

Network Working Group
Internet Draft
<[draft-ietf-calsch-capreq-03.txt](#)>

Andre Courtemanche, CS&T
Frank Dawson, Lotus
Steve Mansour, Netscape
Pete O'Leary, Amplitude
Doug Royer, Sun Microsystems
Tony Small, Microsoft
May 27, 1999

Expires six months after:

CAP Requirements

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>.

Abstract

The Calendar Access Protocol (CAP) is an Internet protocol for accessing an [[ICAL](#)] based calendar store (CS) from a calendar user agent (CUA). The purpose of this document is to set forth a list of requirements that CAP MUST address in order to meet the needs of the calendaring and scheduling community.

This document is based on discussions within the Internet Engineering Task Force (IETF) Calendaring and Scheduling (CALSCH) working group. More information about the IETF CALSCH working group activities can be found on the IMC web site at <http://www.imc.org>, the IETF web site at <http://www.ietf.org/html.charters/calsch-charter.html>. Refer to the references within this document for further information on how to access these various documents.

Distribution of this document is unlimited. Comments and suggestions for improvement should be sent to the authors.

Copyright (C) The Internet Society 1998. All Rights Reserved.

Change History

Version 00 - Initial draft.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small
1 Expires November 1999

Internet Draft CAP Requirements May 27, 1999

Version 01 - Revised format; Included initial set of scenarios and requirements.

Version 02 - Revised format; Significant modification to set of requirements. Included lists of requirements that are deferred to later versions of CAP or to other drafts.

Version 03 - Repost of version 02.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small
2 Expires November 1999

Internet Draft CAP Requirements May 27, 1999

Table of Contents

[1. Introduction.....4](#)
[2. Terminology.....4](#)
[3. Relationship To iCalendar, iTIP and iMIP/iRIP.....7](#)
[4. Requirements.....7](#)
 [4.1 Basic Usage7](#)
 [4.2 Infrastructure8](#)
 [4.2.1 Connection Models8](#)
 [4.2.2 Store Location Models9](#)
 [4.2.3 Calendar Stores10](#)
 [4.2.4 Exception Reporting11](#)
 [4.3 Operations on a Calendar11](#)
 [4.3.1 Deferred Requirements for Operations on a Calendar12](#)
 [4.4 Component Management12](#)
 [4.4.1 Recurrence13](#)
 [4.4.2 Calendar and Component Policies14](#)
 [4.4.3 Deferred Requirements for Component Management14](#)
 [4.5 Lookup, Query and Discovery15](#)
 [4.5.1 Lookup15](#)
 [4.5.2 Query Capabilities15](#)
 [4.5.3 Specific Queries17](#)
 [4.5.4 Discovery17](#)
 [4.5.5 Deferred Requirements for Search18](#)
 [4.6 Security18](#)
 [4.7 Designates and Delegation18](#)
 [4.8 Notification \(deferred\)19](#)
 [4.9 Access Control \(deferred\)19](#)
 [4.10 Resource Groups and Pools \(deferred\)20](#)

4.11	CAP Non-requirements	21
5	Open Issues.....	21
6	Acknowledgments.....	22
7	Bibliography.....	22
8	Authors' Address.....	22
9	Full Copyright Statement.....	24

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

3

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

[1.](#) **Introduction**

The Calendar Access Protocol (CAP) is an Internet protocol for accessing an [[ICAL](#)] based calendar store (CS) from a calendar user agent (CUA). The purpose of this document is to set forth a list of requirements that CAP MUST address in order to meet the needs of the calendaring and scheduling community.

[2.](#) **Terminology**

The terminology used in [[ICAL](#)], [[ITIP](#)] and [[IMIP](#)] are also used within this memo.

Calendar

A collection of logically related objects or entities each of which may be associated with a calendar date and possibly time of day. These entities can include other calendar properties or calendar components. In addition, a calendar might be hierarchically structured and consist of other sub-calendars. The [[ICAL](#)] defines calendar properties, calendar components and component properties. A calendar is identified by its unique calendar identifier.

Calendar Access Protocol

An Internet protocol that allows a CUA to access and operate on a CS.

Calendar Access Rights

Some method for specifying permitted operational capabilities for a calendar user.

Calendar Component

An entity within a calendar. Types of calendar components include events, to-dos, journals, alarms, time zones and freebusy data. A calendar component consists of component properties and possibly other sub-components. For example, an event can consist of an alarm component.

Calendar Component Properties

An attribute of a particular calendar component. Some calendar

component properties are applicable to different types of calendar components. For example, DTSTART is applicable to VEVENT, VTODO, VJOURNAL calendar components. Other calendar components are applicable only to an individual type of calendar component. For example, TZURL is only applicable to VTIMEZONE calendar components.

