

VALARM Extensions for iCalendar

Working 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.

© 2024-07-23 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

Abstract.....	iv
Introduction.....	v
1. Scope.....	1
2. Normative references.....	1
3. Terms and definitions.....	1
4. Conventions.....	1
5. Extensible syntax for VALARM.....	1
6. Alarm Unique Identifier.....	3
7. Alarm Related To.....	3
8. Alarm Acknowledgement.....	4
8.1. Acknowledged Property.....	4
9. Snoozing Alarms.....	5
9.1. Relationship Type Property Parameter.....	6
10. Alarm Proximity Trigger.....	6
10.1.Proximity Property.....	6
10.2.Example.....	7
11. Security Considerations.....	7
12. Privacy Considerations.....	8
13. IANA Considerations.....	8
13.1.Property Registrations.....	8
13.2.Relationship Types Registry.....	8
13.3.Proximity Value Registry.....	8
14. Acknowledgments.....	8
Bibliography.....	9

Abstract

This document defines a set of extensions to the iCalendar VALARM component to enhance use of alarms and improve interoperability between clients and servers.

Introduction

The iCalendar [IETF RFC 5545](#) specification defines a set of components used to describe calendar data. One of those is the “VALARM” component which appears as a sub-component of “VEVENT” and “VTODO” components. The “VALARM” component is used to specify a reminder for an event or task. Different alarm actions are possible, as are different ways to specify how the alarm is triggered.

As iCalendar has become more widely used and as client-server protocols such as CalDAV [IETF RFC 4791](#) have become more popular, several issues with “VALARM” components have arisen. Most of these relate to the need to extend the existing “VALARM” component with new properties and behaviors to allow clients and servers to accomplish specific tasks in an interoperable manner. For example, clients typically need a way to specify that an alarm has been dismissed by a calendar user, or has been “snoozed” by a set amount of time. To date, this has been done through the use of custom “X-” properties specific to each client implementation, leading to poor interoperability.

This specification defines a set of extensions to “VALARM” components to cover common requirements for alarms not currently addressed in iCalendar. Each extension is defined in a separate section below. For the most part, each extension can be supported independently of the others, though in some cases one extension will require another. In addition, this specification describes mechanisms by which clients can interoperably implement common features such as “snoozing”.

VALARM Extensions for iCalendar

1. Scope

This document defines a set of extensions to the iCalendar VALARM component to enhance use of alarms and improve interoperability between clients and servers.

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

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

IETF RFC 5545, Internet Engineering Task Force (committee). *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. 2009. RFC Publisher. <https://www.rfc-editor.org/info/rfc5545>.

IETF RFC 5870, MAYRHOFER, A., C. SPANRING and Internet Engineering Task Force. *A Uniform Resource Identifier for Geographic Locations ('geo' URI)*. 2010. RFC Publisher. <https://www.rfc-editor.org/info/rfc5870>.

IETF RFC 7986, DABOO, C. and Internet Engineering Task Force. *New Properties for iCalendar*. 2016. RFC Publisher. <https://www.rfc-editor.org/info/rfc7986>.

IETF RFC 8174, LEIBA, B. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. 2017. RFC Publisher. <https://www.rfc-editor.org/info/rfc8174>.

Internet-Draft draft-ietf-calext-eventpub-extensions-00, DOUGLASS, Michael. *Event Publishing Extensions to iCalendar*. 2016. <https://datatracker.ietf.org/doc/html/draft-ietf-calext-eventpub-extensions-00>.

3. Terms and definitions

No terms and definitions are listed in this document.

4. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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.

5. Extensible syntax for VALARM

[IETF RFC 5545, Section 3.6.6](#) defines the syntax for "VALARM" components and properties within them. However, as written, it is hard to extend this by adding, e.g., a new property common to all types of alarm. Since many of the extensions defined in this document need to extend the

base syntax, an alternative form for the base syntax is defined here, with the goal of simplifying specification of the extensions.

