CalConnect TC

# Calendaring and scheduling — CalDAV Auditing Status and Feedback

Committee Draft Standard

**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.

The Calendaring and Scheduling Consortium, Inc. 2018

The Calendaring and Scheduling Consortium, Inc.

4390 Chaffin Lane
McKinleyville
California 95519
United States of America

copyright@calconnect.org
www.calconnect.org

# Contents

## Abstract

This document defines an extension to CalDAV that allows clients and servers that audit data on the server, to provide feedback based on the results of such audits. This can be used to signal clients about junk events, or other inappropriate content. It can also be used by clients to provide feedback about junk events, or other inappropriate content, that the server can act on in a manner that prevents the originator of the data from knowing it was discarded.

# Introduction

Internet calendaring and scheduling standards are defined by iCalendar IETF RFC 5545 and iTIP IETF RFC 5546. The CalDAV Access IETF RFC 4791 standard defines a way to access calendar data stored on a server, and the CalDAV Scheduling IETF RFC 6638 draft defines how scheduling occurs between users of a CalDAV server.

CalDAV calendar users can receive event invitations or sharing invitations from other users on the CalDAV server, and in the case of event invitations, they may also be received from users outside the system (e.g., via an email based gateway service). Unfortunately, as has been the case with email for quite some time now, this provides an avenue for abuse. In particular, there has recently been an increase in so called calendar "spam" (unsolicited bulk event invitations — in future referred to as "junk" invitations) that automatically appear on a user's calendar. Whilst users can delete such invitations, the current default behavior of CalDAV servers is to send a scheduling iTIP reply message back to the organizer of the invite indicating that the user (who appears as an attendee in the invite) has declined the invitation. This is not desirable as it signals to the originator that a speculative attendee calendar user address they might have used is in fact valid. It is much more preferable for the invite to be silently discarded without a reply being sent. CalDAV does support such an option via use of the "Schedule-Reply" HTTP header (see IETF RFC 6638, Section 8.1), however, it would be useful to provide an explicit indication from the user to the server that a particular invitation is considered inappropriate so that the server can use that information to potentially filter future invitations that are similar. This type of feedback has been used successfully in email systems, typically exposed to users as a "report as junk" option in their email clients.

Also, as with email, a server can scan invitations as they are being delivered and provide an indication of "junk" status to allow clients to filter invitations or provide warnings to the user. Such status is usually conveyed in email messages via the addition of email message headers during delivery, however, iCalendar invitations typically do not include delivery meta-data.

This specification defines the following extensions to CalDAV to help provide better junk invitation status and reporting:

— A new `CS:audit-status` WebDAV property is defined for use on any resource in a CalDAV server. The value of that property is a comma-separated list of "key=value" pairs that can be used to convey invitation auditing status from the server to a client.

— A new `POST` request action can be targeted at resources to allow a client to report a problem about the resource and trigger the server to take appropriate steps (e.g., discard the resource, record information about the resource so that it can be permanently suppressed).

This specification does not cover how servers or clients analyse invitations, but rather it only covers the reporting of such analysis. What is considered inappropriate is likely dependent on local policies and regulations, which cannot be generally codified. It is expected that email scanning tools and techniques can be adapted for use with calendar data on both clients and servers.

## 1. Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IETF RFC 2119, S. BRADNER. *Key words for use in RFCs to Indicate Requirement Levels*. 1997. RFC Publisher. https://www.rfc-editor.org/info/rfc2119.

IETF RFC 4791, C. DABOO, B. DESRUISSEAUX and L. DUSSEAULT. *Calendaring Extensions to WebDAV (CalDAV)*. 2007. RFC Publisher. https://www.rfc-editor.org/info/rfc4791.

IETF RFC 4918, L. DUSSEAULT (ed.). *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*. 2007. RFC Publisher. https://www.rfc-editor.org/info/rfc4918.

IETF RFC 5545, B. DESRUISSEAUX (ed.). *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. 2009. RFC Publisher. https://www.rfc-editor.org/info/rfc5545.

IETF RFC 5546, C. DABOO (ed.). *iCalendar Transport-Independent Interoperability Protocol (iTIP)*. 2009. RFC Publisher. https://www.rfc-editor.org/info/rfc5546.

