Network Working Group
draft-ietf-calsch-mod-00.txt
Category: INTERNET DRAFT
Expires: Dec 1997
J. Noerenberg
Qualcomm, Inc

Internet Calendar Model Specification

Status of this Memo

Abstract

Internet Calendaring and Scheduling protocols require the definition of objects to encapsulate the notion of an event to be scheduled, a calendar on which the event will be stored, and means for exchanging these objects across the Internet between calendars on behalf of people for whom the calendars are meaningful. This document gives an abstract model of the objects and the protocols necessary to accomplish this kind of information exchange. It will establish the context in which other Calendaring and Scheduling RFCs can be interpreted. Included are brief introductions to the other components of Internet calendar protocols and definitions of nomenclature common to all documents defining these protocols. Reading this document will enable implementors and users of Internet Calendaring and Scheduling protocols to understand the goals and constraints chosen for related protocols.

Table of Contents

Introduction

The Internet Calendar Model specification provides a framework for the set of protocols that define Calendaring and scheduling for the Internet. The protocols specify the contents of calendars, and how the objects stored on calendars are represented during transmission across the Internet. These protocols also define the interaction between a calendar user agent and a calendar store, as well as the actions performed by calendar transfer agents that facilitate communication between calendar stores. These terms will be defined more precisely in the section on nomenclature below. The protocols are the:

"Core Object Specification" [CoreObjSpec]
"Calendar Interoperability Protocol" [CalIntProt]
"Calendar Access Protocol" [CalAccProt]

1. Document framework
Calendar and Scheduling Protocols are contained in a series of related documents. This section describes the relationship between the documents.

1.1 Model Specification
This document - see abstract and introduction above.

1.2 iCalendar: Core Object Specification
The Core Object Specification is the authoritative definition of all properties that may be used in the Internet Calendar and Scheduling Protocols as well as the rules for encoding and representing the objects that are constructed from those properties. iCalendar also specifies the method to be used to define new attributes.

1.3 iTIP: Transport Independent Interoperability Protocol
This document specifies how calendaring systems use iCalendar objects to interoperate with other calendar systems. It does so in a general way so as to allow multiple methods of communication between systems.

Binding of iTIP to a real-time protocol
This document specifies a session-layer iTIP protocol. Multiple bindings are possible. This WG will specify and foster implementation of at least one binding.

Binding of iTIP to E-mail
This document specifies an iTIP protocol over Internet e-mail using MIME. Internet e-mail protocols are given by RFCs 821, 822, 2045-2049

[RFC-821] [RFC-822] [RFC- 2045] [RFC-2046] [RFC-2047]. [RFC-2048] [RFC-2049]. See the references for details for constructing Internet e-mail messages.

1.4 CAP: Calendar Access Protocol Specification
This document specifies how a Calendar User Agent (CUA) will interact with a Calendar Service using iCalendar objects.

A graphical representation of the relationship between the documents is shown below:

```
                        ------------------
                                          | Model Document   |
                        ------------------
                                                   |
                                                   |
                                                   |
                        ------------------
                                          |    iCalendar     |
                        ------------------
                                                   |
                                                   |
                                                   |
               -----------------------------------
                          |                                  |
      ------------------                 ------------------
         |      iTIP        |                |        CAP       |
      ------------------                 ------------------
               |
          -----------------
          |               |
 ----------       -----------
| session  |     |   email   |
|   iTIP   |     |    iTIP   |
 ----------       -----------
```

2. Calendar protocol nomenclature
Calendaring and Scheduling uses a rich lexicon of terms that are specific to the problems of scheduling events and reconciling conflicting requests for time and resources. This document will identify the major components of these systems, and show component relationships. However, for the sake of completeness and to serve as an introduction to the protocols in general, a more extensive list of terms, and brief definitions are included here. Essential parts of the system model have expanded definitions in this document where the components of the model are introduced.

2.1 Calendaring lexicon

Alarm, Reminder

An asynchronous mechanism providing feedback for a pending or past event or to-do.

Busy Time
A period of time that is not available for scheduling.

Calendar
A particular collection of calendar components.

Calendar Component
The objects that can be found in a calendar. The components are events, to-dos, journals, free/busies, time zones and alarms. Components also include properties and may include other components. A calendar component is identified by unique delimiters.

Calendar Date
A day identified by its position within the calendar year.

Calendar Scale
A particular type of calendar year, for example, Gregorian, Buddhist Era, Japanese Emperor Era, Chinese Lunar, Islamic, and Jewish Calendars.

