

Network Working Group
Internet-Draft
Expires: January 16, 2002

B. Mahoney
MIT
G. Babics
Steltor
A. Taler
July 18, 2001

Guide to Internet Calendaring
draft-ietf-calsch-inetcal-guide-01

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on January 16, 2002.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

This document describes the various Internet calendaring and scheduling standards and works in progress and the relationships between them. It's intention is to provide a context for these documents, assist in their understanding, and potentially help implementers in the design of their standards based calendaring and scheduling systems. The standards addressed are [RFC 2445](#) (iCalendar), [RFC 2446](#) (iTIP), and [RFC 2447](#) (iMIP). The work in progress addressed is "Calendar Access Protocol" (CAP). This document also describes issues and problems that are not solved by

these protocols, and could be targets for future work.

Table of Contents

1.	Introduction	3
1.1	Terminology	3
1.2	Concepts and Relationships	5
2.	Requirements	6
2.1	Fundamental Needs	6
2.2	Protocol Requirements	6
3.	Solutions	8
3.1	Examples	8
3.2	Systems	9
3.2.1	Standalone single-user system	9
3.2.2	Single-user systems communicating	9
3.2.3	Single-user with multiple CUA	10
3.2.4	Single-user with multiple calendars	10
3.2.5	Users communicating on a multi-user system	11
3.2.6	Users communicating through different multi-user systems	11
4.	Important Aspects	12
4.1	Timezones	12
4.2	Choice of Transport	12
4.3	Security	12
4.4	Amount of data	12
4.5	Recurring Components	12
5.	Open Issues	14
5.1	Scheduling people, not calendars	14
5.2	Administration	14
5.3	Notification	14
6.	Security considerations	15
6.1	Access Control	15
6.2	Authentication	15
6.3	Using email	15
6.4	Other issues	16
	Authors' Addresses	16
A.	Acknowledgments	17
B.	Bibliography	18
	Full Copyright Statement	19

1. Introduction

The calendaring and scheduling protocols are intended to provide for the needs of individuals attempting to obtain calendaring information and schedule meetings across the Internet, organizations attempting to provide calendaring information on the Internet, as well as organizations looking for a calendaring and scheduling solution to deploy internally.

It is the intent of this document to provide a context for the calendar standards and works in progress, assist in their understanding, and potentially help implementers in the design of their Internet calendaring and scheduling systems.

Problems not solved by these protocols, as well as security issues to be kept in mind, are discussed at the end of the document.

1.1 Terminology

This memo uses much of the same terminology as iCalendar [[RFC-2445](#)], iTIP [[RFC-2446](#)], iMIP [[RFC-2447](#)], and [[CAP](#)]. The following definitions are provided as introductory, the definitions in the protocol specifications are the canonical ones.

Calendar

A collection of events, to-dos, journal entries, etc. A calendar could be the content of a person or a resource's agenda; it could also be a collection of data serving a more specialized need. Calendars are the basic storage containers for calendaring information.

Calendar Access Rights

A set of rules for a calendar describing who may perform which operations on that calendar, such as reading and writing information.

Calendar Service

A running server application which provides access to a collection of calendars.

Calendar Store (CS)

A data store of a calendar service. A calendar service may have several calendar stores, and each store may contain several calendars, as well as properties and components outside of the

calendars.

Calendar User (CU)

An entity (often a human) that accesses calendar information.

Calendar User Agent (CUA)

Software used by the calendar user that communicates with calendar services to provide the user access to calendar information.

Component

A piece of calendar data such as an event, a to-do or an alarm. Information about components is stored as properties of those components.

Delegator

Is a calendar user (sometimes called the delegatee) who has assigned his or her participation in a scheduled calendar component (e.g., VEVENT) to another calendar user (sometimes called the delegate or delegatee).

Delegate

Is a calendar user (sometimes called the delegatee) who has been assigned participation in a scheduled calendar component (e.g., VEVENT) by one of the attendees in the scheduled calendar component (sometimes called the delegator). An example of a delegate is a team member told to go to a particular meeting.

Designate

Is a calendar user who is authorized to act on behalf of another calendar user. An example of a designate is an assistant.

Local Store

A CS which is on the same platform as the CUA.

Property

A property of a component, such as a description or a start time.

Remote Store

A CS which is not on the same platform as the CUA.

1.2 Concepts and Relationships

iCalendar is the language used to describe calendar objects. iTIP is a way to use the language to do scheduling. iMIP is how to do iTIP with email. CAP is a way to use the language, to access a calendar store in real-time.

