

Calendar Interoperability Test Report, September 26-27, 2006

Published Administrative

Warning for drafts

This document is not a CalConnect Standard. It is distributed for review and comment, and is subject to change without notice and may not be referred to as a Standard. Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© 2006 The Calendaring and Scheduling Consortium, Inc.

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from the address below.

The Calendaring and Scheduling Consortium, Inc.

4390 Chaffin Lane
McKinleyville
California 95519
United States of America

copyright@calconnect.org
www.calconnect.org

Contents

Foreword.....	iv
Introduction.....	v
Attendees.....	v
1. General Comments.....	1
1.1. First set of comments.....	1
1.2. Second set of comments.....	2
1.3. Third set of comments.....	2
1.4. Fourth Set of Comments.....	2
1.5. Fifth Set of Comments.....	3
1.6. Sixth Set of Comments.....	3
1.7. Seventh set of Comments.....	3
2. Summary.....	3
Appendix A CalDAV Testing Matrix.....	5

Foreword

This document incorporates by reference the CalConnect Intellectual Property Rights, Appropriate Usage, Trademarks and Disclaimer of Warranty for External (Public) Documents as located at

<http://www.calconnect.org/documents/disclaimerpublic.pdf>.

Chair(s)

Pat Egen

Introduction

This document contains notes and results from the September 2006 calendar interoperability event held at the Apple complex in Cupertino, CA. The basic purpose of the event was to start testing Free Busy and Scheduling in CALDAV. In addition, there was continued testing of iCalendar events by the Eventful organization.

The chart below shows the attendees, their organization and the products they were testing.

Attendees

Attendees	Organization	Products
Chuck Norris	EVDB	Eventful server
Jeffrey Harris	OSAF	Chandler client
Grant Baile	OSAF	Chandler client
Mikael Rogers	OSAF	Chandler client
Brian Mosely	OSAF	Cosmos server
Mike Douglass	RPI	Bedework server
Cyrus Daboo	Apple	Apple server
Wilfredo Sanchez	Apple	Apple server
Red Dutta	Apple	Apple client — ical Version 3.0 (1118)
Scott Adler	Apple	Apple client
Haley Allen	Apple	Apple client
Dan Mosedale	Mozilla	Sunbird / Lightning 0.3 Release Candidate
Matt Willis	Mozilla	Sunbird / Lightning 0.3 Release Candidate

Calendar Interoperability Test Report, September 26-27, 2006

1. General Comments

The following applications and products were tested:

- Four CALDAV servers — RPI, Oracle, Apple and OSAF (Cosmo)
- Three CALDAV clients — OSAF (Chandler), Mozilla, and Apple
- Absorbing Recurrence rules (RFC 2445 and RFC 2447) by Eventful.

Free Busy and Scheduling are recent additions to CALDAV. However, we had three clients testing CALDAV against four CALDAV servers this event. This is a three-fold increase over the last interop testing. Since scheduling is the most recent addition to CALDAV there was limited testing of that part of the specification.

The following paragraphs discuss items tested during the interop. The actual matrix showing the specific test cases is provided at the end of this document.

The first part of the testing covered Event creation and modification. Two of the vendors were able to pass and test all items against two servers. The other vendors were able to pass several, but not all of the items in the testing suite.

The second part of the testing covered Event Modification. Again, two vendors supported and passed testing of the nine items against two servers. One vendor did not participate in this test.

The third part of the testing covered Event Retrieval via CalDAV reports, which, at present no one supports and therefore was not tested.

The fourth part covered Event Deletion. There were five items to test and two vendors supported and passed testing against two servers.

The fifth part covered Access Control and no one either supports or tested any of the four test items.

The sixth part covered Calendar Management. Two vendors do not have support for this and another vendor only supports two of the six test items.

The seventh part of the test was Free Busy Reports. Only one vendor supported and tested this section. They tested and passed eleven items against one of the testing servers.

The eighth and final section tested Scheduling. There were 7 test cases and one vendor supported and passed four of the items against one server only. No other clients tested this section.

Following now are comments submitted by the test participants.

