

Network Working Group	C. Daboo
Internet-Draft	Apple
Intended status: Standards Track	May 03, 2011
Expires: November 04, 2011	

VALARM Extensions for iCalendar
draft-daboo-valarm-extensions-03

[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.

[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 November 04, 2011.

[Copyright Notice](#)

Copyright (c) 2011 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.

[Table of Contents](#)

- *1. [Introduction](#)
- *2. [Conventions Used in This Document](#)
- *3. [Extensible syntax for VALARM](#)
- *4. [VALARM Unique Identifier](#)

- *5. [VALARM URI Action](#)
- *6. [VALARM Agent Property](#)
 - *6.1. [Alarm Agent Property](#)
 - *6.2. [ID Property Parameter](#)
 - *6.3. [Examples](#)
- *7. [VALARM Acknowledged Property](#)
 - *7.1. [Acknowledged Property](#)
 - *7.2. [Examples](#)
- *8. [Snoozing Alarms](#)
- *9. [VALARM Proximity Trigger Property](#)
 - *9.1. [Proximity Property](#)
 - *9.2. [Geo Location Property](#)
 - *9.3. [Examples](#)
- *10. [Default Alarms in CalDAV](#)
- *11. [Security Considerations](#)
- *12. [IANA Considerations](#)
 - *12.1. [Property Registrations](#)
 - *12.2. [Parameter Registrations](#)
 - *12.3. [Actions Registry](#)
- *13. [Acknowledgments](#)
- *14. [References](#)
 - *14.1. [Normative References](#)
 - *14.2. [Informative References](#)
- *Appendix A. [Change History \(To be removed by RFC Editor before publication\)](#)
- *[Author's Address](#)

1. Introduction

The [iCalendar](#) [RFC5545] 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 to-do. 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](#) [RFC4791] 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".

2. Conventions Used in This Document

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 [\[RFC2119\]](#).

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.

3. Extensible syntax for VALARM

Section 3.6.6 of [\[RFC5545\]](#) 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.

```
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

    )

```

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

4. VALARM Unique Identifier

Several of the other extensions in this specification require identifying a specific instance of a "VALARM" component in an iCalendar stream. To aid that, 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 Section 3.8.4.7 of [\[RFC5545\]](#). In particular it MUST be a globally unique identifier.

```

alarmprop /= *(

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

uid

    )

```

The "VALARM" component defined in [Section 3](#) is extended here as:

5. VALARM URI Action

Currently "VALARM" components have actions for audio, display and email. New types of action are of interest, e.g., SMS, instant messaging, etc. Rather than specify separate actions for these, an alternative is to define a "URI" action that allows any URI scheme to be used as an action, where it makes sense. Thus URI schemes for IM [[RFC3860](#)], SIP [[RFC3261](#)], TEL [[RFC3966](#)] etc could be used.

```
actionvalue /= "URI"
              ; Adds a new action for a "VALARM"

actionprop  /= uriprop
              ; Re-defines the "VALARM" component to include
              ; "uriprop" as a new set of action properties

uriprop     = *(
              ; the following is REQUIRED,
              ; and MUST occur only once

              uri /

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

              duration / repeat

              )
```

This extension defines a new "URI" property value for use with the "ACTION" property in "VALARM" components. A new set of action properties is defined for "VALARM" components based on this new action as defined by the syntax below.

6. VALARM Agent Property

With the advent of a standard client/server protocol for calendaring and scheduling data ([RFC4791](#)) there is a need to specify which client or server should handle the presentation of an alarm when it is triggered. For example, calendar users want to be able to receive alarms at all times, even when their desktop client might be unavailable. Since the server is "always on", a service running on the server could monitor alarm status and, when appropriate, trigger those alarms. In addition it may be important for only the server or the client to be set to handle an alarm - and in some cases only specific servers or clients.

```

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

    alarm-agent

)

```

To address this need, this specification defines an "ALARM-AGENT" iCalendar property that can be added to any "VALARM" component. This property specifies whether a client or server or both should be responsible for processing the alarm when it triggers. In addition, a "ID" property parameter can be added to uniquely identify the client or server that should process the alarm. This is defined by the syntax below.

6.1. Alarm Agent Property

```

alarm-agent = "ALARM-AGENT" alarmagentparam ":"
             alarmagentvalue CRLF

```

```

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

    (";" idparam) /

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

    (";" other-param)

)

```

```

alarmagentvalue = "SERVER" / "CLIENT" / "BOTH" / "NONE" /
                 iana-token / x-name

```

ALARM-AGENT:SERVER

ALARM-AGENT;ID="tag:example.com,2011:cyrus-desktop":CLIENT

Property Name:

ALARM-AGENT

Purpose: This property specifies whether a client, server, both or none gets to process an alarm when it is triggered.

Value Type: TEXT

Property Parameters: IANA, non-standard, and id 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 who is responsible for processing an alarm when it is triggered. When the value "SERVER" is specified, only a server that has a copy of the event is responsible. When the value "CLIENT" is specified, only a client that has a copy of the event is responsible. When the value "BOTH" is specified, either a client or server is responsible. When the value "NONE" is specified, neither a client nor server is responsible (i.e., the alarm action is never carried out when it triggers).

