

Network Working Group

INTERNET-DRAFT

Calendaring and Scheduling Working Group

<ietf-draft-calsch-itip-03.txt

Expires in six months from

Steve Silverberg, Microsoft

Steve Mansour, Netscape

Frank Dawson, Lotus

Ross Hopson, ON Technologies

March 13, 1998

**iCalendar Transport-Independent Interoperability Protocol
(iTIP)
Scheduling Events, BusyTime, To-dos and Journal Entries**

Status of this Memo

This document is an Internet-Draft. 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. Internet-Drafts may be updated, replaced, or made obsolete by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as a "working draft" or "work in progress".

To learn the current status of any Internet-Draft, please check the `ltd-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on `ds.internic.net` (US East Coast), `nic.nordu.net` (Europe), `ftp.isi.edu` (US West Coast), or `munniari.oz.au` (Pacific Rim).

Distribution of this document is unlimited.

Copyright (C) The Internet Society 1997. All Rights Reserved.

Abstract

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

Subsequent documents specify interoperable methods of communications between systems that use this protocol.

The document outlines a model for calendar exchange that defines both static and dynamic event, to-do, journal and free/busy objects. Static objects are used to transmit information from one entity to another without the expectation of continuity or referential integrity with the original item. Dynamic objects are a superset of static objects and will gracefully degrade to their static counterparts for clients that only support static objects.

Silverberg/Mansour/Dawson/Hopson 1

Expires September 1998

Internet Draft

iTIP

March 13, 1998

Silverberg/Mansour/Dawson/Hopson 2

Expires September 1998

Internet Draft

iTIP

March 13, 1998

Silverberg/Mansour/Dawson/Hopson 3

Expires September 1998

Internet Draft

iTIP

March 13, 1998

Table of Contents

1	INTRODUCTION.....	9
1.1	FORMATTING CONVENTIONS	9
1.2	RELATED DOCUMENTS	10
1.3	ITIP ROLES AND TRANSACTIONS	10
2	INTEROPERABILITY MODELS.....	11
2.1	APPLICATION PROTOCOL	12
2.1.1	Calendar Entry State	12
2.1.2	Delegation	13
2.1.3	Acting on Behalf of other Calendar Users	13
3	APPLICATION PROTOCOL ELEMENTS.....	13
3.1	COMMON COMPONENT RESTRICTION TABLES	14
3.2	METHODS FOR VEVENT CALENDAR COMPONENTS	15
3.2.1	PUBLISH	16
3.2.2	REQUEST	17
3.2.2.1	Rescheduling an Event.....	18
3.2.2.2	Updating or Reconfirmation of an Event.....	19
3.2.2.3	Delegating an Event to another CU.....	19
3.2.2.4	Changing the Organizer.....	20

3.2.2.5	Sending on Behalf of the Organizer.....	20
3.2.2.6	Forwarding to An Uninvited CU.....	20
3.2.2.7	Updating Attendee Status.....	20
3.2.3	REPLY	20
3.2.4	ADD	22
3.2.5	CANCEL	23
3.2.6	REFRESH	25
3.2.7	COUNTER	26
3.2.8	DECLINECOUNTER	28
3.3	METHODS FOR VFREEBUSY COMPONENTS	29
3.3.1	PUBLISH	30
3.3.2	REQUEST	31
3.3.3	REPLY	32
3.4	METHODS FOR VTODO COMPONENTS	34
3.4.1	PUBLISH	35
3.4.2	REQUEST	36
3.4.2.1	REQUEST for Rescheduling a VTODO.....	37
3.4.2.2	REQUEST for Update or Reconfirmation of a VTODO.....	38
3.4.2.3	REQUEST for Delegating a VTODO.....	38
3.4.2.4	REQUEST Forwarded To An Uninvited Calendar User.....	39
3.4.2.5	REQUEST Updated Attendee Status.....	39
3.4.3	REPLY	39
3.4.4	ADD	41
3.4.5	CANCEL	42
3.4.6	REFRESH	43
3.4.7	COUNTER	45
3.4.8	DECLINECOUNTER	46
3.5	METHODS FOR VJOURNAL COMPONENTS	47