Calendar Service
A Calendar Service (CS) stores a collection of calendars and interacts with the Calendar User Agent (CUA) via the Calendar Access Protocol (CAP).

Calendar Transport Agent (CTA)
A CTA mediates the exchange of calendar objects between calendars. It is generally responsible for interpreting the Transport Independent Interoperability Protocol (iTIP) used to exchange calendar objects either within a domain or across calendar domains. It can be implemented as a daemon that operates automatically without continuous human intervention.

Calendar User Agent (CUA)
A CUA mediates the interactions between a calendar user and his calendar. It represents the information stored in the calendar to the user, and enables the user to manipulate it. This is a particular instance of the interactive process used by a calendar user.

Coordinate Universal Time (UTC)
The time scale maintained by the Bureau International de l'Heure (International Time Bureau). UTC is often incorrectly referred to as GMT.

Daylight Saving Time (DST)
An adjustment to local time to accommodate annual changes in the number of daylight hours. DST is also known as Advanced Time, Summer Time, or Legal Time. Daylight Saving Time adjustments in the Southern Hemisphere are opposite to those in the Northern Hemisphere.

Event

A calendar component that defines a scheduled activity, minimally specified by a start and end calendar date and time of day and a description.

Free Time
A period of time available for scheduling on a calendar.

FreeBusy
FreeBusy components describe blocks of allocated and unallocated time on a calendar. They do not contain a description why the time is allocated.

Gregorian Calendar
A calendar scale in general use beginning in 1582. The Gregorian Calendar scale is based on a solar calendar consisting of common years made up of 365 days and leap years made up of 366 days; both divided into 12 sequential months.

Journal
A calendar component that defines a collection of information intended for human presentation and is minimally specified by a calendar date and one or more descriptions.

Local Time
The clock time in public use in a locale. Local time is often referenced by the customary name for the time zone in which it is located. The relationship between local time and UTC is based on the offset(s) that are in use for a particular time zone. In general, the formula is as follows:

local time = UTC + (offset)

Period
A duration of time, specified as either a defined length of time or by its beginning and end points.

Properties
Attributes that apply to calendar components or calendars. A calendar component is a named set of properties. Properties can also be used to produce variants to suit a particular purpose.

Recurrence Rule
A notation used to represent repeating occurrences, or the exceptions to such a repetition of an event or a to-do. The recurrence rule can also be used in the specification of a time zone description.

Repeating Event or To-do
An event or to-do that repeats for one or more additional occurrences. The recurrence may be defined with discrete dates and times and/or with a recurrence rule.

Standard Time

Introduced by Sir Sanford Fleming and others around 1870, standard time is a scheme for dividing the world into zones where the same time would be kept. The original proposal was to divide the world into 24 zones, each zone having a width of 15 degrees of longitude. The center zone was originally the meridian passing through Greenwich, England, called Greenwich Mean Time (GMT). The time in the zones was decremented by one hour per zone going westwards and was incremented by one hour per zone going eastwards from GMT. Changes have been made to the original proposal to accommodate political boundaries. In addition, some countries and regions specify 30 or 45 minute offsets, rather than the full 60 minute offset. Standard time is also known as Winter Time in some regions.

GMT and UTC are generally equivalent. However, by international agreement, the GMT term is discouraged in favor of the term UTC for all general time keeping.

Time Zone
A geographic region to which a specified offset from UTC applies. While offsets can frequently be calculated by knowing the longitudinal distance from Greenwich, England, local conventions frequently alter the calculation, complicating the determination of local time. Local convention may also assign a label to identify the time zone. There is no world-wide standard for labels.

To-do
A calendar component that defines an action item and is minimally specified by an effective calendar date and time of day, a due calendar date and time of day, a priority and a description.

3. Model Components

There are several principal components in a Calendaring and Scheduling system. Their relationship can be seen in Figure 2 below. This section identifies some of the salient features of the components. The syntax and semantics for creating and transmitting these objects are completely specified in [CoreObjSpec], [CalAccProt], and [CalIntProt].

3.1 Calendar User

A calendar user is the entity that interprets objects on a calendar, creates them, and exchanges them with other calendar users. A calendar user may be a person (Ken Caminiti), a group of people (the games of the San Diego Padres baseball club), or a resource (Padre's home games at the "Q"). From the point of view of other calendar users, groups and resources appear as a single entity to other calendar users, regardless of their composition in the physical world. Calendar users that are resources generally contain properties that identify them as inanimate objects -- anything from a fruit bowl to rubber bats to settle disputes during a meeting.

