

**CALCONNECT DOCUMENT CD 0903**

**Type:** Proposal  
**Title:** Freebusy Read URL  
**Version:** 1.0  
**Date:** 2009-04-15  
**Status:** Published  
**Source:** FREEBUSY Technical Committee

This document incorporates by reference the CalConnect Intellectual Property Rights, Appropriate Usage, Trademarks and Disclaimer of Warranty for External (Public) Documents as located at

<http://www.calconnect.org/documents/disclaimerpublic.pdf>.

CalConnect	E. York
	Apple
	April 9, 2009

# Freebusy read URL

## Abstract

This document defines a standardized form of Freebusy read URL to improve interoperability between client and server implementations, while extending the functionality and utility through the use of optional parameters.

---

## Table of Contents

- 1. Introduction**
- 2. Conventions Used in This Document**
  - 2.1. Glossary of terms**
- 3. Freebusy Read URL**
- 4. URL Query Parameters**
  - 4.1. start**
  - 4.2. end**
  - 4.3. period**
  - 4.4. format**
  - 4.5. URL Parameters: Notes**
- 5. URL Identity Parameters**
  - 5.1. token**
  - 5.2. user**
- 6. HTTP Operations**
  - 6.1. Response Codes**
- 7. Examples**
- 8. Template URLs**
- 9. IANA Considerations**
- 10. Security Considerations**
- 11. Acknowledgements**
- 12. References**
  - 12.1. Normative References**

## **12.2. Informative References**

### **§ Author's Address**

---

## **1. Introduction**

**TOC**

The foundation of scheduling is the availability of information about those periods when an attendee or resource is available to be scheduled and when that attendee or resource is not available to be scheduled. This information is referred to in calendaring and scheduling as "Freebusy" time, and is most commonly expressed as an **[RFC2445]** VFREEBUSY object, which can be a request for Freebusy time, a response to a request, or a published set of busy time.

Calendaring standards, such as **[RFC2445]**, **[RFC2446]**, **[RFC2447]**, **[RFC4791]**, and the draft CalDAV Scheduling (<http://tools.ietf.org/html/draft-desruisseaux-caldav-sched.txt>) provide mechanisms for publishing, retrieving, and interrogating Freebusy status.

Outside these specifications, another mechanism for sharing Freebusy information is widely used - Freebusy URL. Freebusy URL typically allows for the publication and/or retrieval of Freebusy information via HTTP. The exact form of the URL, the specifics of the data actually returned (such as the period, format, etc), and error handling are all proprietary to each product or server. In most instances, a URL is stored, most often in an addressbook of some kind. This does not accommodate requests for Freebusy information across specific periods of time.

Freebusy URL is a weak form of calendar sharing which does not provide for interoperability across the range of calendaring products and services, but it is a very useful form of sharing, nonetheless:

- It is simple (in many respects)
- It is relatively easy to implement Freebusy URL
- It facilitates scheduling in the non-enterprise, standalone calendaring client scenario.
- Microsoft Outlook supports a form of Freebusy URL in the "Internet Free/Busy" feature

The objective of this specification is to preserve the simplicity, ease of implementation, and compatibility with Outlook, while providing additional benefits and functionality:

- Standardize and normalize parameters, using a template approach.
- Provide greater flexibility and functionality through the use of optional parameters
- Standardize error handling
- Allow for strong, weak, and no authentication
- Standardize formats for returned Freebusy data

Freebusy URL potentially bridges the divide between enterprise calendaring and calendar/scheduling augmentation/aggregation sites, and standalone calendaring clients (no server).

This document deals only with the Freebusy Read URL, used to retrieve Freebusy information from servers. Other documents will describe publishing of Freebusy information and discovery issues.

---

## 2. Conventions Used in This Document

**TOC**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **[RFC2119]**.

---

## 2.1. Glossary of terms

TOC

Refer to [\[CalConnectGlossary\]](#) for the terms used in this document.

---

## 3. Freebusy Read URL

TOC

The read URL is used to get Freebusy information for a single user. (Note this document does not define a read URL to get the Freebusy information for a group.) A client or user could use this URL by putting it into an FBURL property of a VCARD as defined in Section 2.3.1 of [\[RFC2739\]](#), or it could be published in a calFBURL attribute of an LDAP directory record for a user as defined in Section 2.4.4.2 of [\[RFC2739\]](#)

This document does not address how the URL is obtained, either by the owner of the Freebusy information or by the user of that information.

It is assumed that the owner has some means of obtaining a URL from the calendaring client and that they can transfer it to interested users through some means, such as email.