IETF RFC 6638, C. DABOO and B. DESRUISSEAUX. *Scheduling Extensions to CalDAV*. 2012. RFC Publisher. https://www.rfc-editor.org/info/rfc6638.

# Calendaring and scheduling — CalDAV Auditing Status and Feedback

## 2. Conventions Used in This Document

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in IETF RFC 2119.

When XML element types in the namespaces “DAV:” and “urn:ietf:params:xml:ns:caldav” are referenced in this document outside of the context of an XML fragment, the string “DAV:” and “CALDAV:” will be prefixed to the element type names respectively.

The namespace “http://calendarserver.org/ns/” is used for XML elements defined in this specification. When XML element types in this namespace are referenced in this document outside of the context of an XML fragment, the string “CS:” will be prefixed to the element type names respectively.

## 3. Server Advertised Capability

A server that supports this specification `MUST` include “calendar-audit” as a field in the `DAV` response header field from an `OPTIONS` request on a calendar home collection (see IETF RFC 4791, Section 6.2.1). Clients `MUST` check for the presence of that field in the `DAV` response header field before supporting the extensions in this specification.

## 4. Audit WebDAV Property

The `CS:audit-status` WebDAV property provides any server side audit results, for the resource the property is present on, to the client. The value of this property is a string containing one or more comma- separated “key=value” pairs that can be used to conveying specific details about the audit results. A set of initial keys defined by this specification are listed below. Additional keys may be added in the future. Servers SHOULD only include audit results that are known to be useful to clients to avoid having this property grow too large.

Clients that support this extension SHOULD request the CS:audit-status property in any PROPFIND or REPORT requests that return properties for resources on the CalDAV server. In particular, calendar object resources will likely all be audited, and so SHOULD be checked by clients.

Clients MAY do their own auditing of resources retrieved from the server. Clients are likely to take into consideration locally available contextual information when doing client-side auditing, information that is typically not available to servers. As a result, when the CS:audit-status property is present, clients SHOULD use it in conjunction with, rather than as a replacement of, any client-side auditing to determine the overall suitability of the resource.

## 4.1. CS:audit-status

Name
audit-status

Namespace
http://calendarserver.org/ns/

Purpose
Indicates server auditing status for the resource on which this property is defined.

Conformance
This property MAY be present on any resource within a CalDAV calendar home collection. If present, it SHOULD NOT be returned by a PROPFIND DAV:allprop request (as defined in IETF RFC 4918, Section 14.2). This is a protected property (as defined in IETF RFC 4918, Section 15). This property MUST be preserved when the resource is copied or moved.

Description
The CS:audit-status property provides details of any server auditing done for the resource on which it is defined. In the absence of this property, clients MUST assume that no auditing has taken place. The supported key/value pairs defined in this specification are listed in the next section.

Definition

```
<!ELEMENT audit-status CDATA>
CDATA is a string using the "audit-status" format defined
below

audit-status = audit-state *("," audit-state)
audit-state = audit-key "=" audit-value
audit-key = token
audit-value = token / quoted-string

token = ALPHA *(ALPHA / DIGIT / "-")
quoted-string = DQUOTE *QSAFE-CHAR DQUOTE

QSAFE-CHAR = WSP / %x21 / %x23-7E / NON-US-ASCII
; Any character except CONTROL and DQUOTE

NON-US-ASCII = UTF8-2 / UTF8-3 / UTF8-4
; UTF8-2, UTF8-3, and UTF8-4 are defined in [RFC3629]

CONTROL = %x00-08 / %x0A-1F / %x7F
; All the controls except HTAB
```

Example
Note that additional line breaks have been added for readability.

```
<CS:audit-status
 xmlns:CS="http://calendarserver.org/ns/">
 status=GOOD,audit-id=51CAA5F8-5C48-45E1-B0E7-3534D7254060
```

```
            </CS:audit-status>
```

## 4.2. Audit Status Items

The following CS:audit-status key/value pair items are defined by this specification. The “status” MUST be present, other keys are OPTIONAL. Clients SHOULD ignore any keys that they do not recognize.