A calendar user owns his own calendar, and can manipulate objects stored there via a CUA. Access control attributes condition access to calendars and their components and properties.

A calendar user can also manipulate the contents of other calendar users' calendars by sending messages containing calendar objects to them. For example, The San Diego Padres sends calendar events for the

1997 season to Ken Caminiti, so he knows when to show up at the ballpark. The Padres sends calendar events for games to be played at home to the Qualcomm Stadium calendar so the concessionaires can order hot dogs.

3.2 Calendar

3.2.1 Collection of objects

Calendar users own calendars. This is a one to many relationship. A single calendar user may have multiple calendars. A calendar is an information store containing information about events,to-dos, alarms, journals and free time, the components stored in it. Also stored in a calendar are properties that are global to all of the objects in the calendar. An example of a global property is the GEO property that identifies the physical location where the calendar user can be found. Global properties such as this establish the context used to interpret the objects stored in the calendar. The principal structural features of a calendar are described below in section 3.2.3. **When components or properties of a calendar are exchanged** between actors in a calendaring and scheduling network (Calendar User Agents and Calendar Services), they are expressed in the form defined in [CoreObjSpec]. Figure 1 below is a schematic representation of a calendar.

3.2.2 Properties

Properties are attributes of a component or a calendar. They consist of a name and a value. Properties are strongly typed. Some properties are multivalued. A property may have parameters that distinguish between related properties. Some properties may occur multiple times in the same component instance, and may be gathered into a logical group. Some properties may be unique to a particular calendar or component.

3.2.3 Components

Components are collections of property values. A particular set of values for the properties of a component define a particular component. Some components may contain certain other components. The set of components in a calendar are identified below.

3.2.3.1 Events

Event components are defined for specific starting date-time, have a duration on a calendar, and a description. Other properties of an event may specify a location or other attributes that define the event, such as resources that are part of the event. Events may also contain an Alarm component.

3.2.3.2 To-do

While like an event, a To-do doesn't reserve a specific block of time on a calendar. A To- do component must have a starting date-time that identifies its first appearance on the calendar. It must also have a date-time that specifies when the To-do expires. A To-do must have a description. It may also contain an alarm component.

3.2.3.3 Alarms

An alarm component may occur in an Event or To-do. It contains a date-time. When present, and the date-time is passed, it will cause a CUA or CS to notify the user the date-time has passed.

3.2.3.4 Journal

A journal component is a textual item that is associated with a particular date. As its name suggests, its purpose is to record information meaningful about the date, but not necessarily tied to other calendar components on a calendar.

3.2.3.5 Timezone

Timezone components encapsulate rules for calculating local time from UTC. Including this component in an event component enables a receiver to calculate the universal time value for time values expressed in the sender's local time. This component is especially useful for expressing recurring events whose instances span a change in the time reference such as the transition between Standard time and Daylight Savings time. Time values expressed in local time are always interpreted in the receiver's local time. The sender must provide another context using UTC values and Timezone components if this is not the interpretation intended by the sender.

```
                               calendar
                                  |
   -----------------------------------------------------------
  |         |            |           |            |           |
to-do     event       journal     freebusy     timezone    property...
            |
     --------------
    |              |
property...      alarm
                   |
                             property...
```

Figure 1: Calendar Object Model

3.3 Calendar User Agent (CUA)

A CUA mediates the interactions between a calendar user and his calendar. It represents the information stored in the calendar to the user, and enables the user to manipulate it. This is a particular instance of the interactive process used by a calendar user.

3.4 Calendar Service

A Calendar Service (CS) stores a collection of calendars and interacts with the Calendar User Agent (CUA) via the Calendar Access Protocol (CAP).

3.5 Calendar Domain

A collection of calendars that can be grouped together constitutes a Calendar Domain. The relation used to bound the group is arbitrary. Frequently membership in an organization will be used to define the domain, but it could be a shared Internet address domain, as well. A Calendar Domain provides a contiguous address space for all the calendars, CTAs and CUAs contained in the domain. It must be possible for any Calendar User (via the facilities of a CUA and/or CTA) to determine whether they are members of a particular domain, or if other Calendar Users are members. CTAs and CUAs can take advantage of domain information when routing event messages.