The relationship between the calendaring protocols is similar to that between the email protocols. In those terms iCalendar is like [RFC 822](#), iTIP and iMIP are like SMTP, and CAP is like POP or IMAP.

2. Requirements

2.1 Fundamental Needs

The following examples illustrate people's and organizations' basic calendaring and scheduling needs:

a] A doctor wishes to keep track of all his appointments.

Need: Read and manipulate one's own calendar with only one CUA.

b] A busy musician wants to maintain her schedule with different devices, such as with an Internet-based agenda or with a PDA.

Need: Read and manipulate one's own calendar, possibly with solutions from different vendors.

c] A software development team wishes to share agenda information by using a group scheduling product in order to more effectively schedule their time.

Need: Share calendar information with users using the same calendar service.

d] A teacher wants his students to be able to schedule calendar entries during his office hours.

Need: Schedule calendar events, to-dos and journals with users using the same calendar service.

e] A movie theater wants to publish its schedule so that prospective customers can easily access it.

Need: Share calendar information with users using other calendar services, possibly from different vendors.

f] A social club wants to be able to schedule calendar entries effectively with its members.

Need: Schedule calendar events and to-dos with users using other calendar services, possibly from different vendors.

2.2 Protocol Requirements

Some of the needs can be met with proprietary solutions (a, c, d), but others can not (b, e, f). From these needs we can establish that protocols are required for accessing information in a calendar store,

and for scheduling calendar entries. In addition these protocols require a data format for representing calendar information.

These roles are filled by the following protocol specifications.

- iCalendar [[RFC-2445](#)] is the data format

iCalendar [[RFC-2445](#)] provides data format for representing calendar information which the other protocols can use. iCalendar [[RFC-2445](#)] can also be used in other contexts such as a drag and drop format or an export/import format. All the other protocols depend on iCalendar [[RFC-2445](#)], so all elements of a standards-based calendaring and scheduling systems will have to interpret iCalendar [[RFC-2445](#)].

- iTIP [[RFC-2446](#)] is the scheduling protocol

iTIP [[RFC-2446](#)] describes the messages used to schedule calendar events. These messages are represented in iCalendar [[RFC-2445](#)], and have semantics that include such things as being an invitation to a meeting, an acceptance of an invitation or the assignment of a task.

iTIP [[RFC-2446](#)] messages are used in the scheduling workflow, where users exchange messages in order to organize things such as events and to-dos. CUAs generate and interpret iTIP [[RFC-2446](#)] messages at the direction of the calendar user. With iTIP [[RFC-2446](#)] one can create, modify, delete, reply to, counter, and decline counters to the various iCalendar [[RFC-2445](#)] components. Furthermore, one can also request the free/busy time of other people.

iTIP [[RFC-2446](#)] is transport-independent, and has one specified transport bindings: iMIP [[RFC-2447](#)] is a binding to email. In addition [[CAP](#)] will provide a real-time binding of iTIP [[RFC-2446](#)], allowing CUAs to perform calendar management as well as scheduling over a single connection.

- [[CAP](#)] is the calendar management protocol

[CAP] describes the messages used to manage calendars on a calendar store. These messages use iCalendar [[RFC-2445](#)] to describe various components such as events and to-dos. With these messages one can do the operations in iTIP [[RFC-2446](#)] and other operations relating to a calendar store, such as search, creating calendars, specifying calendar properties, and being able to specify access rights to one's calendars.

3. Solutions

3.1 Examples

Returning to the examples of [section 2.1](#), they can be solved using the protocols in the following ways:

a] The doctor can use a proprietary CUA with a local store, and perhaps use iCalendar [[RFC-2445](#)] as a storage mechanism. This would allow the doctor to easily import his store into another application that supports iCalendar [[RFC-2445](#)].

b] The musician who wishes to access her agenda from anywhere can use a [[CAP](#)] enabled calendar service accessible through the Internet. She can then use whichever [[CAP](#)] clients are available to access the data.

A proprietary system could also be employed which provides access through a web-based interface, but the use of [[CAP](#)] would be superior in that it would allow the use of third party tools, such as PDA synchronization tools.

c] The development team can use a calendar service which supports [[CAP](#)] and then each member can use a [[CAP](#)]-enabled CUA of their choice.

Alternatively, each member could use an iMIP [[RFC-2447](#)]-enabled CUA, and they could book meetings over email. This solution has the drawback that it is difficult to examine the other agendas, making organizing meetings more difficult.