When the content of a Freebusy URL is accessed, Freebusy information can be returned as VFREEBUSY iCalendar components, as defined by [\[RFC2445\]](#). Such a VFREEBUSY iCalendar component is not meant to conform to the requirements of VFREEBUSY components in [\[RFC2446\]](#). The VFREEBUSY component SHOULD conform to section "4.6.4 Free/Busy Component" of [\[RFC2445\]](#). The VFREEBUSY component SHOULD NOT have a METHOD property, a client MUST ignore the METHOD property, if one is present. The VFREEBUSY component MAY follow the publish setting in regards to the ORGANIZER field. A client SHOULD ignore the ORGANIZER field. Since a Freebusy URL can only refer to a single user, a client will already know how to match the requested

VFREEBUSY component to a user. A server MUST only return a single VFREEBUSY component.

---

## 4. URL Query Parameters

**TOC**

These are the recognized parameters for the Freebusy read URL which relate to the desired range and format of the returned data.

None of these parameters are required except for the conditions noted below. Appropriate defaults will be supplied by the server.

Implementation Notes: Date-only start and end times, i.e. a date/time value without a time component (e.g. 20080101), are not supported because time zone support would be required.

---

### 4.1. start

**TOC**

Default:

The default value is left up to the server. It may be the current day, start of the current month, etc.

Description:

Specifies the start date for the Freebusy data. The server is free to ignore this value and return data in any time range. The client must check the data for the returned time range.

Format:

A profile of an **[RFC3339]** Date/Time. Fractional time is not supported. The server MUST support the expanded version e.g. 2007-01-02T13:00:00-08:00

It is up to the server to interpret local date/times.

Example:

2007-02-03T15:30:00-0800

2007-12-01T10:15:00Z

Notes:

Specifying only a start date/time without specifying an end-date/time or period should be interpreted as in **[RFC2445]**. The effective period should cover the remainder of that day.

Date-only values are disallowed as the server cannot determine the correct start of the day. Only UTC or date/time with offset values are permitted.

---

## 4.2. end

**TOC**

Default:

Same as start

Description:

Specifies the end date for the Freebusy data. The server is free to ignore this value.

Format:

Same as start

Example:

Same as start

---

## 4.3. period

**TOC**

Default:

The default value is left up to the server. The recommended value is "P42D".

Description:

Specifies the amount of Freebusy data to return. A client cannot specify both a period

and an end date. Period is relative to the start parameter.

Format:

An RFC 2445 Duration as defined in section 4.3.6 of **[RFC2445]**

Example:

P42D

---

#### 4.4. format

**TOC**

Default:

"text/calendar"

Description:

Specifies the output format as a MIME type. A server **MUST** support the default "text/calendar" which will return a VFREEBUSY object. Support for other formats is optional.

Format:

A MIME type

Example:

text/calendar

text/html (This format is intended to be viewed by a user in a browser)

Notes:

We anticipate future support for XML and JSON formats.

---

#### 4.5. URL Parameters: Notes

**TOC**

The server is free to ignore the start, end and period parameters. It is recommended that the server return at least 6 weeks of data from the current day.

A client **MUST** check the time range in the VFREEBUSY response as a server may return a different time range than the requested range.

---

## 5. URL Identity Parameters

**TOC**

These are the recognized parameters for the Freebusy read URL which deal with the identity of the requestor and the user for whom freebusy data is being requested and the security of the transaction. These parameters may be encoded in a URL or as query parameters as defined in **Section 8**

Some services may require the presence of a token and/or the user parameter as a form of low-level security.

---

### 5.1. token

**TOC**

Default:

None

Description:

An opaque token that will be recognized by the server. It is generally used for authentication and access control.

Format:

A URL encoded string.

Example:

XfHG65hsjF43

---

### 5.2. user

**TOC**

Default:

None

Description:

A token representing the target user of this Freebusy request. This token may also be referred to as a "userid". It is expected that

some services will use a token which appears to be an email address.

Format:

A UTF-8 URL encoded string.

Example:

user1@example.com

user%201@example.com

jdhdyerhdk (a userid may be an opaque token.)

---

## 6. HTTP Operations

**TOC**

The server SHOULD return an Etag response header for a successful GET request targeting a Freebusy read URL. Clients MAY use the Etag response header value to do subsequent "conditional" GET requests that will avoid re-sending the Freebusy data again if it has not changed.

The read URL is only valid when used with an **HTTP GET** [RFC2616].

---

### 6.1. Response Codes

**TOC**

Below are the typical status codes returned by a GET request targeting a Freebusy URL. Note that other HTTP status codes not listed here might also be returned by a server.