If the "ID" property parameter is specified for "SERVER" or "CLIENT" values, then only the client or server identified by the "ID" value is responsible. Clients or servers that are not responsible for the alarm SHOULD NOT process the alarm action when it is triggered. If multiple clients or servers need to be specified, then the "ALARM-AGENT" property should be included multiple times in the "VALARM" component, with each one containing the appropriate "ID" property parameter value to identify each client or server.

In the absence of this property clients and servers can choose to process the alarm themselves as appropriate. i.e., a default value of "BOTH".

This property MAY occur more than once to allow multiple processors of an alarm.

Format Definition: This property is defined by the following notation:

Example: The following are examples of this property:

[6.2. ID Property Parameter](#)

idparam = "ID" "=" DQUOTE uri DQUOTE

ALARM-AGENT;ID="http://calendar.example.com":SERVER

Parameter Name:

ID

Purpose: This property parameter specifies a URI identifier for the property it is applied to.

Format Definition: This property parameter is defined by the following notation:

Description: This property parameter is used to specify a URI identifier that is associated with the property it is applied to. Each property that allows this parameter to be specified MUST indicate what the value of the URI represents.

Example: The following is an example of this property parameter:

[6.3. Examples](#)

TODO Example of VALARM components

[7. VALARM Acknowledged Property](#)

There is currently no way for a "VALARM" component to indicate whether it has been triggered and acknowledged or not. With the advent of a standard client/server protocol for calendaring and scheduling data ([\[RFC4791\]](#)) 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 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.

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

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.

[7.1. Acknowledged Property](#)

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

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

```
ACKNOWLEDGED:20090604T084500Z
```

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.

Format Definition: This property is defined by the following notation:

Example: The following is an example of this property:

[7.2. Examples](#)

TODO Example of VALARM components

8. Snoozing Alarms

Users often want to "snooze" an alarm. That process effectively creates a new alarm to be triggered a set amount of time after the original being "snoozed". To accomplish this, clients SHOULD create a new "VALARM" component within the same component of the "VALARM" that was triggered and is being "snoozed". The new "VALARM" MUST be set to trigger at the user's chosen "snooze" interval after the original alarm triggered. To accomplish that clients SHOULD use an absolute "TRIGGER" property with a "DATE-TIME" value specified in UTC. When the "snooze" alarm is triggered and dismissed the client SHOULD remove it from its parent component. 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 set.

9. VALARM Proximity Trigger Property

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.

```
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  
  
                geolocation  
  
                )
```

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

*"GEO-LOCATION" - used to indicate the actual location to trigger off, specified using a [geo: URI \[RFC5870\]](#) which allows for two or three co-ordinate values with an optional uncertainty

9.1. Proximity Property

proximity = "PROXIMITY" proximityparam ":" proximityvalue CRLF

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

proximityvalue = "ARRIVE" / "DEPART" / iana-token / x-name

PROXIMITY:ARRIVE

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 "GEO-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.

The time-based "TRIGGER" property MUST also be present in the "VALARM" calendar component and MUST be set to a positive duration value (or zero duration). That value indicates a time delay to be applied to the triggering of the alarm after the location trigger is triggered. e.g., an alarm could be set to trigger 30 minutes after arriving home.

Format Definition: This property is defined by the following notation:

Example: The following is an example of this property:

9.2. Geo Location Property

geolocation = "GEO-LOCATION" geolocationparam ":" uri CRLF

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

    (";" other-param)

