CALCONNECT DOCUMENT CD 0504

Type: **Report**
Title: **June 2005 CalConnect Interoperability Test Report**
Version: **1.0**
Date: **2005-06-18**
Status: **Published**
Source: **IOPTEST Technical Committee**

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.

CALCONNECT.ORG
The CALENDARING & SCHEDULING CONSORTIUM

June 18, 2005

TO: Calconnect Members

Re: June 1-3, 2005 Interop Test Results

The following report details the results from the interop testing event held at Duke University in Durham on June 1-3, 2005. The report includes text as well as a spreadsheet with actual results of scenario testing.

The attached version includes the public, abridged text and is meant for public distribution.

Respectfully submitted,

Patricia Egen
Interop Manager
Calendaring and Scheduling Consortium

:attachments

June 1-3, 2005 CALDAV Interop Test Results
Duke University, Durham, NC

The following organizations and personnel participated in a two-day interoperability testing event held on June 1-3, 2005 at Duke University.

Organization	Representative
Isamet	Cyrus Daboo
Mozilla	Dan Mosedale
Rodney Price	Novell**
Simon Vaillancourt	Oracle
Mike Douglass	Rensselaer Polytechnic Institute (RPI)

** Novell was auditing the testing event – their product was not tested.

Versions tested are as follows:

Isamet:
Mulberry v4.0	- client (publicly released)
Mulberry J2ME v0.1	- client (not released)
ISAMET CalDAV server v0.2	- server (not released)

Mozilla:
Mozilla Lightening Client	– no version number available (yet)

Oracle:
CalConnectCalDAVTest script	- Version 1.0
OracleCalDAV server	- June-2005 version!

RPI:
UWCalendar	- Version RPI-2.3.1

Interop Discussion Items

At the beginning of the interoperability event, we held a general discussion with each of the participants. They in turn discussed what would be tested, and the current status of their implementations. We also agreed upon userids, server names and IP addresses to access the servers.

VendorA, B and C provided URL's for connection to their servers. Two were situated locally on the intranet and one was on a public IP address. These three servers were then used during testing by the two clients.

VendorA tested two clients and one server. They support level 5 of Caldav. Their desktop client has been publicly released. Their server had a few minor changes since the last interop testing. In addition, they are working on a FreeBusy report. Multi Get has been implemented in their client. Their mobile client had no changes since the last interop testing event.

VendorC tested their client; however, their server had an issue arise so it was not tested during the interop. There are no version numbers yet assigned. They support version 5 of the Caldav draft.

VENDORB tested both a client and a server implementation. They also support version 5 of the Caldav draft. They are working on the Keep Alive connection. Simon noted that we need to ensure that applications are also WEBDAV and HTTP compliant since much of the CALDAV draft rides upon those specifications. For the purpose of this event, we are assuming applications support the current levels of both WEBDAV and HTTP.

VendorD has a server and runs on a skeletal Webdav server. They do not support ACLs or recurrences. Their Put/Get works and they support reports. They support level 5 of the Caldav draft as well.

A set of test scenarios had been provided by Bernard Desruisseaux from Oracle. These scenarios focused on a specific set of events for testing. The events tested were as follows:

Calendar Events
- Creation
- Modification
- Deletion
- Query

Access Control
- Creation
- Modification
- Deletion

Several tables at the end of this report breakdown each test scenario. There is a table for each client that show how each of the three servers responded to their calendar submissions. There is also a table that show what each server supports.

Summary

Overall, the specific items tested went well. Not all of the specification was tested – what was tested did not point out any specific issues with the draft.

Items of note are:

Access Control Lists are not yet supported by any of the servers or clients.

Clients and servers both had some issues with respect to their implementation of the draft components. Several of those items were fixed during the testing. All clients still have some components that are not yet working and are still under development.

Two servers support the majority of the items selected for testing during this event. The third server supports less but is well on its way towards being a robust Caldav server.

With regards to RFC2445, the iCalendar specification, there appears to be an issue with recurring rules with DTSTART in UTC. While it is not valid in RFC2445, several of the vendors support DTSTART in UTC. Therefore, this issue needs to be addressed – either the RFC2445 must be changed or the vendors need to change their applications to support the

current RFC2445. This is an issue that will be turned over to the Calsify technical group as it will be pertinent to the simplification work going on for RFC2445.

Going forward, we need to get additional vendors into the testing group. It would be good to see if we can find additional mobile vendors for testing as well. The consortium will be looking into ways of expanding the number of vendors that participate during interoperability events. The more we test, the more we ensure a good specification.

