

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 27, 2016

R. Menderico, Ed.
P. Schlup
L. Kristiansen
Google
November 24, 2015

**v-event URI: An URI scheme for events.
draft-menderico-v-event-uri-00**

Abstract

This document defines the format of Uniform Resource Identifiers (URIs) for calendar events, allowing users to add these events to their calendar application from any source that defines them, like web sites and printed QR codes

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 27, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

Calendar users currently often do not have the ability to quickly add an event to their default calendar app, when encountering event data on a webpage, poster or mobile apps. In this sense events have fallen behind other real world entities, like e-mail [[RFC6068](#)] and geo coordinates [[RFC5870](#)] which allow for performing actions in default apps when encountering these entities anywhere. This recommendations document addresses the problem by proposing best practices when embedding and publishing calendar data. We believe that using a standardized URI scheme for event publishing will make populating of users' calendars much simpler, will make developers' lives easier and will increase calendar apps usage in general. A major additional benefit of URIs is sharing of events on physical media (for example via QR codes) or via URL.

2. Motivation and related work

2.1. Current event publishing practices

Many ways to add events to calendar apps have been developed due to the lack of standardization. While these solve the basic need of sharing events, it does so with limited reach, are harder to use and limit adoption by new providers. We will briefly mention four common ways: using a download link, hCalendar embedding, using VEVENTs in QR codes and provider specific buttons.

2.1.1. Publish a link to an iCalendar file

A simple way to publish an event is to share a link that points to an iCalendar [[RFC5545](#)] file and provide a link to the user for downloading it. In an ideal world, a user would click on the link pointing to a file, the browser would recognize this link as a calendar file and redirect it to the proper calendar application, which would display the event information to the user, allowing simple edit actions, and saving it to the calendar.

There are many problems with this method. Files must be hosted in a server, which is not always feasible or easy, and need to be maintained separately from the web pages they are linked from. This poses unnecessary difficulty for blog and CMS users, especially when compared to linking to email addresses or other web pages.

Furthermore, ICS files might not be recognized by the browser or operating system, or the user might use a web-based calendar app instead of a native one. Files might also contain malware and pose additional risk, with some end users avoiding downloaded files altogether.

One additional problem is coupling of the file and the entity it represents. Currently the user needs to manually keep the event file and all the links and descriptions with the same information in the links about in sync. It is preferable to have this information close together. Ideally the same tools used to generate a page could be used to assure a link and the text information being displayed are synchronized.

2.1.2. Current QR code implementations

Several QR code readers ([\[ZXing\]](#)[\[i-nigma\]](#)) support calendar events to be embedded in QR codes, but they have their own non-standard implementation, usually a single VEVENT iCalendar component. They work well on the mobile platform but rely on OS internals [\[CalendarContract\]](#) to add the event to the calendar.

While QR code does not dictate a format for calendar events, most readers implement the URI standard schemes and would benefit from this proposal.

2.1.3. hCalendar

hCalendar [\[hCalendar\]](#) is a microformat for events that can be used to annotate a web page or another document and indicate to readers that an event is present. While microformats are useful for parsing and styling, they are not meant to be used as links to an event and need to be used in conjunction with the external ICS file method or a proprietary browser extension. They usually require a complex interaction between the website and the calendar application to get it right. This also makes it not ideal for QR scanning, since the annotations were not designed to represent the whole content of a document. They are more appropriate as an annotation on a previously structured text.

2.1.4. Custom calendar application link

Several calendar systems provide a proprietary URL for event creation ([\[AddThisEvent\]](#)[\[GoogleCalendar\]](#)). However, due to lack of standards, applications must implement and link to these separately. This burdens both content creators and web site users: content creators must maintain individual links for each calendar application provider, and end users must find and choose the appropriate one for their own case. This consumes precious space on the page and requires understanding of several APIs, making it difficult for a developer that wants to just publish a simple event. Furthermore, it's impossible to link to all calendar app providers, requiring a combination of this method with the ICS file hosting method.

2.2. Alternative implementations

Though not currently used by the most popular calendar applications, other implementations could theoretically converge into a standard. We will talk briefly about the calendar mime type in combination with RegisterContentHandler, data URI scheme and a custom URI scheme.

2.2.1. text/calendar MIME type in combination with RegisterContentHandler

One way to use iCal files and avoid downloading it would be registering a handler for calendar MIME [[RFC6838](#)] types (text/calendar [[RFC5545](#)]). Operating Systems know how to handle mime type properly, and are able to redirect it to the right application. Web browsers, on the other hand, still have limited capability to handle it. The method RegisterContentHandler [RegisterContentHandler] allows to send a given mime type file directly to a website, but so far has only been implemented by Mozilla and only supports atom/xml MIME type.