Calendar Identifier

Also known as "CalID". A globally unique identifier associated with a calendar within a CS.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

4

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

Calendar Policy

An operational restriction on the access or manipulation of a calendar. For example, "events MUST be scheduled in unit intervals of one hour".

Calendar Properties

An attribute of a calendar. The attribute applies to the calendar, as a whole. For example, CALSCALE specifies the calendar scale (e.g., GREGORIAN) for the whole calendar.

Calendar Service

An implementation of an application that manages one or more calendars.

Calendar Store

Also known as "CS". The data model definition for a Calendar Service.

Calendar Store Identifier

Also known as "CSID". The identifier for an individual calendar store. A CSID consists of the user, password, host and port portions of a "Common Internet Scheme Syntax" part of a URL, as defined by [[RFC1738](#)].

Calendar Store Components

Components maintained in a Calendar Store that are not part of any calendar.

Calendar Store Properties

Properties maintained in a Calendar Store that are not part of any calendar.

Calendar User

The entity that is associated with the credentials presented by the Calendar User Agent to the Calendar Store.

Calendar User Agent

Also known as "CUA". The CUA is the software client operated by the calendar user.

Calendaring and Scheduling System

The computer sub-system that provides the services for accessing, manipulating calendars and scheduling calendar components.

Connected Mode

A mobile computing mode where the CUA is directly connected to the calendar store.

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.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

5

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

Designate

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

Disconnected Mode

A mobile computing mode where a CUA can be disconnected from a calendar store. When the CUA is disconnected, it is in the disconnected mode.

Fan Out

The process by which a calendar store forwards scheduling operations to the attendees not associated with the current calendar.

Hierarchical Calendars

A CS feature where calendars may contain sub-calendars. The top-most calendar in a hierarchy of calendars is called the root calendar. There may be multiple root calendars in a single CS. Within a hierarchy of calendars, all sub-calendars have a calendar with a parent topographical relationship. In addition, sub-calendars may have sub-calendars (child topographical relationship). In addition, the sub-calendar may have one or more calendars with a sibling topographical relationship.

High Bandwidth Connection

A communications connection supporting high transfer rates; transfer rates commonly found within a LAN.

Local Store

A store which is on the same hardware as the CUA.

Low Bandwidth Connection

A communications connection supporting slow transfer rates; transfer rates commonly found in modem technology.

Relative Calendar Identifier

Also known as "Relative CalID". A relative identifier for an individual calendar in a calendar store. A Relative CalID consists of the portion of the "scheme part" of a Qualified CalID, following the Calendar Store Identifier. This is the same as the "URL path" of the "Common Internet Scheme Syntax" portion of a URL, as defined by [[RFC1738](#)].

Qualified Calendar Identifier

Also known as "Qualified CalID". A complete Internet identifier for a specific calendar. A Qualified CalID consists of a CAP specific "scheme" and "scheme part" portion of a URL, as defined in [[RFC1738](#)]. A Qualified CalID are written only with the graphic printable characters of the US-ASCII coded character set.

Remote Store

A store which is not on the same hardware as the CUA.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

6

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

Sub-calendars

Calendars that are contained within other calendars in a hierarchical relationship.

3. Relationship To iCalendar, iTIP and iMIP/iRIP

The CAP data elements MUST be based on the calendar architecture set forth in [[ICAL](#)]. More precisely, CAP will define an Internet protocol for accessing a CS that consists of one or more calendars, each consisting of numerous [[ICAL](#)] components. The definition of CAP might necessitate adding components, properties, parameters and other elements beyond those defined in [[ICAL](#)]. These additions MUST be defined in a manner consistent with and upwards compatible to the data model defined by [[ICAL](#)].

Where it makes practical sense, CAP SHOULD make use of the scheduling workflow defined by [[iTIP](#)].

CAP will not require that [[iMIP](#)] or [[iRIP](#)] MUST be used to communicate with other calendar users.

4. Requirements

4.1 **Basic Usage**

CAP MUST specify:

- How to provide a standard protocol to allow a CUA to interoperate with a CS.
- Using only CAP, a CUA MUST be able to access and manipulate a calendar. CAP MUST provide the complete, if minimal, functionality that allows a CUA to access a calendar.
- How to allow a CUA to be developed independently of a CS and a CS independently of a CUA.
- How to allow an organization to have a mixture of CUAs and CSs from different vendors. With CAP, any CUA in the organization MUST be able to interoperate with any CS.
- How to allow for a single CUA to access multiple CSs or calendars.