1.1. First set of comments

We found the interoperability lab to be a very valuable exercise, and during the course of the two days were able to make changes in our current application. At the end of the two days we were communicating with all four servers, reading data, making changes with three of them. Most of the changes were based on incorrect assumptions, caused by developing with only one server. Moving forward, having access to multiple implementations will let us build a much more robust client.

As for the event itself, having face to face contact, and immediate bug fixing on both sides was extremely valuable to a quick development process. Doing this remotely might work, but the fact

that we can bring laptops right to the people writing the server, or ask the other clients to modify data made the whole event go smoothly.

Some issues we encountered with specifically:

- Their application depends on valid principals with calendar- home-set properties; not all the servers had this set up.
- Calendar User Addresses are an issue, the fact that they can be anything, even email addresses that they are using elsewhere in their application was causing them problems with scheduling.
- They were using relative paths in some cases where servers were expecting full paths
- Also a couple of other minor issues, not really worth getting into here.

We realize this interoperability meeting was focused on client/server compatibility. One thing we would like to see in the future is more multi-client read write access to the same data on the same server and then making sure that everything still works. For example, have one vendor create a lot of calendar data on their server with their client, then let our application read and modify the data, then see that it shows up correctly the receiving application.

“Overall this was a great event, and we look forward to the next one.”

1.2. Second set of comments

Our server team was pleased to participate in this CalConnect IOP. We had the absolute latest version of our CalDAV server available for testing on two machines.

All clients were tested against the server during the course of the event. No major problems were found in our server implementation. On the whole other clients do not yet support some of the more ‘advanced’ features such as access control and scheduling, so significant portions of the code were not exercised, but those portions that were performed as expected. We are not aware of any significant problems clients had with working with our server as compared to others.

Overall we found this to be a valuable experience and look forward to participating again in the future.

1.3. Third set of comments

- 1) One vendor requires a principal resource with the CALDAV:calendar-home-set property so that it can locate a user’s calendar collections. The developer added support to their application for WEBDAV principals (section 4 of the ACL spec).
One application had problems renaming calendars. This operation uses WEBDAV MOVE underneath. It turns out that the application was sending a relative URI in the Destination header which RFC 2518 specifies must contain an absolute URI. The developer beefed up the code to handle both forms and I believe the folks changed to absolute, so all was well after that.
- 2) One vendor did some amount of testing with their application, but the developer wasn’t directly involved and they never communicated any issues to him.
- 3) Another vendor ran the CALDAV test suite from their calendar server and found an issue when DELETEing a null resource.
- 4) An unknown party exposed a bug wherein a vendor application errored when receiving any CALDAV REPORT against a regular (non-calendar) collection.

The developer fixed all of these issues yesterday and then waited to hear about anything else that came up the next day. Nothing was reported.

1.4. Fourth Set of Comments

One vendors application converts individual to UTC time. They also move floating recurr to UTC. Syncing unlimited recurrence fails. It also adds Alarms to every event. We found that this application rewrites exceptions as RDates and Exdates. Another vendor’s handling of modifications

loses non-time pieces of these exceptions. All-day event creation fails on another vendors application. We also noted that another vendor does not send durations.

With regards to the specific items tested, they noted the following:

- 1) While verifying two free busy periods, they actually got three free-busy periods noted. So far as they could tell, both forms are valid (i.e. servers & clients are free to coalesce free-busy as they see fit).
- 2) While testing against one vendor application they noted the following:
 - a) The server deleted VALARMS stored with events
 - b) Modifications to recurring events caused an internal server error (500 HTTP response)
 - c) The server never changes ETags, so a second client isn't able to sync properly, since it thinks events haven't changed since previous sync.
 - d) The HTTP DELETE request succeeded, but the collection stayed on the server.
 - e) The server returned an empty (HTTP 204) response to a valid free-busy-request REPORT.

1.5. Fifth Set of Comments

They were very happy to exchange iCalendar objects to send to their server. Several attendees sent objects to help them test their iCalendar support, in particular recurring events. They found issues when absorbing iCalendar recurrence events. These will be useful in helping them streamline their software.

1.6. Sixth Set of Comments

They tested their products against both servers. They also tested against another application. Not having stop time or no DTStart on timed event did cause some problems. They may need to put some kind of Note on events with no stop time or something like a ragged edge Icon on items with no end times. However, this may break scheduling with no end time. They also noted that the organizer object in one vendor's application is an issue. Another vendor doesn't allow the creator to update the principle.