Silverberg/Mansour/Dawson/Hopson 4

Expires September 1998

Internet Draft

iTIP

March 13, 1998

3.5.1	PUBLISH	47
3.5.2	ADD	49
3.5.3	CANCEL	50
3.6	STATUS REPLIES	51
3.7	IMPLEMENTATION CONSIDERATIONS	53
3.7.1	Working With Recurrence Instances	53
3.7.2	Attendee Property Considerations	54
3.7.3	X-Tokens	55
4	EXAMPLES.....	55
4.1	PUBLISHED EVENT EXAMPLES	55
4.1.1	A Minimal Published Event	56
4.1.2	Changing A Published Event	56
4.1.3	Canceling A Published Event	57

4.1.4	A Rich Published Event	57
4.1.5	Anniversaries or Events attached to entire days	58
4.2	GROUP EVENT EXAMPLES	59
4.2.1	A Group Event Request	60
4.2.2	Reply To A Group Event Request	60
4.2.3	Update An Event	61
4.2.4	Countering an Event Proposal	61
4.2.5	Delegating an Event	63
4.2.6	Delegate Accepts the Meeting	65
4.2.7	Delegate Declines the Meeting	66
4.2.8	Forwarding an Event Request	67
4.2.9	Cancel A Group Event	67
4.2.10	Removing Attendees	68
4.2.11	Replacing the Organizer	69
4.3	BUSY TIME EXAMPLES	70
4.3.1	Request Busy Time	70
4.3.2	Reply To A Busy Time Request	71
4.4	RECURRING EVENT AND TIME ZONE EXAMPLES	71
4.4.1	A Recurring Event Spanning Time Zones	71
4.4.2	Modify A Recurring Instance	73
4.4.3	Cancel an Instance	74
4.4.4	Cancel Recurring Event	75
4.4.5	Change All Future Instances	75
4.4.6	Add A New Instance To A Recurring Event	75
4.4.7	Add A New Series of Instances To A Recurring Event	76
4.4.8	Counter An Instance Of A Recurring Event	80
4.4.9	Error Reply To A Request	80
4.5	GROUP TO-DO EXAMPLES	81
4.5.1	A VTODD Request	82
4.5.2	A VTODD Reply	83
4.5.3	A VTODD Request for Updated Status	83
4.5.4	A Reply: Percent-Complete	84
4.5.5	A Reply: Completed	84
4.5.6	An Updated VTODD Request	84
4.5.7	Recurring VTODDs	85
4.5.7.1	Request for a Recurring VTODD.....	85
4.5.7.2	Calculating due dates in recurring VTODDs.....	85
4.5.7.3	Replying to an instance of a recurring VTODD.....	86
4.6	JOURNAL EXAMPLES	86

Silverberg/Mansour/Dawson/Hopson 5

Expires September 1998

Internet Draft

iTIP

March 13, 1998

4.7	OTHER EXAMPLES	87
4.7.1	Event Refresh	87
4.7.2	Bad RECURRENCE-ID	87

5	APPLICATION PROTOCOL FALLBACKS.....	88
5.1	PARTIAL IMPLEMENTATION	88
5.1.1	Event-Related Fallbacks	89
5.1.2	To-Do-Related Fallbacks	90
5.1.3	Journal-Related Fallbacks	92
5.2	LATENCY ISSUES	93
5.2.1	Cancellation of an Unknown Calendar Component.	93
5.2.2	Unexpected Reply from an Unknown Delegate	93
5.3	SEQUENCE NUMBER	94
6	SECURITY CONSIDERATIONS.....	94
6.1	SECURITY THREATS	94
6.1.1	Spoofing the "Organizer"	94
6.1.2	Spoofing the "Attendee"	94
6.1.3	Unauthorized Replacement of the Organizer	95
6.1.4	Eavesdropping	95
6.1.5	Flooding a Calendar	95
6.1.6	Procedural Alarms	95
6.1.7	Unauthorized REFRESH Requests	95
6.2	RECOMMENDATIONS	95
6.2.1	Use of [RFC1847] to secure iTIP transactions	96
6.2.2	Implementation Controls	96
7	ACKNOWLEDGMENTS.....	96
8	BIBLIOGRAPHY.....	97
9	AUTHORS ADDRESSES.....	98
10	FULL COPYRIGHT STATEMENT.....	99

1 Introduction

This document specifies how calendaring systems use iCalendar objects to interoperate with other calendar systems. In particular, it specifies how to schedule events, to-dos, or daily journal entries. It further specifies how to search for available busy time information. It does so in a general way so as to allow multiple methods of communication between systems. Subsequent documents specify transport bindings between systems that use this protocol.