For example, a CUA MUST be able to create calendar components on multiple calendars and/or retrieve calendar components from multiple calendars.

- How to allow for a rich featured CUA.

For example, allow a CUA to be part of a client which offers rich calendaring, scheduling and related functionality, with the possibility of extending from concepts and services offered by CAP, while still using CAP to access a calendar.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

7

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

- How to extend the base features in CAP.
- How to allow for interpersonal applications to be calendar enabled.

For example, a chat application MUST be able to offer a calendar button to create an event between the two or more parties.

- How to allow for non-interpersonal applications to be calendar enabled.

For example, a reservation system MUST be able to retrieve and update calendar components from a calendar. Other examples of non-interpersonal applications include event planning, resource

scheduling, and Human Asset Management (HAM).

- How to allow for CUAs without local store and with minimal memory.

For example, a cell phone MUST be able to act as a CUA.

- How to allow for CUAs with local store that can be kept synchronized with the remote store.

For example, a robust featured application might act as a CUA and also provide extra functionality using the local store.

- How to allow for CUAs over disconnected, low-bandwidth connections.

For example, a PDA MUST be able to act as a CUA and interoperate with a CS over a low-bandwidth wireless connection.

- How to allow for CUAs over high bandwidth connections.

For example, a PC MUST be able to act as a CUA and interoperate with a CS over a high-speed Ethernet connection.

4.2 Infrastructure

This section defines a set of infrastructure requirements for CAP.

4.2.1 Connection Models

CAP MUST support a number of CUA-to-CS connection models. In particular, CAP MUST allow connected and disconnected operation.

CUAs and CSs often support the "connected model", where clients maintain long-term connections to servers. Because of this, CAP MUST specify how the connection between a CUA and a CS can be maintained.

CAP MUST NOT prevent servers from dropping client connections, particularly idle client connections. CAP MUST provide some ways for clients to indicate that they would like to stay connected, or that

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

8

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

they would like to drop the connection after the current request/response. These requirements are open issues.

4.2.2 Store Location Models

CAP MUST allow CUAs with local stores and CUAs without local stores.

SINGLE REMOTE CS

CAP MUST support the store location model where a CUA needs to retrieve everything from a single remote CS. In this model, the CUA does not have a local CS. Instead, a single CS resides remotely and is accessed through CAP. The client MUST always establish the CAP connection in order to function.

MULTIPLE REMOTE CS

CAP MUST support the model where a CUA has no local CS, but needs to retrieve everything from multiple remote CS. In this model, the CUA does not have any local CS. Instead, multiple CSs reside remotely and are accessed through CAP. The client MUST always establish one or more CAP connections in order to function.

SINGLE LOCAL, SINGLE REMOTE CS

CAP MUST support the model where a CUA can retrieve everything from the local store and periodically synchronize the local store with a single remote CS. When there is no CAP connection, the CUA can still function. When the CAP connection is re-established, the CUA can update the remote CS with information modified on the local CS while it was disconnected. In addition, the CUA can update its single local CS with information that was modified on the remote CS while the CUA was disconnected.

SINGLE LOCAL, MULTIPLE REMOTE CS

CAP MUST support the model where a CUA can retrieve everything from a local CS and periodically synchronize with multiple remote CS. When the CAP connection is not established, the CUA can still function. When the CUA re-establishes a connection with each remote CS, it can update the remote CS with information modified while disconnected. In addition, the CUA can update the single local CS with information that was modified on remote CS while the CUA was disconnected.

MULTIPLE LOCAL, MULTIPLE REMOTE CS

CAP MUST support the model where a CUA can retrieve everything from multiple local CS and periodically synchronize them with multiple remote CS. When a CAP connection is not established, the CUA can still function. When the CUA re-establishes a connection with each remote CS, it can update the remote CS with information modified while disconnected. In addition, the CS associated with the CAP connection can update any of the local CS with information that was

modified while it was disconnected from the remote CS. Multiple local and remote CS can be updated in sequence.

When remote and local CSs are involved, it MUST be possible to identify conflicts between the local and remote CSs. It is the responsibility of the CUA to resolve conflicts.

"Modified information" includes new, changed and deleted components, as well as properties on calendars and components.

4.2.3 Calendar Stores

The diagram below describes the data objects maintained in a CS. CAP defines how data in the CS is manipulated. The data consists of:

- Calendar Store properties, e.g. local time zone
- Calendars, e.g. a user's calendar
- Calendar properties, e.g. a user name
- Calendar components, e.g. an iCAL VEVENT
- Component properties, e.g. DTSTART on a VEVENT
- Property attributes, e.g. TZID on DTSTART
- Access control rights (deferred)

A calendar may contain other calendars to form a hierarchy. Every calendar has its own properties.