2.2.2. data URI

Data URIs [[RFC2397](#)] can be used to replace an external file in an HTML. One advantage is that it gets embedded into the HTML and removes the need of an external file, either a real file or emulated. Unfortunately, browsers treat these the same way they treat files, therefore they would still need to be downloaded or properly redirected to an application, and RegisterContentHandler would need to be implemented by browsers and QR readers before this approach can be used.

2.2.3. Custom URI scheme

A custom URI scheme for events would behave similar to e-mail [[RFC6068](#)] geo [[RFC5870](#)] and other schemes, where the resource (e-mail, geo, or in this case, the event) is properly identified and follows a specified protocol. An HTML page could publish an event using this scheme (as it would with any other link format) or a print page could embed it in an 2D-barcode. The user would have several options of handling it: opening his application of choice, or redirecting to a previously registered website [RegisterProtocolHandler]. Support for URI schemes is widespread, most Operating Systems and browsers support it and its associated APIs.

3. The v-event URI scheme

In this section we will propose a custom URI schema that could be implemented easily by any calendar application and developers alike. We will also discuss some of its special requirements and provide several examples

3.1. Syntax

The v-event URI scheme syntax is based on both iCalendar [[RFC5545](#)] and data URI scheme [[RFC2397](#)], we intend to make it trivial for people used to iCalendar syntax to implement this scheme, and also make it consistent with other existing URI formats. The basic syntax for the URI is:

```
v-event:[base64,] icalendar event
```

To be compatible with the generic URI syntax [[RFC3986](#)], the whole URI needs to follow the percent encoding escaping. The iCalendar event can be either written as an escaped text or, if base64 is specified, converted to base64. Calendar applications MUST recognize both formats to be compliant with this URI scheme For the iCalendar the following restrictions apply:

- o Exactly one entity in the VCALENDAR (VEVENT / VTODO) must be specified.
- o MUST be a valid entity as specified by [RFC 5545](#) [[RFC5545](#)], except for rules specified in this document that can violate the RFC specification
- o Start and End dates MUST contain timezone through the TZID param, as described in [section 3.8.2.2](#) and 3.8.2.4 of [RFC 5545](#) [[RFC5545](#)]
- o Timezones MUST be specified using one of the valid names from the IANA timezone database (tz) [[tz](#)]. Also, VTIMEZONE entries MUST NOT be added to the v-event URI or to the source files. All calendar applications reading events will recognize these names (see [Section 3.4.2](#) for more details)
- o The event MUST contain a UID, as specified by [section 3.8.4.7 of RFC 5545](#) [[RFC5545](#)]
- o The VCALENDAR object MAY contain a SOURCE field [[CalDavExtensions](#)], pointing to an ICS file that can contain extra information about the event contained in the calendar. If the source file and the entry contradict each other, the information presented in the source MUST prevail. If the source is available,

the event contained in the source file MUST have the same UID as the event expressed by the URI

- o The URI size MUST fit in the medium you're choosing to transmit it, For reference, URIs larger than 2048 characters are known to not work properly on all browsers, and QR codes have a hard limit of 2953 characters in its most permissive encoding. In practice, we recommend limiting the URI to 1024 characters and our tests have shown 500 characters are usually enough for most common scenarios.
- o LAST-MODIFIED field, as specified by [section 3.8.7.3 of RFC 5545](#) [[RFC5545](#)] MUST be included to allow for changes to be detected by calendar handler applications

3.2. URI Registration

The v-event URI will be registered with IANA as a provisional scheme, to allow all calendar applications to use it. The authors are not pursuing a permanent registration because they believe that this scheme may be deprecated in the future in favor of a DATA URI scheme, when browser implementations support that scheme with the same level regular URIs are supported.