1.7. Seventh set of Comments

We assume testers will have found some problems in the area of recurrences when working with our product. They are currently working on rewriting that support.

One vendor ran across a problem with a recurrent event that proved to be an ical4j bug. This is now fixed.

Another vendor's application etags were broken and are now fixed.

We partially fixed a problem deleting calendars.

2. Summary

As usual, the interoperability testing revealed problems with servers that no one knew about. These were resolved quickly in many cases or will be resolved when the attendees get back to their respective facilities. It is always better to test something before it goes production and that is one of the things we can provide — a safe, non-public forum and environment for testing software interoperability.

Issues that came out of the testing included principle support, attachments and calendar discovery. There needs to be something defined to better identify and discover calendar servers. In addition, there needs to be a better mechanism for handling attachments, including what kind of controls should be put in place. It was also noted that it would be good to have more report examples to help with development and testing.

CC/A 0612:2006

Following this document is a matrix of the CALDAV test matrix showing what items passed, failed or were not tested or supported.

Respectfully submitted, Pat Egen. Interoperability Event Manager

Appendix A (normative) CalDAV Testing Matrix

Table A.1 — 6th CalDAV Interop Testing Event — September 2006

	Server 1			Server 2			Server 3			Server 4		
	A	B	C	A	B	C	A	B	C	A	B	C
1. Event creation.												
1.1. Create new single-instance meeting titled "Meeting 1.1" with the location "Durham".	P	P	P	P	P		P	P		N	P	
1.2. Create new meeting titled "Meeting 1.2" recurring every Monday from 10:00 AM to 11:00 AM for 4 weeks	P	P	P	P	P		P	P		N	P	
1.3. Create new single-instance meeting titled "Meeting 1.3" with 2 other attendees.	P	P	N	P	P		P	N		N	N	
1.4. Create new single-instance meeting titled "Meeting 1.4" with an alarm set to trigger 15 minutes prior to the schedule time of the meeting	P	P	P	P	P		P	F		N	P	
2. Event modification												
2.1. Modify the title of meeting "Meeting 1.1" to "Meeting 1.1bis".	P	P	P	P	P		P			N	P	
2.2. Modify the location of the meeting "Meeting 1.1bis" to "Seattle bis".	P	P	P	P	P		P			N	P	
2.3. Reschedule meeting "Meeting 1.1bis" to the next day.	P	P	P	P	P		P			N	P	
2.4. Add an attendee to "Meeting 1.1bis".	P	P	N	P	P		N			N	N	
2.5. Add an alarm to "Meeting 1.1bis".	P	P	P	P	P		F			N	P	
2.6. Modify the title of the 1st instance of the recurring meeting created in 1.2.	P	P	P	P	P		F			N	F	
2.7. Modify the participation status of the 1st attendee in meeting 1.3 to DECLINED.	P	P	N	P	N		P			N	N	
2.8. Cancel the 4th instance of the recurring meeting created in 1.2.	P	P	P	P	P		V			N	N	
2.9. One client changes "Meeting 1.1bis" to a different time, second client 'refreshes' its display to see the modification.	P	P	P	P	P		F			N	P	
3. Event retrieval												
3.1. calendar-query REPORT	N	N		N			N			N		
3.1.1. No filtering (match everything)	N	N		N			N			N		
3.1.1.1. Query all components and return all data. (tests <calendar-query> and <filter>)	N	N		N			N			N		
3.1.1.2. Query all components and return ETag WebDAV property and all data. (tests <calendar-query>+<DAV:prop> and <filter>)	N	N		N			N			N		
3.1.1.3. Query all components and return just entire VEVENT components. (tests <calendar-query>, <filter> +<comp-filter>)	N	N		N			N			N		
3.1.1.4. Query all components and return VEVENT components with only DTSTART, DTEND/DURATION, SUMMARY, UID, SEQUENCE, RRULE, RDATE, EXRULE, EXDATE, RECURRENCE-ID. (tests <calendar-query>, <filter>+<comp-filter>, <calendar-data><comp><prop>)	N	N		N			N			N		
3.1.2. time-range filtering	N	N		N			N			N		
3.1.2.1. Query all components within a one day time-range and return all data. Make sure that there is a recurring event that starts prior to the chosen time-range but has one	N	N		N			N			N		

