

CC/CD 51004:2018

# **Calendaring and scheduling – Calendar subscription upgrades**

THE CALENDARING AND SCHEDULING CONSORTIUM

TC CALENDAR

EDITORS:

AUTHORS: MICHAEL DOUGLASS

## **CALCONNECT STANDARD**

© 2018 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](mailto:copyright@calconnect.org)  
[www.calconnect.org](http://www.calconnect.org)

# CONTENTS

Foreword

Introduction

1. Scope

2. Normative references

3. Terms and definitions

4. Enhanced GET

4.1. General

4.2. Deletions

5. Changes to the iCalendar specifications

6. Discovering alternative access methods

6.1. General

6.2. subscribe-caldav

6.3. subscribe-caldav-auth

6.4. subscribe-webdav-sync

6.5. subscribe-enhanced-get

7. Security Considerations

8. IANA Considerations

8.1. Link Relation Registrations

9. Acknowledgements

Appendix A(informative) Open issues

Appendix B(informative) Change log

Bibliography

## FOREWORD

The Calendaring and Scheduling Consortium ("CalConnect") is global non-profit organization with the aim to facilitate interoperability of collaborative technologies and tools through open standards.

CalConnect works closely with international and regional partners, of which the full list is available on our website (<https://www.calconnect.org/about/liaisons-and-relationships>).

The procedures used to develop this document and those intended for its further maintenance are described in the CalConnect Directives.

In particular the different approval criteria needed for the different types of CalConnect documents should be noted. This document was drafted in accordance with the editorial rules of the CalConnect Directives.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CalConnect shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be provided in the Introduction.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

This document was prepared by Technical Committee *CALENDAR*.

## INTRODUCTION

Currently clients subscribe to calendar feeds as an ics file which is often published as a resource accessible using the unofficial 'webcal' scheme.

The only available option for updating that resource is the usual HTTP polling of cached resources using Etags.

There is the usual tension between clients wishing to see a timely response to changes and servers not wishing to be overloaded by frequent requests for possibly large amounts of data.

This specification introduces an approach whereby clients can discover a more performant access method. Given the location of the resource as an ics file, the client can perform an OPTIONS request on the resource and inspect the returned headers which will offer a number of alternative access methods.

Given that many clients already support CalDAV this provides an easy upgrade path for those clients. CalDAV and DAV subsets are specified here to allow lighter weight implementations.

## 1. SCOPE

This document provides a mechanism to allow subscribers to calendar feeds to upgrade to a more performant protocol.

## 2. 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 2518, HTTP Extensions for Distributed Authoring – WEBDAV

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax

IETF RFC 4791, Calendaring Extensions to WebDAV (CalDAV)

IETF RFC 5545, Internet Calendaring and Scheduling Core Object Specification (iCalendar)

IETF RFC 5546, iCalendar Transport-Independent Interoperability Protocol (iTIP)

IETF RFC 6047, iCalendar Message-Based Interoperability Protocol (iMIP)

IETF RFC 6638, Scheduling Extensions to CalDAV

IETF RFC 5988, Web Linking

### **3. TERMS AND DEFINITIONS**

No terms and definitions are listed in this document.



## 4. ENHANCED GET

### 4.1. General

This is a lightweight protocol which allows simple clients to efficiently discover and download changes in the targeted resource.

It has many similarities to WebDAV sync and for a server could be implemented as an extension of the specification.

In this protocol the Etag is used as the sync change token. By adding the If-None-Match header field to the vary header field we can ensure intermediate caching proxies will be able to cache different versions of the data.

The resource is treated as a set of individual events each of which may be updated or deleted separately. The client will first fetch the entire ics file. On subsequent requests it uses the Prefer header with a value of "return=minimal" to indicate that it wants a set of changes since the last fetch.

### 4.2. Deletions

When an entity (VEVENT, VTODO or other valid top-level component) is deleted from the source data the server needs to be able to inform a client of the deletion. This specification introduces a new value for the STATUS property of DELETED.

On the first conditional GET after the entity has been deleted a skeleton, but valid, entity will be returned with STATUS: DELETED. The receiving client is free to remove the entity or update it's STATUS property.

On subsequent conditional fetches the entity will not be returned.

EXAMPLE 1	<p>This is an example of the initial request and response from a server that supports the extended GET protocol.</p> <pre>&gt;&gt; Request &lt;&lt;  GET /events.ics HTTP/1.1 Host: example.com Accept: text/calendar  &gt;&gt; Response &lt;&lt;  HTTP/1.1 200 OK Content-Length: xxxx ETag: "1234"                current ETag (for conditional GET) Vary: Prefer, If-None-Match    so caching proxy can key off of client's ETag (sync token) and preference  BEGIN:VCALENDAR: ? /* full feed */ END:VCALENDAR</pre>
-----------	---

EXAMPLE 2	
-----------	--

	<p>This is an example of the subsequent request and response when no changes have occurred. The Accept header field indicates that a VPATCH format is most desirable but simple text/calendar is acceptable.</p> <pre> &gt;&gt; Request &lt;&lt;  GET /events.ics HTTP/1.1 Host: example.com Accept: text/calendar; q=0.5, component=VPATCH, text/calendar; If-None-Match: ?1234?           conditional request Prefer: return=minimal  &gt;&gt; Response &lt;&lt;  HTTP/1.1 304 Not Modified Content-Length: 0 ETag: ?1234? Vary: Prefer, If-None-Match </pre>
--	---