Proprietary solutions are also available, but they require that all people use clients by the same vendor, and disallow the use of third party applications.

d] The teacher can set up a calendar service, and have students book time through any of the iTIP [[RFC-2446](#)] bindings. [[CAP](#)] provides real-time access, but could require additional configuration. iMIP [[RFC-2447](#)] would be the easiest to configure, but may require more email processing.

If [[CAP](#)] access is provided then determining the state of the teacher's schedule is straightforward. If not, this can be determined through iTIP [[RFC-2446](#)] free/busy requests. Non-standard methods could also be employed, such as serving up ICAL, HTML, XML over HTTP.

A proprietary system could also be used, but would require that

all students be able to use software from a specific vendor.

e] For publishing a movie theater's schedule [CAP] provides the most advanced access and search capabilities. It also allows easy integration with its customer's calendar systems.

Non-standard methods such as serving data over HTTP could also be employed, but would be harder to integrate with customer's systems.

Using a completely proprietary solutions would be very difficult since it would require every user to install and use proprietary software.

f] The social club could distribute meeting information in the form of iTIP [RFC-2446] messages. This could be done over email using iMIP [RFC-2447]. Meeting invitations, as well as a full published agenda could be distributed.

Alternatively, the social club could provide access to a [CAP] enabled calendar service, however this solution would be more expensive since it requires the maintenance of a server.

3.2 Systems

The following diagrams illustrate possible example systems and usage of the protocols.

3.2.1 Standalone single-user system

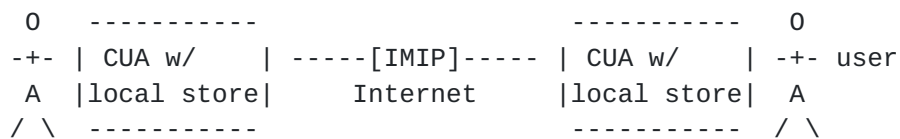
A single user system that does not communicate with other systems need not employ any of the protocols. However, it may use iCalendar [RFC-2445] as a data format in some places.



3.2.2 Single-user systems communicating

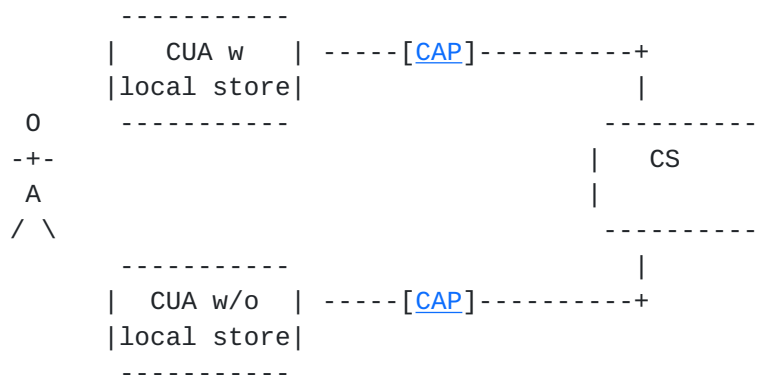
Users with single-user systems may schedule meetings with each other using iTIP [RFC-2446]. The easiest binding of iTIP [RFC-2446] to use is iMIP [RFC-2447], since since the messages can be held in their mail queue, which we assume to already exist. [CAP] could also be

used.



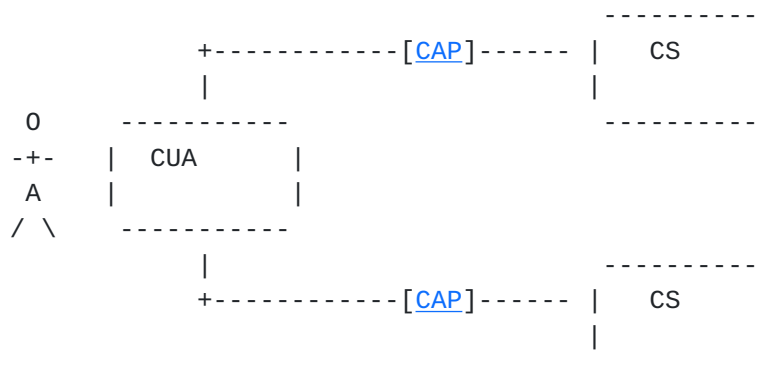
3.2.3 Single-user with multiple CUA

A single user may use more than one CUA to access his or her calendar. The user may use a PDA, a web client, a PC, or some other device, depending on accessibility. Some of these clients may have local stores and others may not. If they do, then they need to ensure that the data on the CUA is synchronized with the data on the CS.