Status Code	Message
-------------	---------

200 OK

302 Redirect

400 Start parameter could not be understood / End

parameter could not be understood / Period parameter could not be understood

401 Unauthorized

403 Forbidden

404 The data for the requested userid is not currently available, but may be available later.

406 The requested format in the Format parameter is not supported.

410 The data for the requested userid is no longer available

500 General server error

---

## 7. Examples

**TOC**

The following are examples of URLs used to retrieve Freebusy data for a user:

```
http://www.example.com/freebusy/user1@example.com?
    start=2007-09-01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00
```

```
http://www.example.com/freebusy/user1@example.com?
    start=2007-09-01T00:00:00-08:00&
    end=2007-09-31T00:00:00-08:00&
    format=text/calendar
```

```
http://www.example.com/calendar/user1@example.com?
    start=2007-09-01T00:00:00-08:00&
    end=2007-09-31T00:00:00-08:00&
    format=text/javascript
```

```
http://www.example.com/freebusy/user1@example.com
```

```
http://www.example.com/freebusy?user=user1@example.com&
    amp;token=xcsfdgetdh&
    start=2008-01-01T00:00:00Z&end=2008-12-
31T00:00:00Z

http://www.example.com/freebusy?user=user%201@example.com&
    start=2008-01-01T00:00:00Z&end=2008-12-
31T00:00:00Z
```

The following are examples of URLs used to retrieve Freebusy data for a user where the userid has been obfuscated:

```
http://www.example.com/freebusy/cbGf65Rfh?
    start=2007-09-01T00:00:00-08:00&
    end=2007-09-31T00:00:00-08:00&
    format=text/calendar

http://www.example.com/calendar/freebusy/cbGf65Rfh
```

Some Request/Response Examples:

An URL with no query parameters:

>> Request <<

```
GET /freebusy/bernard/ HTTP/1.1
Host: www.example.com
```

>> Response <<

```
HTTP/1.1 200 OK
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VFREEBUSY
ORGANIZER;CN="Bernard
Desruisseaux":mailto:bernard@example.com
```

```
UID:76ef34-54a3d2@example.com
DTSTAMP:20050530T123421Z
DTSTART:20060101T000000Z
DTEND:20060108T000000Z
FREEBUSY:20050531T230000Z/20050601T010000Z
FREEBUSY;FBTYPE=BUSY-
TENTATIVE:20060102T100000Z/20060102T120000Z
FREEBUSY:20060103T100000Z/20060103T120000Z
FREEBUSY:20060104T100000Z/20060104T120000Z
FREEBUSY;FBTYPE=BUSY-
UNAVAILABLE:20060105T100000Z/20060105T120000Z
FREEBUSY:20060106T100000Z/20060106T120000Z
END:VFREEBUSY
END:VCALENDAR
```

An URL with start and end parameters:

>> Request <<

```
GET /freebusy/user1@example.com?start=2007-09-
01T00:00:00-08:00&end=2007-09-31T00:00:00-08:00
HTTP/1.1
Host: www.example.com
```

>> Response <<

```
HTTP/1.1 200 OK
Content-Type: text/calendar; charset="utf-8"
Content-Length: xxxx

BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Example Corp.//CalDAV Client//EN
BEGIN:VFREEBUSY
ORGANIZER:
UID:76ef34-54a3d3@example.com
DTSTAMP:20070905T100000Z
DTSTART:20070901T080000Z
DTEND:20070931T080000Z
FREEBUSY:20070903T230000Z/20070904T010000Z
FREEBUSY;FBTYPE=BUSY-
TENTATIVE:20070906T100000Z/20070906T120000Z
FREEBUSY:20070908T100000Z/20070908T120000Z
FREEBUSY:20070909T100000Z/20070909T120000Z
FREEBUSY;FBTYPE=BUSY-
UNAVAILABLE:20070915T100000Z/20070917T120000Z
FREEBUSY:20070920T100000Z/20070922T120000Z
```

```
END:VFREEBUSY
END:VCALENDAR
```

An URL for which the server does not have any data for that user:

>> Request <<

```
GET /freebusy/user1@example.com?start=2012-12-01T00:00:00-08:00&end=2012-12-31T00:00:00-08:00
HTTP/1.1
Host: www.example.com
```

>> Response <<

```
HTTP/1.1 404 No Data
```

---

## 8. Template URLs

**TOC**

In some cases, it may be useful for a client to understand the structure of the read URLs. One such example is a CalDAV server that publishes Freebusy data for an entire group of users.

Template URLs are defined in a draft specification available at [URI-Templates](#). A client can request a read template URL that the client can then use to create normal, i.e. non template, URLs for different users on the same Freebusy server.

The following table defines those keys clients must understand to successfully build a URL from a template.

Template Key	Meaning
userid	The full userid such as "user1@example.com"
user	A userid without a domain, such as "user1"
domain	The domain part of a userid, such as "example.com"

token An auth token that identifies the requesting user.

---

## 9. IANA Considerations

**TOC**

This document does not require any actions on the part of IANA.

---

## 10. Security Considerations

**TOC**

For servers using the token approach for authentication, the token should be considered private. A client should take steps to safeguard the token. The token approach isn't perfect security; a token could become more widely distributed than intended or anticipated.