A "VALARM" calendar component is re-defined by the following notation:

```
alarmcext = "BEGIN" ":" "VALARM" CRLF
           alarmprop
           "END" ":" "VALARM" CRLF

alarmprop = *(
    ; the following are REQUIRED,
    ; but MUST NOT occur more than once

    action / trigger /

    ; one set of action properties MUST be
    ; present and MUST match the action specified
    ; in the ACTION property

    actionprops /

    ; the following is OPTIONAL,
    ; and MAY occur more than once

    x-prop / iana-prop
)

actionprops = audiopropext / disppropext / emailpropext

audiopropext = *(
    ; 'duration' and 'repeat' are both OPTIONAL,
    ; and MUST NOT occur more than once each,
    ; but if one occurs, so MUST the other

    duration / repeat /

    ; the following is OPTIONAL,
    ; but MUST NOT occur more than once

    attach
)

disppropext = *(
    ; the following are REQUIRED,
    ; but MUST NOT occur more than once

    description /

    ; 'duration' and 'repeat' are both OPTIONAL,
    ; and MUST NOT occur more than once each,
    ; but if one occurs, so MUST the other

    duration / repeat
)
```



```
emailpropext = *(
    ; the following are all REQUIRED,
    ; but MUST NOT occur more than once

    description / summary /

    ; the following is REQUIRED,
    ; and MAY occur more than once

    attendee /

    ; 'duration' and 'repeat' are both OPTIONAL,
    ; and MUST NOT occur more than once each,
    ; but if one occurs, so MUST the other

    duration / repeat

)
```

6. Alarm Unique Identifier

This extension adds a “UID” property to “VALARM” components to allow a unique identifier to be specified. The value of this property can then be used to refer uniquely to the “VALARM” component.

The “UID” property defined here follows the definition in [IETF RFC 5545, Section 3.8.4.7](#) with the security and privacy updates in [IETF RFC 7986, Section 5.3](#). In particular it MUST be a globally unique identifier that does not contain any security- or privacy-sensitive information.

The “VALARM” component defined in [Clause 5](#) is extended here as:

```
alarmprop /= *(
    ; the following is OPTIONAL,
    ; but MUST NOT occur more than once

    uid

)
```

7. Alarm Related To

It is often convenient to relate one or more “VALARM” components to other “VALARM” components (e.g., see [Clause 9](#)). This can be accomplished if the “VALARM” components each have their own “UID” property (as per [Clause 6](#)).

This specification updates the usage of the “RELATED-TO” property defined in [IETF RFC 5545, Section 3.8.4.5](#) to enable its use with “VALARM” components. Specific types of relationships between “VALARM” components can be identified by registering new values for the “RELTYPE” property parameter defined in [IETF RFC 5545, Section 3.2.15](#).

The “VALARM” component defined in [Clause 5](#) is extended here as:

```
alarmprop /= *(
    ; the following is OPTIONAL,
```

```

; but MAY occur more than once

related

)

```

8. Alarm Acknowledgement

There is currently no way for a “VALARM” component to indicate whether it has been triggered and acknowledged. With the advent of a standard client/server protocol for calendaring and scheduling data ([IETF RFC 4791](#)) it is quite possible for an event with an alarm to exist on multiple clients in addition to the server. If each of those is responsible for performing the action when an alarm triggers, then multiple “alerts” are generated by different devices. In such a situation, a calendar user would like to be able to “dismiss” the alarm on one device and have it automatically dismissed on the others too.

Also, with recurring events that have alarms, it is important to know when the last alarm in the recurring set was acknowledged, so that the client can determine whether past alarms have been missed.

To address these needs, this specification adds an “ACKNOWLEDGED” property to “VALARM” components to indicate when the alarm was last sent or acknowledged. This is defined by the syntax below.

```

alarmprop      /= *(
                ; the following is OPTIONAL,
                ; but MUST NOT occur more than once

                acknowledged

                )

```

8.1. Acknowledged Property

Property Name	ACKNOWLEDGED
Purpose	This property specifies the UTC date and time at which the corresponding alarm was last sent or acknowledged.
Value Type	DATE-TIME
Property Parameters	IANA and non-standard property parameters can be specified on this property.
Conformance	This property can be specified within “VALARM” calendar components.
Description	This property is used to specify when an alarm was last sent or acknowledged. This allows clients to determine when a pending alarm has been acknowledged by a calendar user so that any alerts can be dismissed across multiple devices. It also allows clients to track repeating alarms or alarms on recurring events or to-dos to ensure that the right number of missed alarms can be tracked. Clients SHOULD set this property to the current date-time value in UTC when a calendar user acknowledges a pending alarm. Certain kinds of alarm may not provide feedback as to when the calendar user sees them, for example email based alerts. For those kinds of alarms, the client

SHOULD set this property when the alarm is triggered and the action successfully carried out.