3.6 Calendar Access Protocol (CAP)

When calendar users need to manipulate calendars that are not stored on the same computer where the CUA executes, the CUA will use the Calendar Access Protocol to exchange components with the Calendar Service (CS). CAP specifies the beginning and ending of the session between the CUA and the CS. Using CAP, the CUA will mediate authentication of the user to the service. CAP requires calendar components and calendar properties to be expressed in the on-the-wire data format defined in [CalObjSpec]. A CUA must not be required to maintain a connection to a CS via CAP in order to display a Calendar for a Calendar User or accept commands from a user to change a Calendar's content. By using CAP a CUA need not have specific information on how a particular CS stores a Calendar and vice versa. This enables specification and exchange of components and properties independently of Calendar storage models adopted by particular CUAs or CSs.

3.7 Transport Independent Interoperability Protocol (iTIP)

CSs in a domain or across domains exchange components and properties using iTIP. Like CAP, iTIP uses iCalendar formats to represent components and properties. iTIP defines the beginning and ending of the exchange session, as well the users for whom the messages are intended. iTIP permits unauthenticated delivery of components and properties to a CS. A CS may accept or reject delivery without interaction with a user. But a CS may require further confirmation of receipt of a component or property before it is acted upon by the CS.

4. Calendar System Model

This section presents the Calender and Scheduling system model. It describes how the elements identified and described in the section 3 relate to each other. This is done in two parts. Table 1 in section **4.1 below summarizes the relationships between pairs of elements.** Section 4.2, Calendar Transport Model, identifies and describes the different transport modes that are supported by Calendaring and Scheduling protocols.

4.1 Model Component Relationship

The table below identifies the defined relationships between pairs of elements. If a cell contains a number that means that the elements specified in the row and column headings have a relationship with each other. If a cell is blank then there is no relationship between the corresponding pair of model elements. Following the table the numbered relationships are described in detail.

Table 1: Component Relationships

	Calendar User	Calendar	Calendar User Agent	Calendar Service	Calendar Domain	iTIP	CAP
Calendar User		1	2				
Calendar			3	4	5		
Calendar User Agent			6	7	8	6,8	7
Calendar Service				9	9	9	
Calendar Domain					10	10	
iTIP							
CAP							

1. **The relationship between a Calendar and a Calendar User is defined** in the Model Components section 3.

2. **A Calendar User interacts with the system through a Calendar User** Agent. The Calendar User Agent is typically (but not necessarily) an interactive program that the Calendar User depends on to maintain their own calendar and to interact with other users' calendars, for example to invite them to meetings.

3. **A Calendar User Agent interacts with a Calendar typically on behalf** of a Calendar User. However, a Calendar User Agent may be a program without a user interface, for example a program that scans multiple calendars for vacation entries and maintains a summary in a separate calendar.

4. **A Calendar Service may store a Calendar although this model does** not require that. In fact, this model does not define how a Calendar is stored, leaving it open to the implementation to determine that. The calendar may be stored in a file system, in memory, in the message store of a messaging system etc.

5. The relationship between a Calendar and a Calendar Domain is

defined in the Model Components, section 3.

6. A Calendar User Agent may interact directly with another Calendar User Agent by using iTIP.

7. A Calendar User Agent may interact with a Calendar Service using the Calendar Access Protocol.

8. A Calendar User Agent may interact with a Calendar Domain via iTIP.

9. A Calendar Service may interact with another Calendar Service or a Calendar Domain using iTIP.

10. Calendar Domains may interact with each other using iTIP.

4.2 Calendar Transport Model

There are several transport modes in this architecture. The figures below illustrate the different modes that are allowed. Three modes are required to handled the different kinds of calendar exchanges across the Internet, person to person, group interactions local to a particular network, and exchanges across networks.

```
 ------------                  ------------
| CUA  | rcvr| -----      ----|rcvr|  CUA  |
|       -----|     \   /     |----        |
|            |      \ /      |            |
|            |       X       |            |
|            |      / \      |            |
|       -----|     /   \     |----        |
|      | sndr| -----      ----|sndr|       |
 ------------       \    /     ------------
                     iTIP
```

Figure 2: Direct Access

```
 ------------                  ------------
| CUA        |                |      CUA  |
|            |                |           |
|            |                |           |
 ------------                  ------------
        \                          /
         \   ------- CAP -------  /
          \                      /
           \   --------------   /
            \ |      CS      | /
              |              |
              |              |
               --------------
```