	Server 1			Server 2			Server 3			Server 4		
	A	B	C	A	B	C	A	B	C	A	B	C
non-overridden instance within the time-range. (tests <calendar-query>, <filter>+<time-range>)												
3.1.2.2. Query all components within a one week time-range and return just entire VEVENT components. Make sure that there is a recurring event that starts prior to the chosen time-range but has one overridden instance within the time-range. (tests <calendar-query>, <filter>+<time-range>)	N	N		N					N		N	
3.1.3. component based filtering	N	N		N					N		N	
3.1.3.1. Query all components that contain an embedded VALARM component. (tests <calendar-query>, <filter>+<comp-filter>)	N	N		N					N		N	
3.1.3.2. Query all components that contain an embedded VALARM component whose trigger falls within a specific time-range. (tests <calendar-query>, <filter><comp-filter><prop-filter>+<time-range>)	N	N		N					N		N	
3.1.4. property based filtering	N	N		N					N		N	
3.1.4.1. Query all components that contain any ORGANIZER property. (tests <calendar-query>, <filter><prop-filter><is-defined>)	N	N		N					N		N	
3.1.4.2. Query all components that contain an ORGANIZER property with a specific CUA text value case-insensitively. (tests <calendar-query>, <filter><prop-filter><text-match>+<caseless>)	N	N		N					N		N	
3.1.4.3. Query all components that contain an ORGANIZER property with a specific CUA text value case-sensitively. (tests <calendar-query>, <filter><prop-filter><text-match>+<caseless>)	N	N		N					N		N	
3.1.5. parameter based filtering	N	N		N					N		N	
3.1.5.1. Query all components that contain a DTSTART property with a TZID parameter. (tests <calendar-query>, <filter><prop-filter><text-match><param-filter><is-defined>)	N	N		N					N		N	
3.1.5.2. Query all components that contain an ATTENDEE property with PARTSTAT=NEEDS-ACTION parameter. (tests <calendar-query>, <filter><prop-filter><text-match><param-filter><text-match>)	N	N		N					N		N	
3.2. calendar-multiget REPORT	N	N		N					N		N	
3.2.1. Query a specific href and return all data. (tests <calendar-multiget>)	N	N		N					N		N	
3.2.2. Query multiple hrefs (some of which do not exist) and return all data. (tests <calendar-multiget>)	N	N		N					N		N	
3.2.3. Query a specific href and return ETag WebDAV property and all data. (tests <calendar-multiget>+<DAV:prop>)	N	N		N					N		N	
3.2.4. Query multiple hrefs (some of which do not exist) and return ETag WebDAV property and all data. (tests <calendar-multiget>+<DAV:prop>)	N	N		N					N		N	
3.2.5. Query a specific href and return VEVENT components with only DTSTART, DTEND/DURATION, SUMMARY, UID, SEQUENCE, RRULE, RDATE, EXRULE, EXDATE, RECURRENCE-ID. (tests <calendar-query>, <calendar-data><comp><prop>)	N	N		N					N		N	
3.2.6. Query multiple hrefs (some of which do not exist) and return VEVENT components with only DTSTART, DTEND/DURATION, SUMMARY, UID, SEQUENCE, RRULE, RDATE, EXRULE, EXDATE, RECURRENCE-ID. (tests <calendar-query>, <calendar-data><comp><prop>)	N	N		N					N		N	
4. Event deletion												