CAP does not define users. CAP does not define any default or implied relationship between a user and a calendar. Users, via a CUA, authenticate themselves to a CS to access information. All access is subject to access control.

Editor Note: Insert calendar store diagram here. In the interim refer to <http://www.imc.org/ietf-calendar/csmodel.jpg>

Calendars are always referenced by their CalIDs. Open issue: whether calendars can also be referenced by their hierarchical name.

CalIDs are used in all operations

Attendees values can be Qualified CalIDs of varying schemes. For example:

```
ATTENDEE[...]:mailto:a@example.com
ATTENDEE[...]:irip://cal-1.example.com/b
ATTENDEE[...]:cap://calendar.example.com/c
```

Editor Note: The MAILTO URL is illustrating an "iMIP" URL.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

10

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

CAP MUST provide a way for the CUA to determine whether or not the CS supports optional or varying capabilities. For example:

- Is a particular optional feature (like fan out, or calendar access rights) supported?
- Determine what, if any, limitation the CS imposes on unbounded recurrence rules.
- Provide a way for the CUA to determine what, if any, limitations the CS imposes on date/time values. For example, there may be dates in the past and future beyond which the CS cannot represent.

4.2.4 Exception Reporting

The granularity of exception report can be at the level of the calendar, the calendar component, component property, or property attribute as appropriate.

CAP MUST provide a way to determine the results of delayed operations. Delayed operations arise from the use of other protocols (iMIP, iRIP), which may take a long time to resolve. Delayed operations have a return code and may have associated calendar data. As an example, an iTIP request for free/busy information may result in the delivery of a VFREEBUSY component. A CUA MUST be able to use CAP to retrieve the VFREEBUSY component. If the operation failed, the CUA MUST be able to determine the reply code. It is an open issue whether CAP MUST support delayed operations and what that means, and how results are returned.

4.3 Operations on a Calendar

A calendar is assumed to reside on a CS. CAP MUST specify:

- How a CUA can create a calendar or sub-calendar.

For example, sub-calendars might be used to organize calendar information based on criteria defined by the CUA.

- How a CUA can rename a calendar or sub-calendar.

For example, the CUA may decide to change the name of a calendar after the calendar has been created.

- How a CUA can delete a calendar or sub-calendar.

When deleting a calendar, all sub-calendars of the calendar MUST be deleted. This is an open issue.

- A CUA can delete a calendar regardless of whether or not the calendar has any entities in it.
- A CUA can set and retrieve properties of a calendar or sub-calendar on a CS.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

11

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

This should include the ability to set and retrieve standard and custom properties, the time zone of a calendar, and the operational hours of a calendar.

For example, the CUA may want to schedule an event based on the operational hours of a calendar.

4.3.1 Deferred Requirements for Operations on a Calendar

CAP is not required to specify:

- How to copy a calendar on a CS or between CSs.
- How a group operations on a set of calendars.
- How to undelete a calendar.
- Auto processing on scheduled components in a calendar that need action.

For example, if someone tries to create a meeting on a calendar for a user who is on vacation, the CS may automatically delegate the meeting to another user.

4.4 Component Management

CAP MUST specify, subject to access control:

- How the CUA can create a component (event/to-do/journal) on a calendar (direct booking)
- How the CUA can create a component on another user's calendar

This capability permits the CUA to create calendar components on another calendar, other than their own, but does not necessarily give them the capability to read or modify calendar components.

- How the CUA can modify a component on a given calendar
- How the CUA can delete a component from one or more calendars
- The requirement to delete from multiple calendars might be met with separate requests.
- How the CUA can modify, add or delete an arbitrary component property within a specified calendar.

For example, modify the SUMMARY property on a calendar component.

- How the CUA can modify a parameter of a multi-valued component property within a specified calendar.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

12

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

For example, modify the PARTSTAT parameter on an ATTENDEE property associated with a particular attendee on a calendar component.

- How the CUA can add or modify attachment properties on a specified component.

The attachments may be large, and the CUA ought not have to resend the entire component.

- How the CUA can send an attachment to the CS.

The attachment would be appended to a particular component.

- How the CUA can remove a specified attachment from the CS and its property from the specified component.

- How hierarchical name can be used for all operations.

This requirement indicates that any calendaring or scheduling operation performed on a component or calendar by specifying the UID or CSID, MUST also be available by specifying the hierarchical name. The two exceptions are the request for the hierarchical name given the UID or vice versa.

