Network Working Group   Tony Small, Microsoft Corporation

INTERNET-DRAFT   Denis Hennessy, ISOCOR
Calendaring and Scheduling Working Group
                        Frank Dawson, Lotus

Expires six months from   June 11, 1998
                Calendar attributes for vCard and LDAP

                    draft-ietf-calsch-locating-00.txt

 Abstract

When scheduling a calendar entity, such as an event, it is a
prerequisite that an organizer has the calendar address of each
attendee that will be invited to the event. Additionally, access to an
attendee's current "busy time" provides an a priori indication of
whether the attendee will be free to participate in the event.
In order to meet these challenges, a calendar user agent (CUA) needs a
mechanism to locate (URI) individual user's calendar and free/busy
time.

This draft defines three mechanisms for obtaining a URI to a user's
calendar and free/busy time. These include:

- Manual transfer of the information;
- Personal data exchange using the vCard format; and
- Directory lookup using the LDAP protocol.

1. URIs

This draft defines four classes of URIs. URIs are more useful if it is

understood what the URIs point to. Here is a brief description:

1.1. Free/Busy URI (FBURL)

The free/busy URI is defined to be a transport independent location where a client can obtain information about when a user is busy. At the present time, this URI only points to busy time data. Future revisions of this specification may provide for the extended capability of publishing free time data.

If a calendaring and scheduling client (i.e., CUA) were to retrieve data from this location using FTP or HTTP, it would get back an iCalendar object [10] containing one or more "VFREEBUSY" calendar components. If a MIME transport is being used, the response will be contained within a "text/calendar" MIME body part as specified in the iCalendar specification [10]. For example:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
METHOD:PUBLISH
BEGIN:VFREEBUSY
ATTENDEE:MAILTO:jane_doe@host1.com
DTSTART:19971013T050000Z
DTEND:19971124T050000Z
DTSTAMP:19970901T083000Z
FREEBUSY:19971015T133000Z/19971015T180000Z
FREEBUSY:19971015T190000Z/19971015T220000Z
FBURL:http://www.host.com/calendar/busy/jdoe.ifb
END:VFREEBUSY
END:VCALENDAR
```

The amount of busy time data pointed to by the FBURL will generally be pre-determined; for example one month of free/busy inforation. As a guideline, it is recommended that the previous six weeks of busy time data be published at the location associated with the FBURL. If this URI points to a file resource, it is recommended that the file extension be "ifb" to distinguish it from an arbitrary iCalendar object.

1.2 Calendar Address URI (CALADRURI)

The calendar address URI is defined to be a transport independent

communication end-point for a user.   The organizer's calendaring and
scheduling client (ie. CUA) would use this URI to determine where to
send an event request when organizing a meeting with a particular
attendee.

If the user prefers to receive event requests via iMIP, then the user
would provide a "mailto" URI containing the user's e-mail address. [13]
For example:

"mailto:user@host1.com"

The URI for an iRIP user is yet to be defined, but that is another
possible URI value in this property. [14]

1.3. Calendar Access URI (CAPURI)

The Calendar Access URI is defined to be a protocol independent
location from which a calendaring and scheduling client (i.e., CUA) can
communicate with a user's entire calendar.

The semantics for using this URI as an access protocol locator are yet
to be defined by the IETF CALSCH Working Group. This will be addressed
in the "Calendar Access Protocol" specification.

1.4 Calendar URI (CALURI)

The Calendar URI is defined to be a protocol independent location from
which a calendaring and scheduling client (ie. CUA) can retrieve an
entire copy of a user's calendar. Retrieving data from this URI
obtains a published "snapshot" of the user's calendar.

HTTP URI -- If the URI is an HTTP URI, then the content returned with a
GET should be a "text/calendar" MIME body part containing one or more
iCalendar object.

FTP URI -- If the URI is an FTP URI, then the resource pointed to
should be a file with an "ics" file extension containing one or more
iCalendar objects.

1.5. Default URIs

There are many cases where a user may have more than one calendar. In
these cases, a user may have multiple URIs, each URI pointing to a
calendar or free/busy data.

To make the case of multiple calendars simpler for clients, the concept
of the "default" calendar is introduced. A "default" calendar is one

that the user has designated as the calendar that other users should
look at when accessing the user's calendar, or retrieving the user's
free/busy time.

The default calendar may, in fact, include rolled-up information from
all the user's other calendars. The other calendars may only exist for
organizational purposes.

2. Distribution

These four URIs provide valuable pointers to calendaring and scheduling
data that other users need in order to know when to schedule meetings,

etc. There are several possibilities on how users can communicate
these URIs to other users. The following section outlines how these
URIs can be distributed to other users.

2.1. Manual Transfer

The simplest way to obtain these URIs is for a user to communicate the
URIs using some out-of-band mechanism such as verbally, or in an e-mail
message, or by printing these URIs on a paper business card.

When using this mechanism, the user obtains these URIs using an out-of-
band mechanism and then enters these URIs into their calendaring
software manually.

2.2. Personal Data Exchange Using A vCard

A more sophisticated way to obtain these URIs is for users to publish
vCards containing these URIs. The vCard object can be transferred
between one another. Since many e-mail clients allow a user to
automatically include a vCard with every message that the user sends,
this provides a simple, transparent way for a user to distribute their
calendaring and scheduling URIs.

On the receiving end, an e-mail client that provides an integrated
vCard database can provide a way to lookup calendaring URIs for users
whose vCards are stored locally.

2.2.1. vCard Schema Extensions

Since the vCard [8] specification doesn't specify how to encode
calendaring URIs in a vCard, this section is provided as an extension

to vCard which specifies how to encode calendaring URIs within a vCard.

Inside a vCard object, four new properties are defined: "CALURI",
CAPURI, _CALADRURI_, and "FBURL", as defined above.

Any vCard can have one or more of these properties, each representing a
calendar or free/busy time that is associated with the user.

One of these properties can be designated as the "default" by adding
the "PREF" parameter.

Here is a simple example of a vCard containing a "FBURL" and a
"CALURI".