	Server 1			Server 2			Server 3			Server 4		
	A	B	C	A	B	C	A	B	C	A	B	C
4.1. Delete a single non-recurring meeting.	P		P						P		N	P
4.2. Delete a single recurring meeting with no overridden instances.	P		P						P		N	P
4.3. Delete a single recurring meeting with overridden instances.	P		P						P		N	P
4.4. Delete a non-overridden instance of a recurring meeting.	P		P		P				P		N	P
4.5. Delete an overridden instance of a recurring meeting.	P		P		P				P		N	P
5. Access Control												
5.1. View access control details on current user's main calendar.	N		N	N	N				N		N	
5.2. Change access control details on current user's main calendar to add another user with read-only access. Verify that other user can view the calendar but not change it.	N		N	N	N				N		N	
5.3. Change access control details on current user's main calendar to add another user with read-write access. Verify that other user can view the calendar and change it. Verify that changes done by one user are seen by the other.	N		N	N	N				N		N	
5.4. Remove another user's access to the current user's main calendar and verify they can no longer access the calendar.	N		N	N	N				N		N	
6. Calendar Management												
6.1. Browse the list of calendars on the server, including the current user's personal calendars.	N	N	P	N	N				P		N	
6.2. Create a new calendar in the current user's personal calendar space.	N	P	P	N	P				P		N	
6.3. Create a regular collection in the current user's personal calendar space.	N	N	N	N	N				N		N	
6.4. Create a new calendar inside the collection created in 6.3.	N	N	N	N	N				N		N	
6.5. Delete the calendar created in 6.2.	N	P	P	N	P				F		N	
6.6. Delete the collection created in 6.3.	N	N	N	N	N				N		N	
7. Free Busy Reports												
Setup Create a new calendar and populate it with the following for one week:	N	N		N	N						N	P
Event on Monday, 9 am — 11 am, recurs every day for five times												
Event on Monday, 12 pm — 1 pm, status tentative												
Event on Monday, 2 pm — 3 pm, status cancelled												
Event on Tuesday, 11 am — 12 pm												
Event on Tuesday, 2 pm — 4 pm, recurs every day for four times												
Event on Tuesday, 3 pm — 5 pm												
Event on Wednesday, 11 am — 12 pm, status tentative												
Event on Wednesday, 3 pm — 5 pm, status tentative												
Event on Thursday, 11 am — 12 pm, status cancelled												
Event on Thursday, 3 pm — 5 pm, status cancelled												
7.1. Run a free-busy report for the entire week.	N	N	P	N	N				F		N	P
7.1.1. Verify two FREEBUSY periods for Monday, the second is BUSY-TENTATIVE.	N	N	P	N	N				F		N	P
7.1.2. Verify two FREEBUSY periods for Tuesday.	N	N	P	N	N				F		N	P
7.1.3. Verify four FREEBUSY periods for Wednesday, second and fourth are BUSY-TENTATIVE and one hour long.	N	N	P	N	N				F		N	P
7.1.4. Verify two FREEBUSY periods for Thursday.	N	N	P	N	N						N	P
7.1.5.	N	N	P	N	N				F		N	P
	N	N		N	N						N	P
	N	N		N	N						N	P
	N	N		N	N						N	P
7.1.5. Verify two FREEBUSY periods for Friday.	N	N	P	N	N						N	P
8. Scheduling												

		Server 1			Server 2			Server 3			Server 4		
		A	B	C	A	B	C	A	B	C	A	B	C
Setup	Three user accounts user1 (role Organizer), user2 (role Attendee), user3 (role Attendee) provisioned with suitable principal properties for calendar home, inbox, outbox and user addresses.	N	P	N	N	N	N						N
8.1	Organizer (user1) sends non-recurring message invite for Monday at 9am (1 hour) to each attendee. Verify that each attendee Inbox receives a copy of the invite.	N	P	N	N	N	N						N
8.2	Attendee (user2) accepts invite and sends back reply. Verify that reply is placed in Organizer Inbox.	N	P	N	N	N	N						N
8.3	Organizer (user1) updates invite with user2 accept state and resends invite. Verify that each attendee Inbox receives a copy of the new invite.	N	N	N	N	N	N						N
8.4	Attendee (user3) accepts updated invite and sends back reply. Verify that reply is placed in Organizer Inbox.	N	N	N	N	N	N						N
8.5	Organizer (user1) updates invite with user3 accept state and resends invite. Verify that each attendee Inbox receives a copy of the new invite.	N	N	N	N	N	N						N
8.6	Organizer (user1) cancels the invite. Verify that each attendee Inbox receives the cancellation.	N	P	N	N	N	N						N

A, B,	Client implementations
C	
P	Pass
F	Fail
N	Not supported