EXAMPLE 3	<p>This is an example of the subsequent request and response when no changes have occurred. The Accept header field indicates that a VPATCH format is most desirable but simple text/calendar is acceptable.</p> <pre> &gt;&gt; Request &lt;&lt;  GET /events.ics HTTP/1.1 Host: example.com Accept: text/calendar; q=0.5, component=VPATCH, text/calendar; If-None-Match: "1234"           conditional request Prefer: return=minimal  &gt;&gt; Response &lt;&lt;  HTTP/1.1 304 Not Modified Content-Length: 0 ETag: "1234" Vary: Prefer, If-None-Match </pre>
-----------	---

EXAMPLE 4	<p>This is an example of the subsequent request and response when changes have occurred and the server can create the minimal format.</p> <pre> &gt;&gt; Request &lt;&lt;  GET /events.ics HTTP/1.1 Host: example.com Accept: text/calendar; q=0.5, component=VPATCH, text/calendar; If-None-Match: "1234"           conditional request Prefer: return=minimal  &gt;&gt; Response &lt;&lt;  HTTP/1.1 200 OK </pre>
-----------	---

	<pre> Content-Type: text/calendar Content-Length: xxxx ETag: "5678"                current ETag (for conditional GET) Preference- Applied: return=minimal    signals to client that stream is changes only Vary: Prefer, If-None- Match                    so caching proxy can key off of client? s ETag (sync token) and preference  BEGIN:VCALENDAR: ... only new/changed events ... when not returning VPATCH, deleted events have STATUS:DELETED END:VCALENDAR </pre>
--	--

EXAMPLE 5	<p>This is an example of the subsequent request and response when changes have occurred and the server cannot create the minimal format - perhaps because of an old or invalid token. Note there is no Preference-Applied header field.</p> <pre> &gt;&gt; Request &lt;&lt;  GET /events.ics HTTP/1.1 Host: example.com Accept: text/calendar; q=0.5, component=VPATCH, text/calendar; If-None-Match: "1234"        conditional request Prefer: return=minimal  &gt;&gt; Response &lt;&lt;  HTTP/1.1 200 OK Content-Type: text/calendar Content-Length: xxxx ETag: "5678"                current ETag (for conditional GET) Vary: Prefer, If-None- Match                    so caching proxy can key off of client? s ETag (sync token) and preference  BEGIN:VCALENDAR: ... full set of data END:VCALENDAR </pre>
-----------	--

## 5. CHANGES TO THE ICALENDAR SPECIFICATIONS

This specification does not require any changes to [RFC 5545](#) or its extensions. However it does introduce the use of some properties to provide more information about the resource, for example the time range it covers. Unresolved directive in csd-subscription-upgrade.adoc - include::sections/05-discovering.adoc[]

## 6. DISCOVERING ALTERNATIVE ACCESS METHODS

The advertising of other access points is achieved through the use of the LINK header as defined in [RFC 5988](#). New link relation types are defined in this specification - each being associated with a protocol or protocol subset.

These LINK headers will be delivered when a client carries out an OPTIONS request targeting the URL of the resource. == Link relations

### 6.1. General

This clause defines a number of new link relations required to facilitate subscription upgrades.

### 6.2. subscribe-caldav

This specifies an access point which is a full implementation of caldav but requires no authentication. The end point allows the full range of reports as defined by the CalDAV specification.

The client **MUST** follow the specification to determine exactly what operations are allowed on the access point - for example to determine if sync-report is supported.

The URL **MAY** include some form of token to allow write access to the targeted collection. The client must check it's permissions to determine whether or not it has been granted write access.

### 6.3. subscribe-caldav-auth

This specifies an access point which is a full implementation of caldav and requires authentication. This may allow read-write access to the resource.

The client **MUST** follow the specification to determine exactly what operations are allowed on the access point - for example to determine if sync-report is supported.

### 6.4. subscribe-webdav-sync

This specifies an access point which supports only webdav sync.

This allows the client to issue a sync-report on the resource to obtain updates.

NOTE say something about initial startup - use ics to populate? Initial token?

The client **MUST** follow that specification.

### 6.5. subscribe-enhanced-get

This specifies an access point which supports something new.

The client **MUST** follow that specification.

## 7. SECURITY CONSIDERATIONS

Applications using these properties need to be aware of the risks entailed in using the URIs provided as values. See [RFC 3986](#) for a discussion of the security considerations relating to URIs. = Privacy Considerations

Properties with a "URI" value type can expose their users to privacy leaks as any network access of the URI data can be tracked. Clients **SHOULD NOT** automatically download data referenced by the URI without explicit instruction from users. This specification does not introduce any additional privacy concerns beyond those described in [RFC 5545](#).

## 8. IANA CONSIDERATIONS

### 8.1. Link Relation Registrations

This document defines the following new iCalendar properties to be added to the registry defined in RFC 5545, Section 8.2.3:

Table 1

Relation Name	Description	Reference
subscribe-caldav	Current	<a href="#">Clause 6.2</a>
subscribe-caldav_auth	Current	<a href="#">la-subscribe-caldav_auth</a>
subscribe-webdav-sync	Current	<a href="#">Clause 6.4</a>
subscribe-enhanced_get	Current	<a href="#">la-subscribe-enhanced_get</a>

## 9. ACKNOWLEDGEMENTS

The author would also like to thank the members of the Calendaring and Scheduling Consortium Calendar Sharing technical committee and the following individuals for contributing their ideas and support:

...

The authors would also like to thank the Calendaring and Scheduling Consortium for advice with this specification.



## **APPENDIX A (INFORMATIVE) OPEN ISSUES**

restype values	Need to determine what if any registry of resource types already exists and use that.
----------------	---

## **APPENDIX B (INFORMATIVE) CHANGE LOG**

v01 2017-07-28 MD

- Examples
- More text for extended get. Talk about deletions.

v01 2017-02-17 MD

- Add text about OPTIONS
- Add text about read/write CalDAV

v00 2017-02-15 MD

- First pass

**BIBLIOGRAPHY**