When an alarm is triggered on a client, clients can check to see if an "ACKNOWLEDGED" property is present. If it is, and the value of that property is greater than or equal to the computed trigger time for the alarm, then the client SHOULD NOT trigger the alarm. Similarly, if an alarm has been triggered and an "alert" presented to a calendar user, clients can monitor the iCalendar data to determine whether an "ACKNOWLEDGED" property is added or changed in the alarm component. If the value of any "ACKNOWLEDGED" property in the alarm changes and is greater than or equal to the trigger time of the alarm, then clients SHOULD dismiss or cancel any "alert" presented to the calendar user.

Format
Definition

This property is defined by the following notation:

```
acknowledged = "ACKNOWLEDGED" acknowledgedparam ":"
datetime CRLF

acknowledgedparam = *(
    ; the following is OPTIONAL,
    ; and MAY occur more than once

    (";" other-param)

)
```

Example

The following is an example of this property:

```
ACKNOWLEDGED:20090604T084500Z
```

9. Snoozing Alarms

Users often want to "snooze" an alarm, and this specification defines a standard approach to accomplish that.

To "snooze" an alarm, clients create a new "VALARM" component within the parent component of the "VALARM" that was triggered and is being "snoozed" (i.e., as a "sibling" component of the "VALARM" being snoozed). The new "VALARM" MUST be set to trigger at the user's chosen "snooze" interval after the original alarm triggered. Clients SHOULD use an absolute "TRIGGER" property with a "DATE-TIME" value specified in UTC.

Clients SHOULD add a "RELATED-TO" property to the new "VALARM" component with a value set to the "UID" property value of the "VALARM" component being snoozed. If the "VALARM" component being snoozed does not already have a "UID" property, the client SHOULD add one. The "RELATED-TO" property added to the new "VALARM" component SHOULD include a "RELTYPE" property parameter with a value set to "SNOOZE".

When the "snooze" alarm is triggered and dismissed the client SHOULD remove the corresponding "VALARM" component, or set the "ACKNOWLEDGED" property (see [Clause 8](#)). Alternatively, if the "snooze" alarm is itself "snoozed", the client SHOULD remove the original "snooze" alarm and create a new one, with the appropriate trigger time and relationship set.

9.1. Relationship Type Property Parameter

This specification adds the “SNOOZE” relationship type for use with the “RELTYPE” property defined in [IETF RFC 5545, Section 3.2.15](#). This is used to relate a “snoozed” “VALARM” component to the original alarm that the “snooze” was generated for.

10. Alarm Proximity Trigger

VALARMS are currently triggered when a specific date-time is reached. It is also desirable to be able to trigger alarms based on location, e.g. when arriving at or departing from a particular location.

This specification adds the following properties to “VALARM” components to indicate when an alarm can be triggered based on location.

- “PROXIMITY” — indicates that a location based trigger is to be used and which direction of motion is used for the trigger
- “STRUCTURED-LOCATION” — used to indicate the actual location to trigger off, specified using a geo: URI [IETF RFC 5870](#) which allows for two or three coordinate values with an optional uncertainty

```
alarmprop      /= *(
                ; the following is OPTIONAL,
                ; but MUST NOT occur more than once

                proximity /

                ; the following is OPTIONAL,
                ; and MAY occur more than once, but only
                ; when a PROXIMITY property is also present

                structured-location

                )
```

Typically, when a “PROXIMITY” property is used there is no need to specify a time-based trigger using the “TRIGGER” property. However, since “TRIGGER” is defined as a required property for a “VALARM” component, for backwards compatibility it has to be present, but ignored. To indicate a “TRIGGER” that is to be ignored, clients SHOULD use a value a long time in the past. A value of “19760401T005545Z” has been commonly used for this purpose.

10.1. Proximity Property

Property Name	PROXIMITY
Purpose	This property indicates that a location based trigger is applied to an alarm.
Value Type	TEXT
Property Parameters	IANA and non-standard property parameters can be specified on this property.
Conformance	This property can be specified within “VALARM” calendar components.
Description	This property is used to indicate that an alarm has a location-based trigger. Its value identifies the direction of motion used to trigger

the alarm. One or more location values are set using “STRUCTURED-LOCATION” properties.

When the property value is set to “ARRIVE”, the alarm is triggered when the calendar user agent arrives in the vicinity of any of the specified locations. When set to “DEPART”, the alarm is triggered when the calendar user agent departs from the vicinity of any specified locations.