**Table 1 — CS:audit-status key/value pair items**

| Key | Value | Description |
|---|---|---|
| status | GOOD | Audit result indicates no problem with this resource |
| status | WARNING | Audit result indicates a possible problem with this resource |
| status | BAD | Audit result indicates a strong possibility of a problem with this resource |
| score | {integer: 0-100} | An integer value in the range 0-100. 0 means the resource is definitely free of problematic content. 100 means the resource definitely contains problematic content. Values in between relate to varying degrees of concern for the resource content. |
| reason | {string} | A textual description of the result of the server audit that can be presented to the user. |
| audit-id | {string} | An opaque identifier used to correlate the reported audit status with the auditing system. |

## 5. Client Audit Reporting

When a client detects a CalDAV resource that it has determined to be inappropriate for some reason, or as the result of the calendar user explicitly indicating that a resource is inappropriate, it can signal that state to the server by issuing a POST request with the request- URI set to the resource URI, and with an “action=audit-failure” query parameter in the request-URI.

Upon receiving such a request the server MUST take the following actions:

1) If the target resource is a calendar object resource, the server MUST delete the resource without sending any scheduling reply messages. The server MUST use the iCalendar UID property value in the target resource to:
    a) delete any other resources in calendar collections owned by the calendar user, or the calendar user’s scheduling inbox collection, that have the same iCalendar UID property
    b) prevent any future scheduling messages with the same iCalendar UID being delivered to that calendar user
2) If the target resource is a sharing invitation in the calendar user’s notification collection, the server MUST delete the notification resource without sending any sharing reply to the sharer. The server MUST also prevent any future sharing invitations for the same calendar collection from being delivered to the calendar user.
3) The behavior for other types of resource is currently undefined. Servers MUST reject such requests with an appropriate HTTP 4xx status code.

Servers MAY perform their own analysis of the resource being reported and act on it accordingly, but this specification does not define how that is done or what the consequences are.

The audit report POST request supports the following request-URI query parameters:

**Table 2 — Request-URI query parameters**

| Name | Description |
|---|---|
| action | this is REQUIRED and MUST have a value of “audit-failure” |
| reason | this is OPTIONAL and contains a short string indicating the nature of the failure |

The client MAY include other query parameters as needed. The server SHOULD ignore all query parameters that it cannot process.

On successful completion of the request, the server returns an appropriate HTTP 2xx status code. Upon receiving a successful response, clients SHOULD immediately re-synchronize their state with the server to ensure any resources that were removed as a result of the POST request are also removed from any cache the client might have.

If the request fails for any reason, the actions described above MUST NOT occur, and in particular, no resources are to be deleted.

This specification discards an entire resource — thus there is no provision to report inappropriate content in one instance of a recurring event since all instances are in the same calendar object resource.

### 5.1. Example

The client issues a POST request to indicate an audit failure on a particular resource:

```
>> Request <<

POST /event.ics?action=audit-failure&reason=junk HTTP/1.1
Host: cal.example.com
Content-Length: 0

>> Response <<

HTTP/1.1 200 OK
Date: Fri, 10 Jan 2017 14:02:20 GMT
Content-Type: text/plain
Content-Length: xxxx
```

Report received and acted upon.

## 6. Security Considerations

It is important that no replies whatsoever be sent back to the originator of the invitation being discarded. In addition, the invite originator MUST NOT be given any other indication that the invite was discarded.

The server MUST protect any information it gets from client audit failure reports to prevent attackers from learning how to work around client auditing procedures.

## 7. Privacy Considerations

The server MUST protect any information it gets from client audit failure reports to prevent calendar users from being “profiled” based on what items they consider to be inappropriate.

## 8. IANA Considerations

None.

## 9. Acknowledgments

This specification is the result of discussions between the Apple calendar server and client teams.

# Appendix A
(normative)
metanorma-extension

## A.1. document history

```
- date:
  - type: updated
    value: 2016-12-09
  amend:
    - description: |
        Clarify that clients should never totally rely on the server audit-
status result, but instead should always do their own auditing and use the
server status as input to that.

        Fix xmlns value used in examples.

        Remove reference to client reporting on the inbox scheduling resource.

        Indicate that servers can implement their own procedures for analyzing
invites reported by the client - but those are out of scope of this spec.
```

**Figure A.1**