Servers SHOULD provide a means for token values to be automatically or manually expired.

Services SHOULD consider obfuscating the userid in a Freebusy URL. A service SHOULD prevent a client from probing for valid userids which might reveal private information about users such as their email address. A server MAY return an HTTP 404 error rather than an HTTP 403 error when the requester does not have permission to view the Freebusy information of the requested user.

Servers MAY support HTTP Authentication **[RFC2616]** for access to the Freebusy URL content.

HTTP protocol transactions are sent in the clear over the network unless protection from snooping is negotiated. This can be accomplished by use of TLS, as defined in **[RFC2818]**. In particular, HTTP Basic authentication, as defined in **[RFC2616]**, MUST NOT be used unless TLS is in effect.

Servers MUST take adequate precautions to ensure that clients cannot consume excessive server resources (CPU, memory, disk, etc.) through intentionally malicious requests. For example, a request may be made for an inappropriate amount of time, e.g. 100 years. A server is free to fail such a request.

When rolling up Freebusy information, more information than necessary about a user's events is exposed if busy periods overlap or are adjacent (this tells the client requesting the Freebusy information that the calendar owner has at least two events, rather than knowing only that the calendar owner has one or more events during the busy period).

Thus, a conservative approach to calendar data privacy would have servers always coalesce such busy periods when they are the same type. Security considerations described in iCalendar [\[RFC2445\]](#) and iTIP [\[RFC2446\]](#) are also applicable.

---

## 11. Acknowledgements

**TOC**

The author would like to thank the members of the Calendaring and Scheduling Consortium Freebusy technical committee and the following individuals for contributions: Gary Schwartz, Michael Douglass, Cyrus Daboo.

---

## 12. References

**TOC**

---

**TOC**

## 12.1. Normative References

- [RFC2119] [Bradner, S.](#), "[Key words for use in RFCs to Indicate Requirement Levels](#)," BCP 14, RFC 2119, March 1997 ([TXT](#), [HTML](#), [XML](#)).
- [RFC2445] [Dawson, F.](#) and [Stenerson, D.](#), "[Internet Calendaring and Scheduling Core Object Specification \(iCalendar\)](#)," RFC 2445, November 1998 ([TXT](#), [HTML](#), [XML](#)).
- [RFC2446] [Silverberg, S.](#), [Mansour, S.](#), [Dawson, F.](#), and [R. Hopson](#), "[iCalendar Transport-Independent Interoperability Protocol \(iTIP\) Scheduling Events, BusyTime, To-dos and Journal Entries](#)," RFC 2446, November 1998 ([TXT](#), [HTML](#), [XML](#)).
- [RFC2447] [Dawson, F.](#), [Mansour, S.](#), and [S. Silverberg](#), "[iCalendar Message-Based Interoperability Protocol \(iMIP\)](#)," RFC 2447, November 1998 ([TXT](#), [HTML](#), [XML](#)).
- [RFC2616] [Fielding, R.](#), [Gettys, J.](#), [Mogul, J.](#), [Frystyk, H.](#), [Masinter, L.](#), [Leach, P.](#), and [T. Berners-Lee](#), "[Hypertext Transfer Protocol -- HTTP/1.1](#)," RFC 2616, June 1999 ([TXT](#), [PS](#), [PDF](#), [HTML](#), [XML](#)).
- [RFC2818] Rescorla, E., "[HTTP Over TLS](#)," RFC 2818, May 2000 ([TXT](#)).
- [RFC3339] [Klyne, G., Ed.](#) and [C. Newman](#), "[Date and Time on the Internet: Timestamps](#)," RFC 3339, July 2002 ([TXT](#), [HTML](#), [XML](#)).
- 

## 12.2. Informative References

**TOC**

- [CalConnectGlossary] CalConnect.org, "[CalConnect Calendar Glossary](#)," STD 1, October 2006 ([PDF](#)).
- [I-D.ietf-vcarddav-vcardrev] Perreault, S. and P. Resnick, "[vCard Format Specification](#)," draft-ietf-vcarddav-vcardrev-06 (work in progress), March 2009 ([TXT](#)).
- [RFC2739] Small, T., Hennessy, D., and F. Dawson, "[Calendar Attributes for vCard and LDAP](#)," RFC 2739, January 2000 ([TXT](#)).
- [RFC4791] [Daboo, C.](#), [Desruisseaux, B.](#), and [L. Dusseault](#), "[Calendaring Extensions to WebDAV \(CalDAV\)](#)," RFC 4791, March 2007 ([TXT](#), [HTML](#), [XML](#)).
- 

## Author's Address

**TOC**

Eric York  
Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
USA

Email: [eyork@apple.com](mailto:eyork@apple.com)  
URI: <http://www.apple.com/>