Open Issue: This is still open to discussion.

CAP MUST also allow the CUA to do synchronization of two calendars to which it has access.

4.4.1 **Recurrence**

CAP MUST specify, subject to access control:

- How the CUA can retrieve or delete a calendar component based on the UID, and an optional RECURRENCE-ID.

For example, retrieve the single instance of a recurring meeting by specifying both the UID for the recurring meeting and the RECURRENCE-ID of the instance.

- How the CUA can modify or delete an instance of a recurring component, or all recurrences.
- How the CUA can modify all instances of a recurring component that occur before a specified date, or after a particular date.

Open issue: whether the following requirements related to expansion of recurrence are deferred.

- How the CUA can request the CS to send down components with or without expansion of recurring calendar components.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

13

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

- CAP MUST permit the CS to refuse this request, so that if the CS normally provides expansions of recurring calendar components, it can refuse the request to not expand. However, the CS MUST terminate the expansion eventually.
- How the CUA can find out, and perhaps change, the length of time for which recurring calendar components are expanded.
- How the CUA can retrieve, and perhaps set, the period for or number of recurrences expanded on a recurring component.

This could be 0, infinity, a specific DTEND, or a consistent length of time from "today" into the future.

4.4.2 **Calendar and Component Policies**

CAP MUST specify:

- How the CUA tells the CS to prevent conflicts (double booking) on a calendar

The scenario is that a calendar could be marked for "allow no conflicts", and the CS would automatically prevent this. This might apply to direct booking or scheduling or both.

- How operational hours for a calendar are enforced

Each calendar MUST have a property for operational hours, and CAP MUST specify how the CUA tells the CS to enforce those operational hours. This means that the CS prevents components from being added in a manner that violates the operational hours set by the user. CAP MUST specify how policy enforcement can be overridden by the owner of the calendar. CAP MUST specify whether this includes alarms. This might apply to direct booking or scheduling or both.

Open issue: Whether the ability to set a policy of turning down requests that exceed a maximum duration (or length of time between DTSTART and DTEND) is a requirement.

4.4.3 Deferred Requirements for Component Management

CAP is not required to specify:

- Undelete and purge deleted calendar entries.
- Modify inline attachment to URL; provide URL to fetch.
- Merge or aggregate more than one calendar.
- Lock access to calendar.
- Delayed or asynchronous transactions.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

14

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

- The CUA requesting a size limit before retrieving components from the CS. The CS might send partial components if the component's size exceeded the limit. CAP would permit the CS to refuse the request or give its preferred value.
- The CS synchronizing 2 calendars.
- How the CUA tells the CS to prevent conflicts with respect to an individual component in the calendar. The scenario for this is that a component could be marked for "allow no conflicts", and the CS would automatically prevent a new component from being added in a way which would cause a conflict. This might apply to either direct booking or scheduling or both.
- How the CUA can create multiple entries on many CSs.

This may occur in one or more operations, perhaps using

different calendar protocols (CAP, iRIP, iMIP).

4.5 **Lookup, Query and Discovery**

Lookup includes the capability to list things. Querying enables searching and filtering features. Discovery covers requests for information such as what features are supported by the CS.

4.5.1 **Lookup**

CAP MUST specify, subject to access control:

- How the CUA can list calendars in a single CS, by fetching all the top level CSIDs of the CS.
- How the CUA can list all the sub-calendars within a given calendar.
- How the CUA can list all component UIDs in a calendar.
- How the CUA can retrieve all the free/busy data for a calendar.
- How the CUA can retrieve a list of properties for a component or a set of components.

For this requirement, the set of components is defined either as all the components in a calendar, or where the set of components is defined by a search query (search query requirements are elaborated in the Query Capabilities section). For example, the CUA could ask for the DTSTART, DTEND and SUMMARY properties for all components with DTSTART later than 9:00 p.m. today.

4.5.2 **Query Capabilities**

CAP MUST allow the CUA to retrieve CalIDs, component UIDs or component hierarchical names (open issue) from a given calendar, using queries which specify:

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

15

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

- How the CUA can retrieve data from multiple calendars.

This requirement allows the CUA to display multiple calendars to the user. This requirement does not constrain this functionality to a single request as it may be done in multiple requests.

- How the CUA to retrieve any named property for a calendar or component.

For example, the CUA can retrieve the SUMMARY property for a given component.

- Property value equivalence query (e.g., equal to)

For example, such as matching the organizer with a specified user, or matching the location with a particular conference room. This should work for all properties.