Figure 3: Calendar Service Mediation

```
 -----------------                     ------------------
|                 |                   |                  |
| Calendar Domain |                   | Calendar Domain  |
|                 |                   |                  |
|                 |                   |                  |
|  -------        |                   |                  |
| |  CUA  |       |                   |                  |
|  -------        |                   |                  |
|     |           |                   |                  |
|     | -- CAP    |                   |                  |
|     |           |                   |                  |
|  ------------   |                   |---------         |
| | CS   | rcvr|----------     -------|rcvr|    |        |
| |       -----|  |       \   /       |----     |        |
| |            |  |        \ /        | Gateway |        |
| |            |  |         X         |         |        |
| |            |  |        / \        |         |        |
| |       -----|  |       /   \       |----     |        |
| |      | sndr| ---------      ------|sndr|    |        |
|  ------------   |       \    /      |---------         |
|                 |        iTIP       |                  |
|                 |                   |                  |
 -----------------                     ------------------
```

Figure 4: Interdomain Exchange

4.2.1 Direct Access

For direct access, two calendar user agents (CUA) exchange calendar components by using iTIP. Each CUA provides an iTIP sender and receiver. As is generally the case, the methods used by the CUA to store calendar data locally are not relevant to the transport model. For this mode, calendar users must be uniquely identifiable across the Internet, and access to CUAs must conform with privacy and security considerations.

4.2.2 Calendar Service Mediation

Using Calendar Service Mediation a CUA interoperates with a Calendar Service (CS) using CAP to exchange calendar components. The CS takes responsibility for mediating receipt and delivery of components with other collaborating CUAs. The principal difference between this mode and Interdomain Exchange is that CSs do not need to use iTIP to facilitate exchange of components. A consequence of this mode is that CUAs and CSs need not use addresses that are unique across the Internet. However, consistency with other modes makes a consistent address model an obvious simplification for implementors.

4.2.3 Interdomain Exchange

With Interdomain Exchange a Calendar Service (CS) supporting a group of users in one domain can exchange calendar components with a different calendar domain. Domains may or may not be within the same Internet network domain. Like Direct Access, iTIP is the vehicle which permits component exchange. In the figure, one domain is illustrated with a Calendar Service providing iTIP service. The 2nd domain in this figure has a distinct iTIP sender and receiver. In order to provide end-to-end privacy components must be wrapped in a cryptographically secure wrapper to insure only the intended corespondents can interpret the components. This wrapper is not required unless privacy must be assured. In order to provide backward compatibility with existing calendar and scheduling systems, a privacy wrapper cannot be a required aspect of the component exchange.

5. Security considerations

The Core Object Specification must provides the means to classify the intended sensitivity level of an event, to-do or journal calendar component (i.e., PUBLIC, PRIVATE, or CONFIDENTIAL). It must also provide a means to wrap all components in an exchange in a cryptographically secure envelope so that only the intended correspondents can decode the message.

The Transport Independent Interoperability Protocol must provides a description of the authentication step that must be defined in any of the iTIP bindings.

The Calendar Access Protocol must provide a description of the elements of the calendaring system security model and the protocol elements for creating and manipulating the access control to a calendar.

6. References

[RFC-821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, USC/Information Sciences Institute, August 1982.

[RFC-822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, UDEL, August 1982.

[RFC-2045] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, Nov 1996

[RFC-2046] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions MIME) Part Two: Media Types", RFC 2046, Nov 1996

[RFC-2047] Borenstein, N. & Freed, N., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, Nov 1996

[RFC-2048] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", RFC 2048, Nov 1996

[RFC-2049] Borenstein, N. & Freed, N., "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, Nov 1996

[CoreObjSpec] "Core Object Specification"

[iTIP] "Calendar Interoperability Protocol"

[CalAccProt] "Calendar Access Protocol"

7. Acknowledgments

The author is extremely grateful for the assistance, patience and tolerance of the members of the CalSch working group. Their ideas and suggestions are crucial to making this a useful document. In particular, the author would like to thank
Anik Ganguly
Derik Stenerson
Frank Dawson
Their comments and ideas were particularly important for the original formulation of this draft. I would also like to thank Qualcomm, Incorporated for allowing the time necessary to bring this effort to fruition.

8. Author's address

For more information, please contact the author via Internet Mail.

John W. Noerenberg, II
Qualcomm, Incorporated
6455 Lusk Blvd
San Diego, CA 92131
USA

email: jwn2@qualcomm.com
ph: +1 619 658 3510

The "Internet Calendar Model Specification" is the work of the Internet Engineering Task Force Working Group on Calendaring and Scheduling. The chairman of that group, Anik Ganguly, may be reached at:

Anik Ganguly
Campbell Services, Inc.
21700 Northwestern Highway
10th Floor
Southfield, MI 48075
email: anik@ontime.com