```
BEGIN:VCARD
VERSION:3.0
FN:Alec Dun
N:Dun;Alec
```

```
ORG:Microsoft Corporation
ADR;WORK;POSTAL;PARCEL:;;One Microsoft Way;
   Redmond;WA;98052-6399;USA
TEL;WORK;MSG:+1-206-936-4544
TEL;WORK;FAX:+1-206-936-7329
EMAIL;INTERNET:user@host1.com
CALADRURI;PREF:mailto:user@host1.com
CALURI;PREF:http://cal.host1.com/user/cal.ics
FBURL;PREF:http://cal.host1.com/user/fb.ifb
CALURI:http://cal.company.com/projectA/pjtA.ics
FBURL:http://cal.company.com/projectA/pjtAfb.ifb
END:VCARD
```

2.2.1.1 FBURL Property IANA Registration

```
To: ietf-mime-directory@imc.org
   Subject: Registration of FBURL type for text/directory MIME type
        vCard profile.
Type name: FBURL
   Type purpose: To specify the URI for a user's busy time in a vCard
        object.
Type encoding: 8bit
Type value: A single URI value.
   Type special notes: Where multiple FBURL properties are specified,
        the default FBURL property is indicated with the PREF
        parameter. The FTP or HTTP type of URI points to an iCalendar
        object associated with a snapshot of the last six weeks of the
        user's busy time data. If the iCalendar object is represented
        as a file or document, it's file type should be "ifb".
```

Intended usage: Refer to [section 1.1](#).

Type examples:
  FBURL;PREF:http://www.host1.com/busy/janedoe
  FBURL:FTP://ftp.host.com/busy/project-a.ifb

[2.2.1.2](#) CALADRURI Property IANA Registration

To: ietf-mime-directory@imc.org
    Subject: Registration of CALADRURI type for application/directory
        MIME type vCard profile.
Type name: CALADRURI
    Type purpose: To specify the location to which an event request
        should be sent for the user.
Type encoding: 8bit
Type value: A single URI value.
    Type special notes: Where multiple CALADRURI properties are
        specified, the default CALADRURI property is indicated with the
        PREF parameter.
    Intended usage: Refer to [section 1.2](#).

Type examples:
  CALADRURI;PREF:mailto:janedoe@host.com

[2.2.1.3](#) CAPURI Property IANA Registration

To: ietf-mime-directory@imc.org
    Subject: Registration of CAPURI type for application/directory MIME
        type vCard profile.
Type name: CAPURI
    Type purpose: To specify a protocol independent location from which
        a calendaring and scheduling client (i.e., CUA) can communicate
        with a user's entire calendar.
Type encoding: 8bit
Type value: A single URI value.
    Type special notes: Where multiple CAPURI properties are specified,
        the default CAPURI property is indicated with the PREF
        parameter.
    Intended usage: Refer to [section 1.3](#).

[2.2.1.4](#) CALURI Property IANA Registration

To: ietf-mime-directory@imc.org
    Subject: Registration of CALURI type for text/directory MIME type
        vCard profile.
Type name: CALURI

Type purpose: To specify the URI for a user's calendar in a vCard
              object.
    Type encoding: 8bit
    Type valuetype: A single URI value.
         Type special notes: Where multiple CALURI properties are specified,
              the default CALURI property is indicated with the PREF
              parameter. The property should contain a URI pointing to an
              iCalendar object associated with a snapshot of the user's
              calendar store. If the iCalendar object is represented as a