In order to move the draft to proposed standard level within the IETF (after it becomes an RFC), we will need to test all the MUSTS/SHOULDS/MUST NOTS/SHOULD NOTS in the draft in order to be considered interoperable. Pat Egen, the interop manager, volunteered to put together a matrix of these components and will post this on the public Calconnect server.

Finally, the following are comments by two of the participants stating how they felt the interop went overall.

APPENDIX A

SUPPORTING TABLES – TEST RESULTS

VENDORA DESKTOP CLIENT

VENDORA Desktop Client			P = Pass / F = Fail / N = Not supported		
VENDORA	**VENDORB**	**VENDORD**			
			1.	**Event creation.**	
P	P	P	1.1.	Create new single-instance meeting titled "Meeting 1.1" with the location "Durham".	
P	P	N	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	N	1.3.	Create new single-instance meeting titled "Meeting 1.3" with 2 other attendees.	
P	P	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.	
			2.	**Event modification**	
P	P	P	2.1.	Modify the title of meeting "Meeting 1.1" to "Meeting 1.1bis".	
P	P	P	2.2.	Modify the location of the meeting "Meeting 1.1bis" to "Seattle bis".	
P	P	P	2.3.	Reschedule meeting "Meeting 1.1bis" to the next day.	
P	P	N	2.4.	Add an attendee to "Meeting 1.1bis".	
P	P	N	2.5.	Add an alarm to "Meeting 1.1bis".	
N	N	N	2.6.	Modify the title of the 1st instance of the recurring meeting created in 1.2.	
N	N	N	2.7.	Modify the participation status of 1st instance to DECLINED.	
P	P	N	2.8.	Cancel the 4th instance of the recurring meeting created in 1.2.	
P	P	P	2.9.	One client changes "Meeting 1.1bis" to a different time, second client 'refreshes' its display to see the modification.	
			3.	**Event retrieval**	
			3.1.	calendar-query REPORT	
			3.1.1.	No filtering (match everything)	
N	N	N	3.1.1.1.	Query all components and return all data. (tests <calendar-query> and <filter>)	
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	3.1.1.3.	Query all components and return just entire VEVENT components. (tests <calendar-query> , <filter>+<comp-filter>)	
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>)	
			3.1.2.	time-range filtering	

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 non-overridden instance within the time-range. (tests <calendar-query> , <filter>+<time-range>)
N	N	N	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>)
			3.1.3.	component based filtering
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	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>)
			3.1.4.	property based filtering
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	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	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>)
			3.1.5.	parameter based filtering
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	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>)
			3.2.	calendar-multiget REPORT
N	N	N	3.2.1.	Query a specific href and return all data. (tests <calendar-multiget >)
P	P	P	3.2.2.	Query multiple hrefs (some of which do not exist) and return all data. (tests <calendar-multiget >)
N	N	N	3.2.3.	Query a specific href and return ETag WebDAV property and all data. (tests <calendar-multiget >+ <DAV:prop >)
P	P	P	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	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	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>)
			4.	**Event deletion**
P	P	P	4.1.	Delete a single non-recurring meeting.
P	P	N	4.2.	Delete a single recurring meeting with no overridden instances.
N	N	N	4.3.	Delete a single recurring meeting with overridden instances.
N	P	N	4.4.	Delete a non-overridden instance of a recurring meeting.
N	N	N	4.5.	Delete an overridden instance of a recurring meeting.
			5.	**Access Control**
P	N	N	5.1.	View access control details on current user's main calendar.
P	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.
P	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.
P	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.

VENDORA Mobile Client					
VENDORA	**VENDORB**	**VENDORD**		**P = Pass / F = Fail / N = Not supported**	
			1.	**Event creation.**	
N	N	N	1.1.	Create new single-instance meeting titled "Meeting 1.1" with the location "Durham".	
N	N	N	1.2.	Create new meeting titled "Meeting 1.2" recurring every Monday from 10:00 AM to 11:00 AM for 4 weeks.	
N	N	N	1.3.	Create new single-instance meeting titled "Meeting 1.3" with 2 other attendees.	
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.	
			2.	**Event modification**	
N	N	N	2.1.	Modify the title of meeting "Meeting 1.1" to "Meeting 1.1bis".	
N	N	N	2.2.	Modify the location of the meeting "Meeting 1.1bis" to "Seattle bis".	
N	N	N	2.3.	Reschedule meeting "Meeting 1.1bis" to the next day.	
N	N	N	2.4.	Add an attendee to "Meeting 1.1bis".	
N	N	N	2.5.	Add an alarm to "Meeting 1.1bis".	
N	N	N	2.6.	Modify the title of the 1st instance of the recurring meeting created in 1.2.	
N	N	N	2.7.	Modify the participation status of 1st instance to DECLINED.	
N	N	N	2.8.	Cancel the 4th instance of the recurring meeting created in 1.2.	
N	N	N	2.9.	One client changes "Meeting 1.1bis" to a different time, second client 'refreshes' its display to see the modification.	
			3.	**Event retrieval**	
			3.1.	calendar-query REPORT	
			3.1.1.	No filtering (match everything)	
N	N	N	3.1.1.1.	Query all components and return all data. (tests <calendar-query> and <filter>)	
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	3.1.1.3.	Query all components and return just entire VEVENT components. (tests <calendar-query> , <filter>+<comp-filter>)	
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>)	
			3.1.2.	time-range filtering	