When the property value is set to “CONNECT”, the alarm is triggered when the calendar user agent connects to a [Bluetooth®](#)-enabled automobile. When set to “DISCONNECT”, the alarm is triggered when the calendar user agent disconnects from a Bluetooth®-enabled automobile.

**Format
Definition**

This property is defined by the following notation:

```
proximity = "PROXIMITY" proximityparam ":" proximityvalue
CRLF
```

```
proximityparam = *(
    ; the following is OPTIONAL,
    ; and MAY occur more than once
    (";" other-param)
)
```

```
proximityvalue = "ARRIVE" / "DEPART" /
    "CONNECT" / "DISCONNECT" / iana-token /
    x-name
```

Example

The following is an example of this property:

```
PROXIMITY:ARRIVE
```

10.2. Example

The following example shows a “VALARM” component with a proximity trigger set to trigger when the device running the calendar user agent leaves the vicinity defined by the structured location property. Note use of the “u=” parameter with the “geo” URI to define the precision of the location determination.

```
BEGIN:VALARM
UID:77D80D14-906B-4257-963F-85B1E734DBB6
TRIGGER;VALUE=DATE-TIME:19760401T005545Z
ACTION:DISPLAY
DESCRIPTION:Remember to buy milk
TRIGGER;VALUE=DATE-TIME:19760401T005545Z
PROXIMITY:DEPART
STRUCTURED-LOCATION;VALUE=URI:geo:40.443,-79.945;u=10
END:VALARM
```

11. Security Considerations

VALARMS, if not monitored properly, can be used to “spam” users and/or leak personal information. For instance, an unwanted audio or display alert could be considered spam. Or an email alert could be used to leak a user’s location to a third party or to send unsolicited email to multiple users. Therefore, CalDAV clients and servers that accept iCalendar data from a third party

(e.g. via [iTIP](#), a subscription feed, or a shared calendar) SHOULD remove all VALARMS from the data prior to storing in their calendar system.

12. Privacy Considerations

Proximity VALARMS, if not used carefully, can leak a user's past, present, or future location. For instance, storing an iCalendar resource containing proximity VALARMS to a shared calendar on CalDAV server can expose to anyone that has access to that calendar the user's intent to leave from or arrive at a particular location at some future time. Furthermore, if a CalDAV client updates the shared iCalendar resource with an ACKNOWLEDGED property when the alarm is triggered, will leak the exact date and time that the user left from or arrived at the location.

Therefore, CalDAV clients that implement proximity alarms SHOULD give users the option of storing and/or acknowledging the alarms on the local device only and not storing the alarm and/or acknowledgment on a remote server.

13. IANA Considerations

13.1. Property Registrations

This document defines the following new iCalendar properties to be added to the registry defined in [IETF RFC 5545, Section 8.2.3](#):

Table 1

Property	Status	Reference
ACKNOWLEDGED	Current	RFCXXXX, Clause 8
PROXIMITY	Current	RFCXXXX, Clause 10.1

13.2. Relationship Types Registry

This document defines the following new iCalendar relationship type to be added to the registry defined in [IETF RFC 5545, Section 8.3.8](#):

Table 2

Relationship Type	Status	Reference
SNOOZE	Current	RFCXXXX, Clause 9.1

13.3. Proximity Value Registry

This document creates a new iCalendar registry for values of the "PROXIMITY" property:

Table 3

Value	Status	Reference
ARRIVE	Current	RFCXXXX, Clause 10.1
DEPART	Current	RFCXXXX, Clause 10.1
CONNECT	Current	RFCXXXX, Clause 10.1
DISCONNECT	Current	RFCXXXX, Clause 10.1

14. Acknowledgments

This specification came about via discussions at the Calendaring and Scheduling Consortium. Also, thanks to the following for providing feedback: Bernard Desruisseaux, Mike Douglass, Jacob Farkas, Jeffrey Harris, and Ciny Joy.

Bibliography

- [1] IETF RFC 5546, Internet Engineering Task Force (committee). *iCalendar Transport-Independent Interoperability Protocol (iTIP)*. 2009. RFC Publisher. <https://www.rfc-editor.org/info/rfc5546>.
- [2] BTcore, Bluetooth Special Interest Group, "Bluetooth Core Specification Version 5.0", December 2016, <https://www.bluetooth.com/specifications/bluetooth-core-specification>