              file or document, it's file type should be "ics".
         Intended usage: Refer to section 1.4.

    Type examples:
      CALURI;PREF:http://cal.host1.com/calA
      CALURI:ftp://ftp.host1.com/calA.ics

2.3. Directory Lookup Using The LDAP v3 Protocol

Another way to obtain these URIs is to look them up in a directory
using the LDAP protocol.

If an organizer knows an attendee's e-mail address, then using DNS,
the attendee's directory server can be found. The mechanism for this
is described in detail in [7]. From the directory server, the client

can look up the URLs for a user's calendar. Here's a summary of how it
works. For more detail, please see the draft [7].

The client first parses the domain name out from the rfc822 mailbox
name. For the fictitious mailbox "janedoe@host1.com", the domain name
would be "host1.com".

Given the domain name, the client prepends "ldap.tcp" to the domain
name and formulating a host. Next the client retrieves the queries the
DNS server for the SRV record for "ldap.tcp.host1.com". The mechanism
for adding "ldap.tcp" onto the original domain name is described in
detail in [5]. The DNS server returns the IP address for the
associated server for 'ldap.tcp.host1.com'.

Once the IP address for the LDAP server has been obtained, the client
constructs a DN from which to search using the DNS name. In this case,
it would be "DC=host1,DC=COM". The mechanism to construct the DN is
described in detail in [6]. With the IP address and the DN, the client
issues a search request to the server where the attribute named "mail"
[4] "equalityMatch"es the user's email address. From the first
matching entry, client obtains the calendaring and scheduling URLs.

If a user's URIs can be found using directory lookup, they should, in

general, be considered "more up-to-date" than URIs in any vCards that
are stored locally.

[2.3.1]. LDAP Schema Extensions

In order to encode the calendaring URIs in the directory, the following
are defined:

    one object class:

    @ calEntry

    and eight attributes:

    @ calCalURI
    @ calFBURL
    @ calCAPURI
    @ calCalAdrURI
    @ calOtherCalURIs
    @ calOtherFBURLs
    @ calOtherCAPURIs
    @ calOtherCalAdrURIs

The calCalURI contains the URI to a snapshot of the user's entire
default calendar. The calFBURL contains the URI to the user's default
busy time data. The calCAPURI represents contains a URI that can be
used to communicate with the user's calendar. The calCalAdrURI

contains a URI that points to the location to which event requests
should be sent for that user.

The calOtherCalURIs is a multi-valued property containing URIs to
snapshots of other calendars that the user may have. The
calOtherFBURLs is a multi-valued property containing URIs to other
free/busy data that the user may have. The calOtherCAPURIs attribute
is a multi-valued property containing URIs to other calendars that the
user may have. The calOtherCalAdrURIs attribute is a multi-valued
property containing URIs to other locations that a user may want event
requests sent to.

There is no predetermined order to the values in either multi-valued
property.

[2.3.2]. Notation

The notation used in this document is the same as that used in [2].

[2.3.3](). Object Definitions

[2.3.3.1](). calEntry

The Calendar Entry is a class derived from _TOP_ [[2]()], which contains
the four calendaring attributes.


