use in project management, financial management, and other areas focused primarily on scheduling per se.

3 CALENDAR DATA ACCESS AND EXCHANGE PROTOCOLS

3.1 Internet Calendar Downloads

Calendar data can be downloaded from a web server via a web browser and then imported directly into an iCalendar aware client. Alternatively, calendar data could be retrieved in some other fashion from the Internet (e.g., sent via email as an attachment) and then imported into a calendar client.

This solution works well for calendar data that is not likely to change over time (for example the list of national holidays for the next year).

3.2 Internet Calendar Subscriptions

An Internet calendar subscription is simply an iCalendar data file made available on a web server. However, rather than using a web browser to download the calendar data, calendar clients can instead use the URL to the calendar data on the web server to download the calendar data themselves. Additionally, the clients can check the web server on a regular basis for updates to the calendar data, and then update their own cached copy of it. This allows calendar data that changes over time to be kept synchronized.

3.3 CalDAV

CalDAV is a calendar access protocol and is defined in RFC 4791 [6]. The protocol is based on WebDAV which is an extension to HTTP that provides enhanced capabilities for document management on web servers.

CalDAV is used in a variety of different environments, ranging from very large internet service providers, to large and small corporations or institutions, and to small businesses and individuals.

CalDAV clients include desktop applications, mobile devices and browser-based solutions. It can also be used by "applets", for example, a web page panel that displays a user's upcoming events.

One of the key aspects of CalDAV is its data model. Simply put, it defines a "calendar home" for each calendar user, within which any number of "calendars" can be created. Each "calendar" can contain any number of iCalendar objects representing individual events, tasks or journal entries. This data model ensures that clients and servers can interoperate well.

In addition to providing simple operations to read, write and delete calendar data, CalDAV provides a querying mechanism to allow clients to fetch calendar data matching specific criteria. This is commonly used by clients to do "time-range" queries, i.e., find the set of events that occur within a given start/end time period.

CalDAV also supports access control allowing for features such as delegated calendars and calendar sharing.

CalDAV also specifies how scheduling operations can be done using the protocol, rather than, for example, sending scheduling messages via email. Whilst it uses the semantics of the iTIP protocol, it simplifies the process by allowing simple calendar data write operations to trigger the sending of scheduling messages, and it has the server automatically process the receipt of scheduling messages. Scheduling can be done with other users on the CalDAV server or with calendar users on other systems (via some form of "gateway").

3.4 CalWS-REST

CalWS-REST[10] is a RESTful web services calendar access application programming interface developed by The Calendaring and Scheduling Consortium and the OASIS organization, to be used as part of the OASIS WS-Calendar standard. It provides a programming interface to access and manipulate calendar data stored on a server that is more suited to a web services programming environment. It follows a similar data model to CalDAV and has been designed to co-exist with a CalDAV service offering the same data.

3.5 CalWS-SOAP

CalWS-SOAP (under development) is a SOAP web services calendar access application programming interface still under development by The Calendaring and Scheduling Consortium and OASIS and is also to be used as part of the OASIS WS-Calendar standard. As with CalWS-REST, it follows a similar model to CalDAV and provides slightly more features than the current CalWS-REST protocol.

3.6 iSchedule

iSchedule [11] is a protocol to allow scheduling between users on different calendaring systems and across different internet domains. It transports iTIP scheduling messages using HTTP between servers. Servers use DNS and various security mechanisms to determine the authenticity of messages received.

It has been specifically designed to be independent of any calendar system in use at the endpoints, so that it is compatible with many different systems. This allows organizations with different calendar systems to exchange scheduling messages with each other, and also allows a single organization with multiple calendar systems (for example due to mergers, or different departmental requirements) to exchange scheduling messages between users of each system.

3.7 ActiveSync/SyncML

ActiveSync [8] and SyncML [9] are technologies that allow multiple devices to synchronize data with a server, with calendar data being one of the classes of data supported. These have typically been used for mobile devices with a broad spectrum of capabilities, but most often in "smart" phones.

3.8 Exchange Web Services (EWS)

Exchange Web Services [14] is another technology that allow multiple devices to synchronize data with an Exchange server, with calendar data being one of the classes of data supported. It is a SOAP based protocol introduced with Exchange 2007 which manages a proprietary data format.

4 Acknowledgments

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

4.1 Participants:

[Cyrus Daboo, The Calendaring and Scheduling Consortium]
[Mike Douglass, The Calendaring and Scheduling Consortium, Rensselaer Polytechnic Institute]

5 Revision History

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