P	P	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 non-overridden instance within the time-range. (tests <calendar-query> , <filter>+<time-range>)
P	P	N	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>)
			3.1.3.	component based filtering
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	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>)
			3.1.4.	property based filtering
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	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	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>)
			3.1.5.	parameter based filtering
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	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>)
			3.2.	calendar-multiget REPORT
N	N	N	3.2.1.	Query a specific href and return all data. (tests <calendar-multiget >)
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	3.2.3.	Query a specific href and return ETag WebDAV property and all data. (tests <calendar-multiget >+ <DAV:prop >)
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	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	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>)
			4.	**Event deletion**
N	N	N	4.1.	Delete a single non-recurring meeting.
N	N	N	4.2.	Delete a single recurring meeting with no overridden instances.
N	N	N	4.3.	Delete a single recurring meeting with overridden instances.
N	N	N	4.4.	Delete a non-overridden instance of a recurring meeting.
N	N	N	4.5.	Delete an overridden instance of a recurring meeting.
			5.	**Access Control**
N	N	N	5.1.	View access control details on current user's main calendar.
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	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	5.4.	Remove another user's access to the current user's main calendar and verify they can no longer access the calendar.

VENDORC CLIENT

VENDORC Client					**P = Pass / F = Fail / N = Not supported**
VENDORA	**VENDORB**	**VENDORD**			
			1.	**Event creation.**	
N (3)	N	N	1.1.	Create new single-instance meeting titled "Meeting 1.1" with the location "Durham".	
N (3)	N	N	1.2.	Create new meeting titled "Meeting 1.2" recurring every Monday from 10:00 AM to 11:00 AM for 4 weeks.	
N (3)	N	N	1.3.	Create new single-instance meeting titled "Meeting 1.3" with 2 other attendees.	
N (3)	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.	
			2.	**Event modification**	
P	P	P	2.1.	Modify the title of meeting "Meeting 1.1" to "Meeting 1.1bis".	
P	P	P	2.2.	Modify the location of the meeting "Meeting 1.1bis" to "Seattle bis".	
P	P	P	2.3.	Reschedule meeting "Meeting 1.1bis" to the next day.	
P	P	N	2.4.	Add an attendee to "Meeting 1.1bis".	
N	N	N	2.5.	Add an alarm to "Meeting 1.1bis".	
N	N	N	2.6.	Modify the title of the 1st instance of the recurring meeting created in 1.2.	
N	N	N	2.7.	Modify the participation status of 1st instance to DECLINED.	
N	N	N	2.8.	Cancel the 4th instance of the recurring meeting created in 1.2.	
(NOT TESTED)	(NOT TESTED)	P	2.9.	One client changes "Meeting 1.1bis" to a different time, second client 'refreshes' its display to see the modification.	
			3.	**Event retrieval**	
			3.1.	calendar-query REPORT	
			3.1.1.	No filtering (match everything)	
N	N	N	3.1.1.1.	Query all components and return all data. (tests <calendar-query> and <filter>)	
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	3.1.1.3.	Query all components and return just entire VEVENT components. (tests <calendar-query> , <filter>+<comp-filter>)	
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>)	
			3.1.2.	time-range filtering	

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 non-overridden instance within the time-range. (tests <calendar-query> , <filter>+<time-range>)
P (2) - except override	P (1) - except recurr	P (1) - except recurr	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>)
			3.1.3.	component based filtering
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	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>)
			3.1.4.	property based filtering
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	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	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>)
			3.1.5.	parameter based filtering
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	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>)
			3.2.	calendar-multiget REPORT
N	N	N	3.2.1.	Query a specific href and return all data. (tests <calendar-multiget >)
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	3.2.3.	Query a specific href and return ETag WebDAV property and all data. (tests <calendar-multiget >+ <DAV:prop >)
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	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	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>)
			4.	**Event deletion**
P	P	P	4.1.	Delete a single non-recurring meeting.
P	N	N	4.2.	Delete a single recurring meeting with no overridden instances.
N	N	N	4.3.	Delete a single recurring meeting with overridden instances.
N	N	N	4.4.	Delete a non-overridden instance of a recurring meeting.
N	N	N	4.5.	Delete an overridden instance of a recurring meeting.
			5.	**Access Control**
N	N	N	5.1.	View access control details on current user's main calendar.
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	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	5.4.	Remove another user's access to the current user's main calendar and verify they can no longer access the calendar.