- Pattern-matching string comparison query (e.g., contains)

The pattern-matching query MUST be applied to a string property such as a "name" property. The response MUST include a list of calendar (or component) identifiers and MAY include property values for those items. The CUA MUST be able to request which properties (or the entire component) to retrieve.

- Property value comparison query (e.g., greater than, less than)

This requirement includes only those properties for which a comparison query is possible. This list of properties MUST include DTSTART, DTEND, DURATION. For example, this allows the CUA to ask for all components that begin later than (greater than) the specified time.

This requirement MUST be met in a way which allows the CUA to discover which alarms will trigger in a given period.

- Multiple property value comparisons combined using Boolean elements (e.g., logical and, logical or, negation).

The CUA MUST be able to discover which components have durations which occur at least partially in a given range of time, either by retrieving the list of UIDs or by retrieving all of the components properties. This requirement allows the CUA to discover all components during a particular day or other time period. This includes instantaneous and all-day calendar items. This requirement does not state that the components MUST be retrieved in their entirety in the same interaction as the query; the query and the component retrieval can be separate interactions between the CUA and the CS.

CAP MUST allow synchronization, meaning at a minimum that the CUA is able to find and retrieve new, modified or deleted entries for a given time period. The CUA MUST be able to find out which entries

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

16

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

have been added, modified or deleted since it last synchronized, in

order to operate in disconnected mode. This requirement may be satisfied using the query requirements already defined.

4.5.3 Specific Queries

These are specific scenarios required by calendaring operations which may require special attention to ensure that they can be done with the querying and lookup functionality of CAP.

CAP MUST allow, subject to access control:

- The CUA to retrieve the CSID for a calendar specified by giving its hierarchical name, and vice versa.

This requirement means that hierarchical calendar names MUST be unique on the CS.

Open Issue: This is an open issue.

- The CUA to discover conflicts given a list of calendars and a time period

This feature is to allow the CUA to schedule a group of attendees for a meeting at a particular time; the CUA MUST be able to find out if those attendees are free at that time. This requirement does not specify whether the discovery is done mainly on the CS or on the CUA. For example, the CUA could ask for the free/busy data for each calendar during the period and then determine conflicts itself, or the CUA could ask the CS which attendees have conflicts during the period. Either approach would satisfy this requirement.

- The CUA to discover which calendar components need action.

iCalendar entries which need action include scheduled components that have not yet been accepted or declined. The CUA needs to know this list in order to present the items to the user to resolve them.

4.5.4 Discovery

CAP MUST specify:

- How the CUA can query the calendar components which are supported on a named calendar.
- How the CUA can query the names of all properties which exist on a given calendar, or the properties which exist on a given component.
- How the CUA can query the names of component properties supported by a CS.

Internet Draft

CAP Requirements

May 27, 1999

- How the CUA can query the time zones supported by the CS

4.5.5 Deferred Requirements for Search

CAP is not required to specify:

- How to roll up free/busy information for a number of calendars (perhaps sub-calendars) into a single free/busy component.
- How to fetch all components marked for deletion in a certain range. This could include components somehow marked for deletion but not yet purged from the CS.
- How the CUA can determine whether the CS supports multiple reads/writes in the same operation.

This would include, for example, the ability to name a number of calendars to add a component to in a single operation

4.6 Security

CAP MUST specify:

- How the CUA authenticates itself to the CS (not the calendar).

Authentication to the CS is required for all access to the CS. The CS MUST be able to uniquely identify each user for the purposes of authentication and authorization.

- How anonymous access to calendars is specified (if allowed).
- How the CUA authenticates to CS using SASL.
- How the CUA and the CS negotiate encryption mechanism for a secure connection.
- How calendars can be made secure from unwanted access and false entries.
- How the CUA and CS can specify denial of service to another calendar user.

4.7 Designates and Delegation

CAP MUST specify, subject to access control:

- How a list of designates is created.

- How the CS knows that a user is a valid designate, through authentication or some other mechanism.
- How the CUA can delegate to another calendar user.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

18

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

4.8 Notification (deferred)

There are no notification features that are required in CAP. Notification is deemed not to be a requirement, because:

- A CUA can poll the CS for new/changed/deleted components, including new ITIP messages. This functionality is needed to support synchronization.
- A CUA can handle alarms on components itself.

The notification features that were considered but have been deferred include:

- Notify the CUA when a change is made to a calendar.
- Notify the CUA when an alarm triggers.
- Notify the CUA when a calendar component NEEDS-ACTION.
- Notify when a CUA attempts to access a calendar, delete, modify, or add a calendar component.