```
    (   1.2.840.113556.1.5.87
        NAME 'calEntry'
        TOP
        AUXILIARY
        MAY (calCalURI calFBURL calOtherCalURIs calOtherFBURLs
calCAPURI calOtherCAPURLs)
     )
```


[2.3.4](). Attribute Definitions

[2.3.4.1](). calCalURI

```
    (  1.2.840.113556.1.4.478
        NAME 'calCalURI'
        EQUALITY caseIgnoreMatch
        SUBSTRING caseIgnoreMatch
        SYNTAX 'Directory String'
        USAGE userApplications
    )
```


[2.3.4.2](). calFBURL

```
    (  1.2.840.113556.1.4.479
        NAME 'calFBURL'
        EQUALITY caseIgnoreMatch
        SUBSTRING caseIgnoreMatch
        SYNTAX 'Directory String'
        USAGE userApplications
    )
```


[2.3.4.3](). calCAPURI

```
    (  1.2.840.113556.1.4.480
        NAME 'calCAPURI'
        EQUALITY caseIgnoreMatch
```

```
          SUBSTRING caseIgnoreMatch
          SYNTAX 'Directory String'
          USAGE userApplications
       )
```

[2.3.4.4](#). **calCalAdrURI**

```
       (  1.2.840.113556.1.4.481
          NAME 'calCalAdrURI'
          EQUALITY caseIgnoreMatch
          SUBSTRING caseIgnoreMatch
          SYNTAX 'Directory String'
          USAGE userApplications
       )
```

[2.3.4.5](#). **calOtherCalURIs**

```
       (  1.2.840.113556.1.4.482
          NAME 'calOtherCalURIs'
          EQUALITY caseIgnoreMatch
          SUBSTRING caseIgnoreMatch
          SYNTAX 'Directory String'
          MULTI-VALUE
          USAGE userApplications
       )
```

[2.3.4.6](#). **calOtherFBURLs**

```
       (  1.2.840.113556.1.4.483
          NAME 'calOtherFBURLs'
          EQUALITY caseIgnoreMatch
```

```
          SUBSTRING caseIgnoreMatch
          SYNTAX 'Directory String'
          MULTI-VALUE
          USAGE userApplications
       )
```

[2.3.4.7](#). **calOtherCAPURIs**

```
       (  1.2.840.113556.1.4.484
          NAME 'calOtherCAPURIs'
```

```
        EQUALITY caseIgnoreMatch
        SUBSTRING caseIgnoreMatch
        SYNTAX 'Directory String'
        MULTI-VALUE
        USAGE userApplications
    )
```

[2.3.4.8]. calOtherCalAdrURIs

```
(   1.2.840.113556.1.4.485
    NAME 'calOtherCalAdrURIs'
    EQUALITY caseIgnoreMatch
    SUBSTRING caseIgnoreMatch
    SYNTAX 'Directory String'
    MULTI-VALUE
    USAGE userApplications
)
```

Authors' Addresses

```
BEGIN:VCARD
VERSION:2.1
N:Small;Tony
FN:Tony Small

ORG:Microsoft Corporation
ADR;WORK;POSTAL;PARCEL:;;One Microsoft Way;
Redmond;WA;98052-6399;USA
TEL;WORK;MSG:+1-425-703-2190
TEL;WORK;FAX:+1-206-936-7329
EMAIL;INTERNET:tonysm@Microsoft.com
CALADRURI:MAIL-TO:tonysm@Microsoft.com
END:VCARD