Notes:

(1)   Recurrence between VENDORC and VENDORB is currently non-functional

(2)   VENDORC's overriding/exceptions is not working yet

(3)   Current VENDORC Lightening VI doesn't support creation other than via drag-to-create and modify to change things like title. That will change soon. The backend code supports 1.1-1.3 but not yet 1.4

VENDORC used their Lightening product for testing

SERVER SUPPORT

Server - Support					P = Pass / F = Fail / N = Not supported
VENDORA	**VENDORB**	**VENDORD**			
			1.	**Event creation.**	
P	P	P	1.1.	Create new single-instance meeting titled "Meeting 1.1" with the location "Durham".	
P	P	N	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	N	1.3.	Create new single-instance meeting titled "Meeting 1.3" with 2 other attendees.	
P	P	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.	
			2.	**Event modification**	
P	P	P	2.1.	Modify the title of meeting "Meeting 1.1" to "Meeting 1.1bis".	
P	P	P	2.2.	Modify the location of the meeting "Meeting 1.1bis" to "Seattle bis".	
P	P	P	2.3.	Reschedule meeting "Meeting 1.1bis" to the next day.	
P	P	N	2.4.	Add an attendee to "Meeting 1.1bis".	
P	P	N	2.5.	Add an alarm to "Meeting 1.1bis".	
P	P	N	2.6.	Modify the title of the 1st instance of the recurring meeting created in 1.2.	
P	P	N	2.7.	Modify the participation status of 1st instance to DECLINED.	
P	P	N	2.8.	Cancel the 4th instance of the recurring meeting created in 1.2.	
P	P	P	2.9.	One client changes "Meeting 1.1bis" to a different time, second client 'refreshes' its display to see the modification.	
			3.	**Event retrieval**	
			3.1.	calendar-query REPORT	
			3.1.1.	No filtering (match everything)	
P	P	P	3.1.1.1.	Query all components and return all data. (tests <calendar-query> and <filter>)	
P	P	P	3.1.1.2.	Query all components and return ETag WebDAV property and all data. (tests <calendar-query>+ <DAV:prop> and <filter>)	
P	P	P	3.1.1.3.	Query all components and return just entire VEVENT components. (tests <calendar-query> , <filter>+<comp-filter>)	
P	P	P	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>)	
			3.1.2.	time-range filtering	

P	P	P	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 non-overridden instance within the time-range. (tests <calendar-query> , <filter>+<time-range>)
P	P	P	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>)
			3.1.3.	component based filtering
P	P	N	3.1.3.1.	Query all components that contain an embedded VALARM component. (tests <calendar-query> , <filter>+<comp-filter>)
P	P	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>)
			3.1.4.	property based filtering
P	P	P	3.1.4.1.	Query all components that contain any ORGANIZER property. (tests <calendar-query> , <filter>+<prop-filter>+<is-defined>)
P	P	F	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>)
P	P	F	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>)
			3.1.5.	parameter based filtering
P	P	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>)
P	P	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>)
			3.2.	calendar-multiget REPORT
P	P	P	3.2.1.	Query a specific href and return all data. (tests <calendar-multiget >)
P	P	P	3.2.2.	Query multiple hrefs (some of which do not exist) and return all data. (tests <calendar-multiget >)
P	P	P	3.2.3.	Query a specific href and return ETag WebDAV property and all data. (tests <calendar-multiget >+ <DAV:prop >)
P	P	P	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 >)

P	P	P	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>)
P	P	P	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>)
			4.	**Event deletion**
P	P	P	4.1.	Delete a single non-recurring meeting.
P	P	N	4.2.	Delete a single recurring meeting with no overridden instances.
P	P	N	4.3.	Delete a single recurring meeting with overridden instances.
P	P	N	4.4.	Delete a non-overridden instance of a recurring meeting.
P	P	N	4.5.	Delete an overridden instance of a recurring meeting.
			5.	**Access Control**
N	N	N	5.1.	View access control details on current user's main calendar.
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	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	5.4.	Remove another user's access to the current user's main calendar and verify they can no longer access the calendar.