This protocol is based on messages sent from an originator to one or more recipients. For certain types of messages, a recipient may reply, in order to update their status and may also return transaction/request status information. The protocol supports the ability for the message originator to modify or cancel the original message. The protocol also supports the ability for recipients to suggest changes to the originator of a message. The elements of the protocol also define the user roles for its transactions.

1.1 Formatting Conventions

In order to refer to elements of the calendaring and scheduling model, core object or interoperability protocol defined in [[ICMS](#)], [[ICAL](#)] and [ITIP] several formatting conventions have been utilized.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [[RFC 2119](#)].

Calendaring and scheduling roles described by [[ICMS](#)] are referred to in quoted-strings of text with the first character of each word in upper case. For example, "Organizer" refers to a role of a "Calendar User" (CU) within the scheduling protocol defined by [ITIP]. Calendar components defined by [[ICAL](#)] are referred to with capitalized, quoted-strings of text. All calendar components start with the letter "V". For example, "VEVENT" refers to the event calendar component, "VTODO" refers to the to-do calendar component and "VJOURNAL" refers to the daily journal calendar component. Scheduling methods defined by [ITIP] are referred to with capitalized, quoted-strings of text. For example, "REQUEST" refers to the method for requesting a scheduling calendar component be created or modified, "REPLY" refers to the method a recipient of a request uses to update their status with the "Organizer" of the calendar component.

Properties defined by [[ICAL](#)] are referred to with capitalized, quoted-strings of text, followed by the word "property". For example, "ATTENDEE" property refers to the iCalendar property used to convey the calendar address of a "Calendar User". Property parameters defined by this memo are referred to with lower case, quoted-strings of text, followed by the word "parameter". For example, "value" parameter refers to the iCalendar property parameter used to override the default data

Silverberg/Mansour/Dawson/Hopson 7

Expires September 1998

Internet Draft

iTIP

March 13, 1998

type for a property value. Enumerated values defined by this memo are referred to with capitalized text, either alone or followed by the word "value".

In tables, the quoted-string text is specified without quotes in order to minimize the table length.

[1.2](#) Related Documents

Implementers will need to be familiar with several other memos that, along with this one, describe the Internet calendaring and scheduling standards. This document, [ITIP], specifies an interoperability protocol for scheduling between different implementations. The related documents are:

[ICMS] - describes the abstract model and defines common terms and concepts;

[ICAL] - specifies the objects, data types, properties and property parameters used in the protocols, along with the methods for representing and encoding them;

[IRIP] - specifies an Internet real time protocol binding for [ITIP];

[IMIP] specifies an Internet email binding for [ITIP].

This memo does not attempt to repeat the specification of concepts or definitions from these other memos. Where possible, references are made to the memo that provides for the specification of these concepts or definitions.

[1.3](#) ITIP Roles and Transactions

ITIP defines seven methods for exchanging [[ICAL](#)] objects for the

purposes of group calendaring and scheduling between "Calendar Users" (CUs). CUs take on one of two roles in iTIP. The CU who initiates an exchange takes on the role of "Organizer". For example, the CU who proposes a group meeting is the "Organizer". The CUs asked to participate in the group meeting by the "Organizer" take on the role of "Attendee". Note that "role" is also a descriptive parameter to the _ATTENDEE_ property. Its use is to convey descriptive context to an "Attendee" such as "chair", "req-participant" or "non-participant" and has nothing to do with the calendaring workflow.

The iTIP methods are defined below and their usage and semantics are outlined in [section 3](#) of this document.

Method	Description
PUBLISH	Used to publish a calendar entry to one or more

Silverberg/Mansour/Dawson/Hopson 8

Expires September 1998

Internet Draft

iTIP

March 13, 1998

	Calendar Users. There is no interactivity between the publisher and any other calendar user. An example might include a baseball team publishing its schedule to the public.
REQUEST	Used to schedule a calendar entry with other Calendar Users. Requests are interactive in that they require the receiver to respond using the the Reply methods. Meeting Requests, Busy Time requests and the assignment of VTODOs to other Calendar Users are all examples. Requests are also used by the "Organizer" to update the status of a calendar entry.
REPLY	A Reply is used in response to a Request to convey "Attendee" status to the "Organizer". Replies are commonly used to respond to meeting and task requests.
ADD	Add one or more instances to an existing VEVENT, VTOD0, or VJOURNAL.
CANCEL	Cancel one or more instances of an existing VEVENT, VTOD0, or VJOURNAL.
REFRESH	The Refresh method is used by an "Attendee" to

	request the latest version of a calendar entry.
COUNTER	The Counter method is used by an "Attendee" to negotiate a change in the calendar entry. Examples include the request to change a proposed Event time or change the due date for a VTODDO.
DECLINE-COUNTER	Used by the "Organizer" to decline the proposed counter-proposal.