The following are the fields required by [RFC 7595](#) [[RFC7595](#)]

- o Scheme name: v-event
- o Status: provisional
- o Applications/protocols that use this scheme name: Hypertext (for example, web pages, e-mail. QR code readers), calendar applications.
- o Contact: Raphael Menderico (menderico@google.com)
- o Change Controller: CalConnect [[CalConnect](#)]
- o References: this document, plus references in it

3.3. Examples

3.3.1. V-Event URI 101 - a simple example

In this first example, we will start with a simple event that follows all the recommendations above. This event starts on March 23rd, 2233, at midnight and finishes at 11:59 PM at the same day, in

Eastern Time. It has been last modified on April 1st, 2015. From these, we have the following icalendar event:

```
BEGIN:VCALENDAR
BEGIN:VEVENT
SUMMARY:James T. Kirk's birthday
DTSTART;TZID=US/Eastern:22330322T000000
DTEND;TZID=US/Eastern:22330322T235900
UID:8726bc91-a168-4c42-9568-a0e7d35724d6@example.com
LAST-MODIFIED:20150401T000000Z
END:VEVENT
END:VCALENDAR
```

which leads to the v-event URI:

```
v-event:BEGIN%3AVCALENDAR%0D%0ABEGIN%3AVEVENT%0D%0ASUMMARY
%3AJames%20T.%20Kirk%27s%20birthday%0D%0ADTSTART%3BTZID%3DUS
%2FEastern%3A22330322T000000%0D%0ADTEND%3BTZID%3DUS%2F
Eastern%3A22330322T235900%0D%0AUID%3A8726bc91-a168-4c42-
9568-a0e7d35724d6%40example.com%0D%0ALAST-MODIFIED%3A
20150401T000000Z%0D%0AEND%3AVEVENT%0D%0AEND%3AVCALENDAR
```

3.3.2. Base 64 encoding

As mentioned before, calendar applications also need to be able to interpret base64 versions of the URIs, the example below represents the same event described in [Section 3.3.1](#):

```
v-event:base64,QkVHSU46VkNBTEVOREFSdQpCRUdJTjpWRVZFTlQNC1NVTU1BU
lk6SmFtZXMgVC4gS2lyaydzIGJpcnRoZGF5DQpEVFNuQVJU01RaSUQ9VVMvRWF
zdGVybjoyMjMzMMyMlQwMDAwMDANckRURU5E01RaSUQ9VVMvRWFzdGVybjoyM
jMzMMyMlQyMzU5MDANClVJRDo4NzI2YmM5MS1hMTY4LTRjNDItOTU2OC1hMGU
3ZDM1NzI0ZDZAZXhhbXBsZS5jb20NCkxBU1QtTU9ESUZJRuQ6MjAxNTA0MDFUM
DAwMDAwWg0KRu5E01ZFVkvOvA0KRu5E01ZDQUxFTkRBUg==
```

3.3.3. Source link

A source link should be added if the URI cannot fit all information about a given event or for any other reason you believe that an ICS file may better suit your needs. For the same example in [Section 3.3.1](#), we can add the source URL 'http://www.example.com/kirk.ics' and we would obtain the following URIs:

v-

```
event:BEGIN%3AVCALENDAR%0D%0ASOURCE%3Ahttp%3A%2F%2Fwww.example.com
%2Fkirk.ics%0D%0ABEGIN%3AVEVENT%0D%0ASUMMARY%3AJames%20T.
%20Kirk%27

s%20birthday%0D%0ADTSTART%3BTZID%3DUS%2FEastern%3A22330322T000000%0D
%0ADTEND%3BTZID%3DUS%2FEastern%3A22330322T235900%0D%0AUID%3Af41cb1b
3-e071-425d-a200-5e1384a22758%40example.com%0D%0ALAST-
MODIFIED%3A
20150401T000000Z%0D%0AEND%3AVEVENT%0D%0AEND%3AVCALENDAR
```

v-

```
event:base64,QkVHSU46VknBTEVOREFSQpTT1VSQ0U6aHR0cDovL3d3dy5leGFtcGx1L
mNvbS9raXJrLmljcw0KQkVHSU46VkvVWU5UDQpTVU1NQVJZ0kphbWVzIFQuIETpcmsnc
yBiaXJ0aGRheQ0KRFRtVEFSVDtUwklEPVVTl0Vhc3Rlcm46MjIzMzAzMjJUMDAwMDAwD
QpEVEVORDtUwklEPVVTl0Vhc3Rlcm46MjIzMzAzMjJUMjM1OTAwDQpVSUQ6ZjQxY2Ix
YjMtZTA3MS00MjVklWEyMDAtNWUxMzgz0YTIyNzU4QGV4YW1wbGUuY29tDQpMQVNU
LU1PRElGSUVE0jIwMTUwNDaxVDAwMDAwMFoNckVORDpWRVZFTlQNckVORDpWQ0
FMRU5EQVI=
```

The iCal object in this case would be:

```
BEGIN:VCALENDAR
SOURCE:http://www.example.com/kirk.ics
BEGIN:VEVENT
SUMMARY:James T. Kirk's birthday
DTSTART;TZID=US/Eastern:22330322T000000
DTEND;TZID=US/Eastern:22330322T235900
UID:f41cb1b3-e071-425d-a200-5e1384a22758@example.com
LAST-MODIFIED:20150401T000000Z
END:VEVENT
END:VCALENDAR
```

3.3.4. QR code examples

A QR code containing the first example ([Section 3.3.1](#)) can be found at <https://goo.gl/lQXIwP>. It has been generated using the ZXing barcode generator([\[ZXing\]](#)).