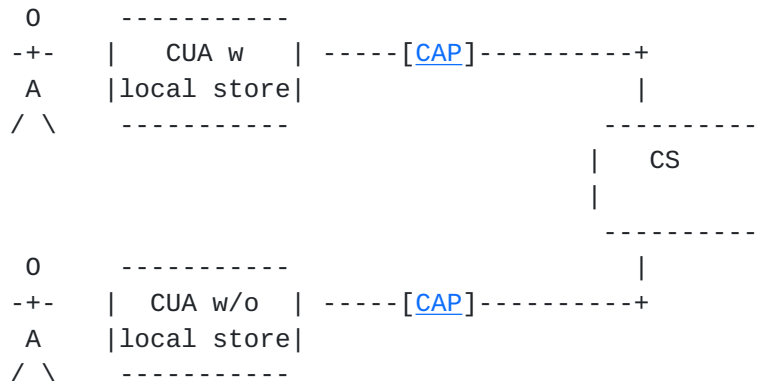
3.2.4 Single-user with multiple calendars

A single user may have many independent calendars. One may be work related, another for personal use. The CUA may or may not have a local store. If it does, then it needs to ensure that the data on the CUA is synchronized with the data on both of the CS.



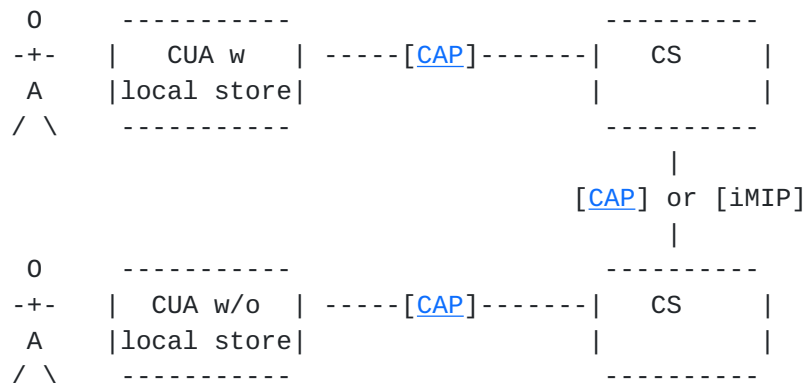
3.2.5 Users communicating on a multi-user system

Users on a multi-user system may schedule meetings with each other using [\[CAP\]](#)-enabled CUA and service. The CUA may or may not have a local store. If they do, then they need to ensure that the data on the CUA is synchronized with the data on the CS.



3.2.6 Users communicating through different multi-user systems

Users on a multi-user system may need to schedule meetings with user on a different multi user system. The services can communicate using [\[CAP\]](#) or iMIP [\[RFC-2447\]](#).



4. Important Aspects

There are a number of important aspects of these calendaring documents of which people, especially implementers, should be aware.

4.1 Timezones

The dates and times in components can refer to time zones. These time zones can be defined in some central store, or they may be defined by a user to fit his or her needs. Any user and application should be aware of time zones and time zone differences. New time zones may be added, and others removed. Two different vendors may describe the same time zone differently (such as by using a different name).

4.2 Choice of Transport

There are issues to be aware of in choosing a transport mechanism. The choices are a network protocol, such as CAP, or a store and forward (email) solution.

The use of a network ("on-the-wire") mechanism may require some organizations to make provisions to allow calendaring traffic to traverse a corporate firewall on the required ports. Depending on the organizational culture, this may be a challenging social exercise.

The use of an email-based mechanism exposes innately time sensitive data to unbounded latency. Large or heavily utilized mail systems may experience an unacceptable delay in message receipt.

4.3 Security

See the "Security Considerations" ([Section 6](#)) section below.

4.4 Amount of data

In some cases a component may be very large. For instance, some attachments may be very large. Some applications may be low-bandwidth or be limited in the amount of data they can store. The size of the data may be controlled in [[CAP](#)], by specifying maximums. In iMIP [[RFC-2447](#)] it can be controlled, by restricting the maximum size of the email that the application can download.

4.5 Recurring Components

In iCAL [[RFC-2445](#)] one can specify complex recurrence rules for VEVENTs, VTODOs, and VJOURNALS. There is the danger that

applications interpret these rules differently. Thus, one must make sure that one is careful with recurrence rules.