)
```

GEO-LOCATION:geo:40.443,-79.945;u=10

Property Name: GEO-LOCATION

Purpose: This property specifies a [geo: URI \[RFC5870\]](#) value.

Value Type: URI

Property Parameters: IANA and non-standard property parameters can be specified on this property.

Conformance: This property can be specified multiple times within "VALARM" calendar components, but only when a "PROXIMITY" property is also present.

Description: This property is used to specify a precise geo-location that location aware calendar user agents can use to trigger an alarm. The location is set using a [geo: URI \[RFC5870\]](#) which can include a level of uncertainty (e.g., it can specify the "vicinity" of a specific location). Calendar user agents MAY apply their own tolerance to the specified location based on their current location value precision.

Format Definition: This property is defined by the following notation:

Example: The following is an example of this property:

9.3. Examples

TODO Example of VALARM components

10. Default Alarms in CalDAV

Users often want to have a default alarm applied to new events that they create or to new invitations that arrive on a [CalDAV \[RFC4791\]](#) server. Since this behavior is expected to occur no matter which client a user is using, or whether any client is even connected at the time,

it is beneficial if the server itself is responsible for managing the creation of the default alarm.

This specification defines four new WebDAV properties that can be used to specify different sets of default alarms. Clients can store "VALARM" components in these properties to setup the defaults. When a new event or todo is created on the server, the server will automatically add the default, as appropriate. When a new event or todo invitation is delivered to the calendar user, the server will automatically add the default alarm, as appropriate.

The four new WebDAV properties are:

CALDAV:default-alarm-vevent-datetime A default alarm applied to "VEVENT" components whose "DTSTART" property value type is "DATE-TIME"

CALDAV:default-alarm-vevent-date A default alarm applied to "VEVENT" components whose "DTSTART" property value type is "DATE"

CALDAV:default-alarm-vtodo-datetime A default alarm applied to "VTODO" components whose "DUE" or "DTSTART" property value type is "DATE-TIME"

CALDAV:default-alarm-vtodo-date A default alarm applied to "VTODO" components whose "DUE" or "DTSTART" property value type is "DATE", or when neither of those properties is present

The WebDAV properties are defined on a calendar user's "calendar home" collection, or on individual calendar collections. When events or tasks are created in a calendar, the server will first inspect the WebDAV properties on the calendar collection to see if the appropriate property is present. If it is, the server will use the value of that property to set the default in the new component. If the property is not present on the calendar collection, the server will determine whether the property is present on the calendar home collection that contains the calendar collection. If present on the calendar home, the property value there is used as the default.

The WebDAV property value MUST be one or more "VALARM" components or the empty string. If an empty string is used, no default alarms are applied. Note that these values contain just the "BEGIN:VALARM...END:VALARM" data (i.e., syntactically the "alarmcext" element).

If a "DESCRIPTION" property is present in the "VALARM" component, but its value is empty, the server SHOULD insert the value from any "SUMMARY" property of the component in which the default "VALARM" is being added.

11. Security Considerations

TODO:talk about importance of stripping VALARMS from incoming iTIP. Talk about VALARMS being used to "spam" - particularly nasty if the server handles it.

12. IANA Considerations

12.1. Property Registrations

This document defines the following new iCalendar properties to be added to the registry defined in Section 8.2.3 of [\[RFC5545\]](#):

Property	Status	Reference
ALARM-AGENT	Current	RFCXXXX, Section 6.1
ACKNOWLEDGED	Current	RFCXXXX, Section 7.1
PROXIMITY	Current	RFCXXXX, Section 9.1
GEO-LOCATION	Current	RFCXXXX, Section 9.2

12.2. Parameter Registrations

This document defines the following new iCalendar property parameters to be added to the registry defined in Section 8.2.4 of [\[RFC5545\]](#):

Property Parameter	Status	Reference
ID	Current	RFCXXXX, Section 6.2

12.3. Actions Registry

This document defines the following new iCalendar action to be added to the registry defined in Section 8.3.10 of [\[RFC5545\]](#):

Action	Status	Reference
URI	Current	RFCXXXX, Section 5

13. 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, Jeffrey Harris, and Ciny Joy.

14. References

14.1. Normative References

[RFC2119]

	Bradner, S. , " Key words for use in RFCs to Indicate Requirement Levels ", BCP 14, RFC 2119, March 1997.
[RFC4791]	Daboo, C. , Desruisseaux, B. and L.M. Dusseault , " Calendaring Extensions to WebDAV (CalDAV) ", RFC 4791, March 2007.
[RFC5545]	Desruisseaux, B. , " Internet Calendaring and Scheduling Core Object Specification (iCalendar) ", RFC 5545, September 2009.
[RFC5870]	Mayrhofer, A. and C. Spanring , " A Uniform Resource Identifier for Geographic Locations ('geo' URI) ", RFC 5870, June 2010.

14.2. Informative References

[RFC3261]	Rosenberg, J. , Schulzrinne, H. , Camarillo, G. , Johnston, A. , Peterson, J. , Sparks, R. , Handley, M. and E. Schooler , " SIP: Session Initiation Protocol ", RFC 3261, June 2002.
[RFC3860]	Peterson, J. , " Common Profile for Instant Messaging (CPIM) ", RFC 3860, August 2004.
[RFC3966]	Schulzrinne, H. , " The tel URI for Telephone Numbers ", RFC 3966, December 2004.

Appendix A. Change History (To be removed by RFC Editor before publication)

Changes in -02:

1. Updated to 5545 reference.
2. Clarified use of absolute trigger in UTC in snooze alarms
3. Snooze alarms should be removed when completed
4. Removed status and replaced last-triggered by acknowledged property
5. Added location-based trigger
6. IANA registry tables added

Changes in -01:

1. Removed DESCRIPTION as an allowed property in the URI alarm.
2. Added statement about what to do when ALARM-AGENT is not present.
3. Allow multiple ALARM-AGENT properties to be present.

4. Removed SNOOZE-UNTIL - snoozing now accomplished by creating a new VALARM.
5. Remove VALARM by reference section.
6. Added more detail to CalDAV default alarms.

[Author's Address](#)

Cyrus Daboo Daboo Apple Inc. 1 Infinite Loop Cupertino , CA 95014
USA EMail: cyrus@daboo.name URI: <http://www.apple.com/>