3.4. Application requirements and best practices

3.4.1. Event publisher

For event publishers, the following extra requirements must be met:

- o If your entry contains a SOURCE field pointing to an URI, the publisher is responsible for keeping the link live and with up-to-date information while the event information is relevant (i.e, the link must exist until the event expires).

There are also some best practices that need to be followed by these publishers in regard to UID generation and the LAST-MODIFIED field, which are discussed in the following subsections.

3.4.1.1. UID generation

According to [RFC 5545](#) [[RFC5545](#)], every event MUST be published with an UID, so calendar applications can detect multiple occurrences of it and remove them. The UID MUST be a globally unique identifier, and the system generating the event must guarantee it is unique. The recommended way is to generate an id that is internal to a given system (for instance, a database incremental id, an UUID, or something similar) and append the domain name or IP address at the end, separating them by an @.

For example, all these are valid unique ids for domain example.com that fit this recommendation and also [RFC 5545](#) [[RFC5545](#)]:

- o [1@example.com]: a simple numerical id, useful if you are creating your first event and has no intention to create another or can manage the ids manually.
- o [user-29960401T080000Z-1@example.com]: An UID for an event from user 'user' that starts April first, 2996, at 8:00 AM, and uses the username and date as keys.
- o [f47935ee-ec5e-4d87-ba26-05e970674a88@example.com]: a UID which uses UUIDs based on [RFC 4122](#) [[RFC4122](#)]. Theoretically, UUIDs are themselves unique, but to conform with the recommendation we also appended the domain name.

3.4.2. Calendar applications requirements

For calendar data handlers, the following set of extra rules apply:

- o An event MUST only be handled by a calendar application after an user performed an action, such as clicking on a link or scanning a QR code. Events published using the URI SHOULD NOT be added automatically.
- o Calendar data handlers MUST retain sufficient information to determine that an event has changed so that it can inform the user.
- o If a user deletes a previously downloaded event the handler should recognize that and ignore the event unless explicitly clicked on.
- o A calendar application must keep its timezone database always up-to-date and adjust events accordingly. Timezones will be specified by reference (i.e., their ISO names, according to [[tz](#)]) and any calendar application MUST understand these.

4. Security and privacy considerations

Below are some guidelines applications implementing v-event URI generators and parsers need to follow in order to avoid security and privacy issues.

- o Whenever a SOURCE link is available, the application MUST ask the user whether to follow the link, since there may be costs associated with downloading data and the user may want to perform this operation in a different environment.
- o Calendar applications MAY check a SOURCE link periodically to check for changes, but MUST NOT update an event automatically based on new information provided by the user. If new information is available through the SOURCE link, calendar applications SHOULD inform the user and ask for his consent before performing any change in his calendar.
- o Reading an v-event URI or following a SOURCE link and downloading a file may pose a security thread if not carefully handled. Particularly code reading these files should be careful to not get exposed to common security bugs like buffer overflows.
- o A SOURCE link SHOULD not be used only as a tracking mechanism, if a link is provided there should be some extra information being provided by it or at least the possibility that the information will be updated if necessary
- o A SOURCE link MUST not require a calendar account in any calendar manager, and MUST NOT represent any form of event subscription by a particular system. Any event subscription action REQUIRES user acknowledgement and approval before being performed.
- o Note that there is no hard limit on the size of a SOURCE file, but it is expected that these contain information only about a single event (i.e., one VEVENT) or recurring event (several VEVENTs with the same RECURRENCE-ID) This has implications for both writers and readers of these source files:
 - * Writers MUST always provided well-formed data that complies to this document and, more generally, to iCalendar format [[RFC5545](#)].
 - * Readers can't rely on the size of an input to decide whether it is valid or not, and SHOULD implement parsers that detect inconsistencies.

5. Future work

As mentioned in section [Section 2.2.2](#), the data URI scheme would be a nice fit for providing an uniform format for specifying events in the Web and printed media (QR and other formats), and we have only chosen another method because data support is currently limited.