4.9 Access Control (deferred)

The ability to get and set access control data on calendars and calendar components is useful, but beyond the scope of the initial CAP specification. A CS may apply access control entries to calendars and components, but CAP need not specify how these are to be set and retrieved. Access control entries could be set and retrieved by administrators on the CS through another protocol. Additionally, a CS can enforce the class of each component, by restricting access to "confidential" and "private" components, without any design in CAP to allow this.

Access control requirements that were considered but have been deferred include:

- CUA query the calendar access rights for the currently authenticated calendar user.

- CUA grant/deny designate rights to a given calendar user.
- CUA removes OR denies all rights to a given calendar for given calendar users.
- CUA grants free/busy view rights on a given calendar to a calendar user.
- CUA grants full access rights on a given calendar to a calendar user.
- CS performs calendar access rights inheritance on sub-calendars.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

19

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

- CUA grants/denies access rights to individual properties of individual components.

For example, user A can see subject of event B, or user A can set time of event B.

- CUA can set operation limits for user A on calendar B.

For example:

- (a) set time limit C on events for user D can schedule on my calendar,
- (b) set limit number of events by user E, or
- (c) grant limited access on a calendar such that a user F can only create events during specified hours.

4.10 Resource Groups and Pools (deferred)

A specification for how to handle resource groups and pools is not a requirement. These kinds of features will be addressed in a separate, later draft.

A group has a list of members, all of whom should be included in any scheduled calendar component. For example, a group could contain a particular conference room and a particular projector, both of which should be scheduled for a meeting. Or a group could contain a number of people working together, all of whom should be scheduled for group meetings. When a group is included as an attendee, the CS MUST schedule each of the members of the group.

A pool has a list of members, all of which are identical for the purposes of scheduling, and only one of them should be scheduled.

When a pool is included as an attendee, the CS MUST schedule the number of them that was requested. For example, a pool could contain a number of VCRs, any of which can be scheduled for meetings, when only one VCR is usually needed.

Resource group and pool features that were considered but were deferred include:

- CUA can create a list of CSIDs which is a group or pool.
- CUA can request to add a component to every calendar for every CSID in a group
- CUA can send a single schedule request to the CS, where the attendee list includes a group. The CS then fans out the request. Fanning out might be achieved by forwarding the request to all members of a group, or by placing the request on the calendar of every member of the group. The members of the group may be on multiple CSs.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

20

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

- CUA can schedule one CSID by scheduling one instance from a pool.
- Each group/pool itself has properties.
- Access restrictions can be set on a group/pool, to restrict who can get/set properties and who can schedule the group/pool.

4.11 CAP Non-requirements

CAP is not required to be:

- A client-to-client protocol.
- A server-to-server protocol.

For example, it would not specify how server-to-server updates of calendaring or scheduling operations are accomplished.

The initial version of CAP is not required to specify:

- How to do a full administration CUA, e.g., add user, delete user, move user, archive calendar, delete user, change user/calendar name.
- Inheritance of calendar access rights, property values, etc.

5. Open Issues

Open issues include the following:

- Whether CAP should specify how clients can request their connections to be kept open, and whether servers can drop connections at any time.
- Whether calendars can also be referenced by their hierarchical name.
- Is a particular optional feature (like fan-out, or calendar access rights) supported?
- Determine what, if any, limitation the CS imposes on unbounded recurrence rules.
- Whether CAP MUST support delayed operations and what that means, and how results are returned.
- Whether all sub-calendars of a deleted calendar MUST be deleted.
- Whether the ability to set a policy of turning down requests that exceed a maximum duration is a requirement.
- Whether expansion of recurrences is required.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

21

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

- Whether or not CAP operations support bounding the latency of responses to operations.

6. Acknowledgments

The following have provided input in the drafting of this memo:

Lisa Lippert, Surendra Reddy, Richard Shusterman.

7. Bibliography

[ICAL] "Internet Calendaring and Scheduling Core Object Specification - iCalendar", [RFC 2445](http://www.ietf.org/rfc/rfc2445.txt), November 1998, <http://www.ietf.org/rfc/rfc2445.txt>.

[ITIP] "iCalendar Transport-Independent Interoperability Protocol (iTIP) : Scheduling Events, Busy Time, To-dos and Journal Entries ", [RFC 2446](http://www.ietf.org/rfc/rfc2446.txt), November 1998, <http://www.ietf.org/rfc/rfc2446.txt>.

[IMIP] "iCalendar Message-based Interoperability Protocol (iMIP), [RFC](#)

[2447](http://www.ietf.org/rfc/rfc2447.txt), November 1998, <http://www.ietf.org/rfc/rfc2447.txt>.

