CALCONNECT THE CALENDARING & SCHEDULING CONSORTIUM

CALCONNECT DOCUMENT CD 0502

Type:	**Report**
Title:	**January 2005 CalConnect Interoperability Test Report**
Version:	**1.0**
Date:	**2005-01-12**
Status:	**Published**
Source:	**IOPTEST Technical Committee**

Calconnect V
Interoperability Testing of RFC 2445 and the current CALDAV draft*
University of Washington - Seattle
January 11-12, 2005

Participants:
Pat Egen - Interop Manager - Calendaring and Scheduling Consortium
Cyrus Daboo – Isamet
Mike Shaver - Mozilla
Dan Mosedale – Mozilla
Bernard Desruisseaux – Oracle
Simon Vaillancourt – Oracle
Greg Barnes - University of Washington
Scott Heyano – University of Washington

Products Tested:
Oracle CALDAV server
Isamet CALDAV server and two Isamet clients
Mozilla clients

Introduction:

In January, the Calendaring and Scheduling Consortium hosted their second interoperability event at the University of Washington in Seattle. This was actually the fifth in the series of interop events testing calendaring standards. The main purpose of the event is to exercise as many client/server pairs as possible to identify what interoperates and what does not.

During this event, two different groups did testing. The first group tested interoperability between CalDav clients and servers. CALDAV is current a draft (see footnote below) and not yet an RFC. However, it is always a good idea to start testing draft specifications as early as possible to ensure a draft is heading in the right direction. Considerable work has already been implemented using the draft in its current state. Therefore, there is enough of a code base to facilitate interoperability testing. At this event, two servers and three client applications were tested.

The second set of testing was reading iCalendar objects created by an application developed by the University of Washington. Their objects only used the RFC 2445 specification. They have not added iTIP (RFC 2447) and iMIP (RFC 2446) functionality but this testing was an excellent opportunity to test how different applications and servers reacted to their ics files.

Summary:

In summary, the server implementations basically worked and did what they were supposed to do. The clients are well along and passed the majority of the scenarios. There are some pieces left to fill in but this is early in the interop testing scheme of things for CALDAV. The results are pretty great for a specification so young. This bodes well for the future of this spec.

CALDAV testing results:

A set of 19 basic testing scenarios were used to test the servers and clients. We tested event creation, event modification, event retrieval and event deletion. One significant thing to note about this interop is this is the first time we are using a server-based application to store and retrieve events.

Some general items were noted by each of the vendors present. They are as follows:

Issues and findings:

It was determined there is no real good reason for a server to remember a URL specified by a client. Implementers want to be able to forget a URI but do want to be able to send back the real URI of the meeting that is the final object. There was discussion that there may have been something in the CAP draft that handles this. That may be a resource for text to add to CALDAV.

When a client does a PUT the server sends back what the URI is at the end of the process. There is dialog out on the CALDAV list that may point to possible resolutions.

If an underlying WEBDAV server does not support ACL or locking, then the client must handle everything. Locking would then be handled by tags and "if match" - you have to do a put operation that fails - a lock will guarantee you have a connection if you don't get data back down the pipe.

It was determined it is hard to lock a collection. The question is will this cause a performance hit on the server?

In order to determine if locking is required, we need good usecases or test scenarios to prove or justify requiring locking. It was suggested that implementers use eTags as a workaround. However, server implementations will need to handle limits on locks. This gives rise to the question of how to handle things when the lock is broken and the eTag has changed. Consensus among the group was to not make it mandatory.

One question that came up pertained to HREFs. The question is “are HREF’s allowed to be relative in Webdav?”

The majority of issues found during testing were not with the CALDAV specifications, but more with RFC2445 interpretations. For example, section 4.6.5 of RFC2445 addresses vtimezones and rrules. The first paragraph in this section seems to be interpreted differently by various clients. We will need to post this issue to the Calsify mailing list in the hopes that group can add this to their list of required changes/enhancements to RFC2445.

It was discussed that there might be an issue with propstat in the Caldav draft. Bernard from Oracle is not sure and will research this.

The following page shows the cases tested. The names of the clients are not noted – we use a code instead. C1, C2 and C3 are clients and S1 and S2 are servers.

Results of Test Scenarios:

C1 > S1	C1 > S2	C2 > S1	C3 > S1	C2 > S2	C3 > S2	**Test Scenario**
						1. Event creation.
Y	Y	Y	N/A	Y	N/A	1.1. Create new single-instance meeting titled "Meeting 1.1" with the location "Seattle".
F	Y	Y	N/A	Y	N/A	1.2. Create new meeting titled "Meeting 1.2" recurring every Monday from 10:00 AM to 11:00 AM for 4 weeks.
Y	Y	N	N/A	N	N/A	1.3. Create new single-instance meeting titled "Meeting 1.3" with 2 other attendees.
n/A	n/a	N	N/A	N	N/A	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
Y		Y	N/A	Y	N/A	2.1. Modify title of meeting "Meeting 1.1" to "Meeting 1.1bis".
Y		Y	N/A	Y	N/A	2.2. Modify the location of the meeting "Meeting 1.1bis" to "Seattle bis".
Y		F	N/A	Y	N/A	2.3. Reschedule meeting "Meeting 1.1bis" to the next day.
Y		F	N/A	F	N/A	2.4. Add an attendee to "Meeting 1.1bis".
		F	N/A	Y	N/A	2.5. Add an alarm to "Meeting 1.1bis".
		F	N/A	F	N/A	2.6. Modify the title of the 1st instance of the recurring meeting created in 1.2.
		F	N/A	F	N/A	2.7. Modify the participation status of
		F	N/A	Y	N/A	2.8. Cancel the 4th instance of the recurring meeting created in 1.2.
Y		F	N/A	N/A	N/A	2.9. One client changes "Meeting 1.1bis" to a different time, second client 'refreshes' its display to see the modification.
						3. Event retrieval
Y		Y	Y	Y	Y	3.1. Retrieve all the components and properties of the meetings scheduled for the current day (i.e., midnight to midnight local time).
Y		Y	Y	Y	Y	3.2. Retrieve all the components and properties of the meetings scheduled for the current week.
F		Y	Y	Y	Y	3.3. Retrieve all the components and properties of the meetings scheduled for the current month.
F		N/A	N/A	Y	N/A	3.4. Retrieve only the SUMMARY, DTSTART and DTEND (or DURATION) properties of meetings scheduled for current week.
						4. Event deletion
Y		Y	N/A	Y	N/A	4.1. Delete the meeting titled "Meeting 1.1bis"
N		Y	N/A	Y	N/A	4.2. Delete the meeting titles "Meeting 1.2"

*draft-dusseault-caldav-04