

June 2005 CalConnect Interoperability Test Scenarios

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.

© 2005 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
1. Test Scenarios.....1

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>.

June 2005 CalConnect Interoperability Test Scenarios

1. Test Scenarios

1. Event creation.

- 1.1. Create new single-instance meeting titled "Meeting 1.1" with the location "Durham".
- 1.2. Create new meeting titled "Meeting 1.2" recurring every Monday from 10:00 AM to 11:00 AM for 4 weeks.
- 1.3. Create new single-instance meeting titled "Meeting 1.3" with 2 other attendees.
- 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

- 2.1. Modify the title of meeting "Meeting 1.1" to "Meeting 1.1bis".
- 2.2. Modify the location of the meeting "Meeting 1.1bis" to "Seattle bis".
- 2.3. Reschedule meeting "Meeting 1.1bis" to the next day.
- 2.4. Add an attendee to "Meeting 1.1bis".
- 2.5. Add an alarm to "Meeting 1.1bis".
- 2.6. Modify the title of the 1st instance of the recurring meeting created in 1.2.
- 2.7. Modify the participation status of 1st instance to DECLINED.
- 2.8. Cancel the 4th instance of the recurring meeting created in 1.2.
- 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)
 - 3.1.1.1. Query all components and return all data. (tests <calendar-query> and <filter>)
 - 3.1.1.2. Query all components and return ETag WebDAV property and all data. (tests <calendar-query>+<DAV:prop> and <filter>)
 - 3.1.1.3. Query all components and return just entire VEVENT components. (tests <calendar-query>, <filter>+<comp-filter>)
 - 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
 - 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>)
 - 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
 - 3.1.3.1. Query all components that contain an embedded VALARM component. (tests <calendar-query>, <filter>+<comp-filter>)
 - 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
 - 3.1.4.1. Query all components that contain any ORGANIZER property. (tests <calendar-query>, <filter><prop-filter><is-defined>)
 - 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>)
 - 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

- 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>)
- 3.1.5.2. Query all components that contain an ATTENDEE property with PARTSTAT=NEEDSACTION parameter. (tests <calendar-query>, <filter><prop-filter><text-match><param-filter><text-match>)
- 3.2. calendar-multiget REPORT
 - 3.2.1. Query a specific href and return all data. (tests <calendar-multiget>)
 - 3.2.2. Query multiple hrefs (some of which do not exist) and return all data. (tests <calendar-multiget>)
 - 3.2.3. Query a specific href and return ETag WebDAV property and all data. (tests <calendar-multiget>+<DAV:prop>)
 - 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>)
 - 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>)
 - 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**
 - 4.1. Delete a single non-recurring meeting.
 - 4.2. Delete a single recurring meeting with no overridden instances.
 - 4.3. Delete a single recurring meeting with overridden instances.
 - 4.4. Delete a non-overridden instance of a recurring meeting.
 - 4.5. Delete an overridden instance of a recurring meeting.
- 5. Access Control**
 - 5.1. View access control details on current user's main calendar.
 - 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.
 - 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.
 - 5.4. Remove another user's access to the current user's main calendar and verify they can no longer access the calendar.
- 6. Calendar Management**
 - 6.1. Browse the list of calendars on the server, including the current user's personal calendars.
 - 6.2. Create a new calendar in the current user's personal calendar space.
 - 6.3. Create a regular collection in the current user's personal calendar space.
 - 6.4. Create a new calendar inside the collection created in 6.3.
 - 6.5. Delete the calendar created in 6.2.
 - 6.6. Delete the collection created in 6.3.