[IRIP] "iCalendar Message-based Interoperability Protocol (iRIP),
Internet-Draft, November 1998, , <ftp://ftp.ietf.org/internet-drafts/draft-ietf-calsch-irip-02.txt>.

8. Authors' Address

The following address information is provided in a vCard v2.1,
Electronic Business Card, format.

```
BEGIN:VCARD
VERSION:3.0
FN:Andre Courtemanche
N:Courtemanche;Andre
ORG:CS&T
ADR;TYPE=WORK,POSTAL,PARCEL;;;3333 Graham Boulevard;Montreal;QC;H3R
  3L5;Canada
TEL;TYPE=WORK,MSG:+1-514-733-8500
TEL;TYPE=WORK,FAX:+1-514-733-8788
EMAIL;TYPE=INTERNET:andre@cst.ca
END:VCARD
```

```
BEGIN:VCARD
VERSION:3.0
FN:Frank Dawson
N:Dawson;Frank
ORG:Lotus Development Corporation
ADR;TYPE=WORK,POSTAL,PARCEL;;;6544 Battleford Drive; Raleigh;
  NC;27613-3502;USA
TEL;TYPE=WORK,MSG:+1-617-693-8728
TEL;TYPE=WORK,FAX:+1-617-693-5552
EMAIL;TYPE=INTERNET:Frank_Dawson@Lotus.com
URL:http://home.earthlink.net/~fdawson
```

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small

22

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

```
END:VCARD
```

```
BEGIN:VCARD
VERSION:3.0
FN:Tony Small
N:Small;Tony
ORG:Microsoft Corporation
ADR;TYPE=WORK,POSTAL,PARCEL;;;One Microsoft Way; Redmond;WA;
  98052-6399;USA
TEL;TYPE=WORK,MSG:+1-206-703-2190
TEL;TYPE=WORK,FAX:+1-206-936-7329
```


EMAIL;TYPE=INTERNET:tonysm@Microsoft.com
END:VCARD

BEGIN:VCARD
VERSION:3.0
FN:Steve Mansour
N:Mansour;Steve
ORG:Netscape Communications Corporation
ADR;TYPE=WORK,POSTAL,PARCEL;;;501 East Middlefield Road;Mountain
View;CA;94043;USA
TEL;TYPE=WORK,MSG:+1-650-937-2378
TEL;TYPE=WORK,FAX:+1-650-937-2103
EMAIL;TYPE=INTERNET:sman@netscape.com
END:VCARD

BEGIN:VCARD
VERSION:3.0
FN:Peter O'Leary
N:O'Leary;Peter
ORG:Amplitude Software Corp.
ADR;TYPE=WORK,POSTAL,PARCEL;;;185 Berry St. Suite 4700; San
Francisco;CA;94107;USA
TEL;TYPE=WORK,MSG:+1-415-659-3511
TEL;TYPE=WORK,FAX:+1-415-659-0006
EMAIL;TYPE=INTERNET:poleary@amplitude.com
END:VCARD

BEGIN:VCARD
VERSION:3.0
FN:Doug Royer
N:Royer;Doug
ORG:Sun Microsystems
ADR;TYPE=WORK,POSTAL,PARCEL;;;901 San Antonio Road, MS MPK17-105;
Palo Alto;CA;94303;USA
TEL;TYPE=WORK,MSG:+1-650-786-7599
EMAIL;TYPE=INTERNET:doug.royer@sun.com
END:VCARD

This work is part of the Internet Engineering Task Force Calendaring
and scheduling Working Group. The chairmen of that working group are:

BEGIN:VCARD
VERSION:3.0

Courtemanche,Dawson,Mansour,O'Leary,Royer,Small

23

Expires November 1999

Internet Draft

CAP Requirements

May 27, 1999

FN:Anik Ganguly
N:Ganguly;Anik

ORG:Open Text, Inc.
ADR;TYPE=WORK,POSTAL,PARCEL;;;38777 West Six Mile Road Suite 101;
Livonia;MI;48152;USA
TEL;TYPE=WORK,MSG:+1-734-542-5955
EMAIL;TYPE=INTERNET:ganguly@acm.org
END:VCARD

BEGIN:VCARD
VERSION:3.0
FN:Robert Moskowitz
N:Moskowitz;Robert
EMAIL;TYPE=INTERNET:rgm-ietf@htt-consult.com
END:VCARD

9. Full Copyright Statement

Copyright (C) The Internet Society (1998). 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.

Courtemanche, Dawson, Mansour, O'Leary, Royer, Small