We plan to update this document with a data URI compatible format as soon as its support is more widespread, allowing it to be used by native applications, browser applications and physical media with the same support currently available for regular URIs. The format specified here is compatible with data URI and minimal changes would be needed to convert from one format to another.

6. Acknowledgements

The authors would like to thank the members of CalConnect TC-EVENTPUB committee for its contributions to this document, particularly Dave Thewlis, Cyrus Daboo and Thomas Schaefer.

7. References

[AddThisEvent]

AddThisEvent, "AddThisEvent", 2012, <<http://addthisevent.com>>.

Last checked in August 26, 2015

[CalConnect]

CalConnect, "CalConnect: The Calendaring and Scheduling Consortium", January 2004, <<http://calconnect.org>>.

Last checked in November 1, 2015

[CalendarContract]

Google Inc., "Android Calendar Contract", October 2011, <<http://developer.android.com/reference/android/provider/CalendarContract.html>>.

Last checked in August 26, 2015

[CalDavExtensions]

Daboo, C., "New Properties for iCalendar", April 2015, <<https://tools.ietf.org/html/draft-ietf-calext-extensions-00>>.

Last checked in August 26, 2015

[GoogleCalendar]

Google Inc., "Google Calendar", April 2006, <
<http://calendar.google.com>>.

Last checked in August 26, 2015

[hCalendar]

Celik, T. and B. Suda, "hCalendar Microformat", June 2005,
<<http://microformats.org/wiki/hcalendar>>.

Last checked in July 27, 2015

[i-nigma]

3GVision, "i-nigma", August 2015.

Last checked in August 27, 2015

[RFC2397]

Masinter, L., "The "data" URL scheme", [RFC 2397](#),
DOI 10.17487/RFC2397, August 1998.

[RFC3986]

Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
Resource Identifier (URI): Generic Syntax", STD 66,
[RFC 3986](#), DOI 10.17487/RFC3986, January 2005,
<<http://www.rfc-editor.org/info/rfc3986>>.

[RFC4122]

Leach, P., Mealling, M., and R. Salz, "A Universally
Unique IDentifier (UUID) URN Namespace", [RFC 4122](#),
DOI 10.17487/RFC4122, July 2005,
<<http://www.rfc-editor.org/info/rfc4122>>.

[RFC5545]

Desruisseaux, B., Ed., "Internet Calendaring and
Scheduling Core Object Specification (iCalendar)",
[RFC 5545](#), DOI 10.17487/RFC5545, September 2009,
<<http://www.rfc-editor.org/info/rfc5545>>.

[RFC5870]

Mayrhofer, A. and C. Spanring, "A Uniform Resource
Identifier for Geographic Locations ('geo' URI)",
[RFC 5870](#), DOI 10.17487/RFC5870, June 2010,
<<http://www.rfc-editor.org/info/rfc5870>>.

[RFC6068]

Duerst, M., Masinter, L., and J. Zawinski, "The 'mailto'
URI Scheme", [RFC 6068](#), DOI 10.17487/RFC6068, October 2010.

[RFC6838]

Freed, N., Klensin, J., and T. Hansen, "Media Type
Specifications and Registration Procedures", [BCP 13](#),
[RFC 6838](#), DOI 10.17487/RFC6838, January 2013,
<<http://www.rfc-editor.org/info/rfc6838>>.

[tz] IANA, "Time Zone Database", 1986, <<https://www.iana.org/time-zones>>.

Last checked in August 27, 2015

[RFC7595] Thaler, D., Ed., Hansen, T., and T. Hardie, "Guidelines and Registration Procedures for URI Schemes", [BCP 35](#), [RFC 7595](#), DOI 10.17487/RFC7595, June 2015, <<http://www.rfc-editor.org/info/rfc7595>>.

[ZXing] Owen, S., "ZXing Project", November 2007.

Last checked in July 27, 2015

Authors' Addresses

Raphael Menderico (editor)
Google Inc.
Brandschenkestrasse 110
Zurich, ZH 8002
CH

Email: menderico@google.com
URI: <http://www.google.com/>

Paulo Schlup
Google Inc.
Brandschenkestrasse 110
Zurich, ZH 8002
CH

Email: pschlup@google.com
URI: <http://www.google.com/>

Lucia Kristiansen
Google Inc.
Brandschenkestrasse 110
Zurich, ZH 8002
CH

Email: lucka@google.com
URI: <http://www.google.com/>