2 Interoperability Models

There are two distinct protocols relevant to interoperability: an "Application Protocol" and a "Transport Protocol". The Application Protocol defines the content of the iCalendar objects sent between sender and receiver to accomplish the scheduling transactions described above. The Transport Protocol defines how the iCalendar objects are sent between the sender and receiver. This document focuses on the Application Protocol. Binding documents such as [\[IMIP\]](#) and [\[IRIP\]](#) focus on the Transport Protocol.

The connection between Sender and Receiver in the diagram below refers to the Application Protocol. The iCalendar objects passed from the Sender to the Receiver are presented in [Section 3](#), Application Protocol Elements.

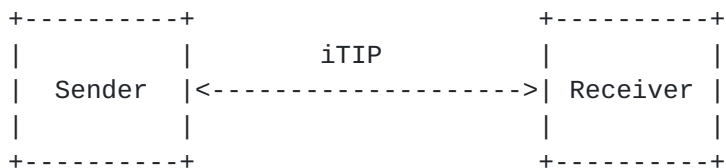
Silverberg/Mansour/Dawson/Hopson 9

Expires September 1998

Internet Draft

iTIP

March 13, 1998



There are several variations of this diagram in which the Sender and Receiver take on various roles of a "Calendar User Agent" (CUA) or a "Calendar Service" (CS). These variants are detailed in the Model document [\[ICMS\]](#).

The architecture of iTIP is depicted in the diagram below. An application written to this specification may work with bindings for the store-and-forward transport, the real time transport, or both. Also note that iTIP could be bound to other transports.

+-----+			
	iTIP		
+-----+			
Real-time	Store-and-Fwd	Other	
Transport	Transport	Transports...	
+-----+			

2.1 Application Protocol

The iTIP model is in which a calendar entry is created and managed by an "Organizer". The "Organizer" interacts with other CUs by sending one or more of the iTIP messages described above. "Attendees" use the "REPLY" method to communicate their status. "Attendees" do not make direct changes to the master calendar entry. They can, however, use the "COUNTER" method to suggest changes to the "Organizer". The "Organizer" has complete control over the master calendar entry.

2.1.1 Calendar Entry State

There are two distinct states relevant to calendar entries: the overall state of the entry and the state associated with an "Attendee" to that entry.

The state of an entry is defined by the "STATUS" property and is controlled by the "Organizer." There is no default value for the "STATUS" property. The "Organizer" sets the "STATUS" property to the appropriate value for each calendar entry.

The state of a particular "Attendee" relative to an entry is defined by the "attstat" parameter in the "ATTENDEE" property for each "Attendee". When an "Organizer" issues the initial entry, "Attendee" status is unknown. The "Organizer" specifies this by setting the "attstat" parameter to "needs-action". Each "Attendee" modifies their "ATTENDEE" property "attstat" parameter to an appropriate value as part of a "REPLY" message sent back to the "Organizer".

Silverberg/Mansour/Dawson/Hopson 10

Expires September 1998

Internet Draft

iTIP

March 13, 1998

2.1.2 Delegation

Delegation is defined as the process by which an "Attendee" grants another CU (or several CUs) the right to attend on their behalf. The "Organizer" is made aware of this change because the delegating "Attendee" informs the "Organizer" of this change. These steps are

detailed in the REQUEST method section.

2.1.3 Acting on Behalf of other Calendar Users

In many organizations one user will act on behalf of another to organize and/or respond to meeting requests. ITIP provides two mechanisms that support these activities.

First, the "Organizer" is treated as a special entity, separate from "Attendees". All responses from "Attendees" flow to the "Organizer", making it easy to separate a calendar user organizing a meeting from calendar users attending the meeting. Additionally, iCalendar provides descriptive roles for each "Attendee". For instance, a role of "chair" may be ascribed to one or more "Attendees". The "chair and the "Organizer" may or may not be the same calendar user. This maps well to scenarios where an assistant may manage meeting logistics for another individual who chairs a one-time or recurring meeting.