BEGIN:VCARD
VERSION:2.1
N:Hennessy;Denis
FN:Denis Hennessy
```

```
ORG:ISOCOR
ADR;WORK;POSTAL;PARCEL:;;42-47 Lower Mount St;
Dublin 2;Ireland
TEL;WORK;MSG:+353-1-676-0366
TEL;WORK;FAX:+353-1-676-0856
EMAIL;INTERNET:denis.hennessy@isocor.com
```

```
        CALADRURI:MAIL-TO:denis.hennessy@isocor.com
        END:VCARD


        BEGIN:VCARD
        VERSION:2.1
        N:Dawson;Frank
        FN:Frank Dawson
        ORG:Lotus Development Corporation
        ADR;WORK;POSTAL;PARCEL:;;6544 Battleford Drive;
        Raleigh;NC;27613-3502;USA
        TEL;WORK;MSG:+1-919-676-9515
        TEL;WORK;FAX:+1-919-676-9564
        EMAIL;INTERNET;PREF:Frank_Dawson@Lotus.com
        EMAIL;INTERNET:fdawson@earthlink.net
        CALADRURI;PREF:MAIL-TO:Frank_Dawson@Lotus.com
        CALADRURI:MAIL-TO:fdawson@earthlink.net
        URI:http://home.earthlink.net/~fdawson
        END:VCARD
```

Bibliography

   [1] M. Wahl, T. Howes, S. Kille, 'Lightweight Directory Access
       Protocol (v3)', RFC 2251, December 1997,
       <URL:ftp://ds.internic.net/rfc/rfc2251.txt>

   [2] M. Wahl, A. Coulbeck, T. Howes, S. Kille, 'Lightweight Directory
       Access Protocol (v3): Attribute Syntax Definitions', RFC 2252,
       December 1997, <URL:ftp://ds.internic.net/rfc/rfc2252.txt>

   [3] M. Wahl, A Summary of the X.500(96) User Schema for use with
       LDAPv3',
        RFC 2256, December 1997,
       <URL:ftp://ds.internic.net/rfc/rfc2256.txt>

 [4] The Directory: Selected Attribute Types. ITU-T Recommendation
     X.520, 1993.

 [5] The Directory: Selected Object Classes. ITU-T Recommendation
     X.521, 1993.

 [6] P. Leach `Selecting a server from among many replicas',
     INTERNET DRAFT <draft-ietf-asid-replica-selection-00.txt>,
     February 1997

   [7] P. Leach `Locating Native Internet LDAP Servers',

     INTERNET DRAFT <draft-leach-asid-ldap-locating-00.txt>,
     March 1997

  [8] F. Dawson, T. Howes, 'vCard MIME Directory Profile',
       INTERNET-DRAFT <draft-ietf-asid-mime-vcard-03.txt>,
       July 1997

  [9] A. Gulbrandsen, P. Vixie, "A DNS RR for specifying the location
      of services (DNS SRV)", RFC 2052,
      <ftp://ds.internic.net/rfc/rfc2052.txt>, October 1996.

  [10] F. Dawson, D. Stenerson 'Internet Calendaring and Scheduling
      Core Ojbect Specification (iCalendar)', Nov 1997,
      <URL:ftp://www.ietf.org/internet-drafts/draft-ietf-calsch-ical-

      05.txt

             >

  [11] Mockapetris, P., "Domain Names - Implementation and
      Specification", STD 13, RFC 1035, USC/Information Sciences
       Institute, November 1987.
      <URL:ftp://ds.internic.net/rfc/rfc1035.txt>

  [12] Network Applications Consortium 'Lightweight Internet Person
      Schema', April 1997, <URL:http://www.netapps.org>

[13] F. Dawson, S. Mansour `iCalendar Message-Based Interopability
Protocal (iMIP)', March 1998,

                            <URL: ftp://www.ietf.org/internet-

drafts/draft-ietf-calsch-imip-04.txt

                              >

[14] A.

        Courtemanche, S. Mansour, P. O'Leary `iCalendar Real-Time
Interopability Protocol (

iRIP)', November 1997, <URL:
ftp://www.ietf.org/internet-drafts/draft-ietf-calsch-irip-00.txt>