5. Open Issues

Many issues are not currently resolved by these protocols, and many desirable features are not yet provided. Some of the more prominent ones follow.

5.1 Scheduling people, not calendars

Meetings are scheduled with people, however people may have many calendars, and may store these calendars in many places. There may also be many routes to contact them. These protocols do not attempt to provide unique access for contacting a single person. Instead, 'calendar addresses' are booked, which may be email addresses or individual calendars. It is up to the users themselves to orchestrate mechanisms to ensure that the bookings go to the right place.

5.2 Administration

These protocols do not address the issues of administering users and calendars on a calendar service. This must be handled by proprietary mechanisms for each implementation.

5.3 Notification

People often wish to be notified of upcoming events, new events, or changes to events. These protocols do not attempt to address these needs in a real-time fashion. Instead, the ability to store alarm information on events is provided, which can be used to provide client-side notification of upcoming events. To organize notification of new or changed events clients will have to poll the data store.

6. Security considerations

6.1 Access Control

There has to be reasonable granularity in the configuration options for access to data through [\[CAP\]](#), so that what should be released to requesters is released, and what shouldn't isn't. Details of handling this are described in [\[CAP\]](#).

6.2 Authentication

Access control must be coupled with a good authentication system, so that the right people get the right information. For [\[CAP\]](#) this means requiring authentication before any database access can be performed, and checking access rights and authentication credentials before releasing information. [\[CAP\]](#) uses SASL for this authentication. In iMIP [\[RFC-2447\]](#), this may present some challenges, as authentication is often not a consideration in store-and-forward protocols.

Authentication is also important for scheduling, in that receivers of scheduling messages should be able to validate the apparent sender. Since scheduling messages are wrapped in MIME [\[RFC-2045\]](#), signing and encryption is available for free. For messages transmitted over mail this is the only available alternative. It is suggested that developers take care in implementing the security features in iMIP [\[RFC-2447\]](#), bearing in mind that the concept and need may be foreign or non-obvious to users, yet essential for the system to function as they might expect.

The real-time protocols provide for the authentication of users, and the preservation of that authentication information, allowing for validation by the receiving end-user or server.

6.3 Using email

Because scheduling information can be transmitted over mail without any authentication information, email spoofing is extremely easy if the receiver is not checking for authentication. It is suggested that implementers consider requiring authentication as a default, using mechanisms such as are described in [Section 3](#) of iMIP [\[RFC-2447\]](#). The use of email, and the potential for anonymous connections, means that 'calendar spam' is possible. Developers should consider this threat when designing systems, particularly those that allow for automated request processing.

6.4 Other issues

The current security context should be obvious to users. Because the underlying mechanisms may not be clear to users, efforts to make clear the current state in the UI should be made. One example is the 'lock' icon used in some web browsers during secure connections. With both iMIP [[RFC-2447](#)] and [[CAP](#)], the possibilities of Denial of Service attacks must be considered. The ability to flood a calendar system with bogus requests is likely to be exploited once these systems become widely deployed, and detection and recovery methods will need to be considered.

Authors' Addresses

Bob Mahoney
MIT
E40-327
77 Massachusetts Avenue
Cambridge, MA 02139
US

Phone: (617) 253-0774
EMail: bobmah@mit.edu

George Babics
Steltor
2000 Peel Street
Montreal, Quebec H3A 2W5
CA

Phone: (514) 733-8500 x4201
EMail: georgeb@steltor.com

Alex Taler

EMail: dissent@elea.dhs.org

[Appendix A](#). Acknowledgments

Thanks to the following who have participated in the development of this document:

Eric Busboom, Pat Egen, David Madeo, Shawn Packwood, Bruce Kahn.

Appendix B. Bibliography

[RFC-2445] Dawson, F. and D. Stenerson, "Internet Calendaring and Scheduling Core Object Specification - iCalendar", [RFC 2445](#), November 1998.

[RFC-2446] Silverberg, S., Mansour, S., Dawson, F. and R. Hopson, "iCalendar Transport-Independent Interoperability Protocol (iTIP): Scheduling Events, Busy Time, To-dos and Journal Entries", [RFC 2446](#), November 1998.

[RFC-2447] Dawson, F., Mansour, S. and S. Silverberg, "iCalendar Message-Based Interoperability Protocol - iMIP", [RFC 2447](#), November 1998.

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

[CAP] Mansour, S., Royer, D., Babics, G., and Hill, P. "Calendar Access Protocol (CAP)" [draft-ietf-calsch-cap-04.txt](#)

Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

