VALARM Extensions for iCalendar
draft-daboo-valarm-extensions-04

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

Copyright Notice

Table of Contents

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.

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

)

[4](#). Alarm 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 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.

The "VALARM" component defined in Section 3 is extended here as:

```
alarmprop  /= *(

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

           uid

           )
```

5. Alarm Related To

It is often convenient to relate one or more "VALARM" components to
other "VALARM" components (e.g., see Section 9). This can be
accomplished if the "VALARM" components each have their own "UID"
property (as per Section 4).

This specification updates the usage of the "RELATED-TO" property
defined in Section 3.8.4.5 of [RFC5545] 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 Section 3.2.15 of
[RFC5545].

The "VALARM" component defined in Section 3 is extended here as:

```
alarmprop  /= *(

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

           related

           )
```

6. Alarm 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 then 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.

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.

   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

)

7. Alarm Agent

   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.

   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,
   an "AGENT-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.

```
   alarmprop  /= *(

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

                alarm-agent

                )
```

7.1. Alarm Agent Property

   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 "AGENT-ID" property parameter is specified for "SERVER" or
      "CLIENT" values, then only the client or server identified by the
      "AGENT-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 "AGENT-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:

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

   alarmagentparam = *(

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

                    (";" agentidparam) /

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

                    (";" other-param)

)

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

   Example: The following are examples of this property:

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

7.2. Agent ID Property Parameter

   Parameter Name: AGENT-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:

   agentidparam = "AGENT-ID" "=" DQUOTE uri DQUOTE

   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:

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

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
   ([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 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" is added or changed in the
      alarm component. If the value of any "ACKNOWLEDGED" 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 Section 8.1). 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 Section 3.2.15 of [RFC5545]. 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 [RFC5870] which allows for
two or three co-ordinate 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 that.

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.

      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:

   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

   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
      PROXIMITY:DEPART
      STRUCTURE-LOCATION;VALUE=URI:geo:40.443,-79.945;u=10
      DESCRIPTION:Remember to buy milk
      END:VALARM
```

11. 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.

11.1. Server Behavior

   A server supporting the features described in this document MUST
   include "calendar-default-alarms" as a field in the DAV response
   header from an OPTIONS request on a calendar home or calendar
   collection.

   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 as per Section 3).

   When adding a default alarm to an event or task, the server MUST
   ensure that a "DEFAULT-ALARM" property with a value of "TRUE" is
   present in the "VALARM" component.

   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.

   A default alarm MUST NOT be added to individual event or task
   components in a calendar object resource being created, if a "VALARM"
   component is already present in the component.

   When the client stores a new alarm in any of the WebDAV properties,
   servers MUST NOT change any existing calendar data within the
   calendar home or calendar where the WebDAV property is stored. This
   avoids the server having to rewrite potentially large amounts of
   calendar data to update the default alarm, and thus avoids the need
   for clients to re-download that data when they next synchronize with
   the server. The reasoning for this is described further in the next
   section.

11.2. Client Behavior

   Clients that support default alarms SHOULD allow users to set the
   four types of default provided by the WebDAV properties defined in
   Section 11.1. Clients SHOULD check for the presence of those

   properties and the calendar user's calendar home and individual
   calendars and use the values for defaults that they themselves apply.

   Since existing calendar data with a default alarm is not changed when
   the default alarm property changes, clients SHOULD do the following:

      Treat the default alarm values in the WebDAV properties as
      "overrides" for any alarms appearing in cached calendar data that
      have the "DEFAULT-ALARM" property set to a value of "TRUE". In
      effect the client ignores the default alarm in the calendar data
      and use the current default from the WebDAV properties.

      When a client updates calendar data on the server, it SHOULD
      replace any existing default alarms in the calendar data with the
      latest defaults retrieved from the WebDAV properties.

   The approach of not having the server rewrite existing calendar data
   when an alarm changes, and having the client "override" defaults in
   calendar data with the current default values is a compromise
   designed to provide some backwards compatibility with clients not
   supporting default alarm behavior.

11.3. Action None

   When default alarms are being used it is necessary to know when a
   calendar user has set a default alarm that does nothing as opposed to
   specifying that no alarm be used. This is important because clients
   are expected to "override" any default alarm present in calendar data
   with the current value retrieved from the server. However, if the
   calendar user explicitly removed the default alarm, then the
   "override" does not happen.

   This extension defines a new "NONE" 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. The "NONE" action is used
   solely to indicate a default alarm that does not alert the calendar
   user.

```
   actionvalue /= "NONE"
                  ; Adds a new action for a "VALARM"


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


   noneprop     = (


                  ; No properties used


                  )
```

11.4. Default Alarm Property

   Property Name: DEFAULT-ALARM

   Purpose: This property indicates that an alarm is a default alarm.

   Value Type: BOOLEAN

   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 a "VALARM"
      component is a default alarm.

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

```
   alarmprop        /= *(

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

                     default-alarm

                     )

   default-alarm     = "DEFAULT-ALARM" defaultalarmparam ":"
                     boolean CRLF

   defaultalarmparam = *(

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

                    (";" other-param)

                    )
```

12. 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.

13. IANA Considerations

13.1. 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 7.1  |
            | ACKNOWLEDGED  | Current | RFCXXXX, Section 8.1  |
            | PROXIMITY     | Current | RFCXXXX, Section 10.1 |
            | DEFAULT-ALARM | Current | RFCXXXX, Section 11.4 |
            +---------------+---------+-----------------------+
```

13.2. 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           |
      +--------------------+---------+---------------------+
      | AGENT-ID           | Current | RFCXXXX, Section 7.2 |
      +--------------------+---------+---------------------+
```

13.3. 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 6   |
        | NONE   | Current | RFCXXXX, Section 11.3 |
        +--------+---------+----------------------+
```

13.4. Relationship Types Registry

This document defines the following new iCalendar relationship type
to be added to the registry defined in Section 8.3.8 of [RFC5545]:

```
      +-------------------+---------+---------------------+
      | Relationship Type | Status  | Reference           |
      +-------------------+---------+---------------------+
      | SNOOZE            | Current | RFCXXXX, Section 9.1 |
      +-------------------+---------+---------------------+
```

13.5. Proximity Value Registry

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

```
        +--------+---------+----------------------+
        | Value  | Status  | Reference            |
        +--------+---------+----------------------+
        | ARRIVE | Current | RFCXXXX, Section 10.1 |
        | DEPART | Current | RFCXXXX, Section 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.

15. References

15.1. 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. 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.

15.2. 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 -04:

   1. Changed "ID" to "AGENT-ID".

Daboo   Expires December 11, 2012   [Page 21]

2. Add more text on using "ACKNOWLEDGED" property.

3. Add "RELATED-TO" as a valid property for VALARMs.

4. Add "SNOOZE" relationship type for use with VALARMs.

5. State that "TRIGGER" is typically ignored in proximity alarms.

6. Added "PROXIMITY" value registry.

7. Added a lot more detail on default alarms including new action
    and property.

Changes in -03: none - resubmission of -02

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
   Apple Inc.
   1 Infinite Loop
   Cupertino, CA 95014
   USA

   Email: cyrus@daboo.name
   URI: http://www.apple.com/