Second, a "sent-by" parameter may be specified in either the "Organizer" or "Attendee" properties. When specified, the "sent-by" parameter indicates that the responding CU acted on behalf of the specified "Attendee" or "Organizer".

3 Application Protocol Elements

ITIP messages are "text/calendar" MIME entities that contain calendaring and scheduling information. The particular type of [[ICAL](#)] message is referred to as the "method type". Each method type is identified by a "METHOD" property specified as part of the "text/calendar" content type. The table below shows various combinations of calendar components and the method types that this memo supports.

+=====+					
	VEVENT	VTOD	VJOURNAL	VFREEBUSY	
+=====+					
Publish	Yes	Yes	Yes	Yes	
Request	Yes	Yes	No	Yes	
Refresh	Yes	Yes	No	No	
Cancel	Yes	Yes	Yes	No	
Add	Yes	Yes	Yes	No	
Reply	Yes	Yes	No	Yes	
Counter	Yes	Yes	No	No	
Decline-					
Counter	Yes	Yes	No	No	
+=====+					

The property restrictions in the table below apply to any "VALARM"

component in an iTIP message.

Component/Property	Presence
--------------------	----------

VALARM	MAY
ALARM-TYPE	MUST
DURATION	MUST

Silverberg/Mansour/Dawson/Hopson 12

Expires September 1998

Internet Draft

iTIP

March 13, 1998

REPEAT	MUST
TRIGGER	MUST
ATTACH	MAY
DESCRIPTION	MAY
SUMMARY	MAY
COMMENT	NOT
CREATED	NOT
LAST-MODIFIED	NOT
RELATED-TO	NOT
URL	NOT

[3.2](#) Methods for VEVENT Calendar Components

This section defines the property set restrictions for the method types that are applicable to the "VEVENT" calendar component. Each method is defined using a table that clarifies the property constraints that define the particular method.

The following summarizes the methods that are defined for the "VEVENT" calendar component.

Method	Description
PUBLISH	Post notification of an event. Used primarily as a method of advertising the existence of an event.
REQUEST	Make a request for an event. This is an explicit invitation to one or more "Attendees". Event Requests are also used to update or change an existing event. Clients that cannot handle REQUEST may degrade the event to view it as an PUBLISH.

REPLY	Reply to an event request. Clients may set their	
	status ("attstat") to ACCEPTED, DECLINED,	
	TENTATIVE, or DELEGATED.	
ADD	Add one or more instances to an existing event.	
CANCEL	Cancel one or more instances of an existing	
	event.	
REFRESH	A request is sent to an "Organizer" by an	
	"Attendee" asking for the latest version of an	
	event to be resent to the requester.	
COUNTER	Counter a REQUEST with an alternative proposal,	
	Sent by an "Attendee" to the "Organizer".	
DECLINECOUNTER	Decline a counter proposal. Sent to an	

Silverberg/Mansour/Dawson/Hopson 13

Expires September 1998

Internet Draft

iTIP

March 13, 1998

	"Attendee" by the "Organizer".	
+=====+		

3.2.1 PUBLISH

The "PUBLISH" method in a "VEVENT" calendar component is an unsolicited posting of an iCalendar object. Any CU may add published components to their calendar. The "Organizer" MUST be present in a published iCalendar component. Other "Attendees" MAY be present. However, no replies are expected. Its expected usage is for encapsulating an arbitrary event as an iCalendar object. The "Organizer" may subsequently update (with another "PUBLISH" method), add instances to (with an "ADD" method), or cancel (with a "CANCEL" method) a previously published "VEVENT" calendar component.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property Presence

-----	-----
PROID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "PUBLISH"
VEVENT	MUST
DTSTAMP	MUST
DTSTART	MUST

ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it must NOT be present.
SEQUENCE	MUST be present if not 0, may be present if 0
SUMMARY	MUST; may be null.
UID	MUST
ATTENDEE	MAY
ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY be present and MAY be NULL
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY
PRIORITY	MAY
RELATED-TO	MAY
RDATE	MAY
RESOURCES	MAY
RRULE	MAY

Silverberg/Mansour/Dawson/Hopson 14

Expires September 1998

Internet Draft

iTIP

March 13, 1998

STATUS	MAY be one of TENTATIVE/CONFIRMED/CANCELLED
TRANSP	MAY
URL	MAY
DURATION	NOT
VTODO	NOT
VJOURNAL	NOT
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
VFREEBUSY	NOT
X-TOKENS	MAY

[3.2.2](#) REQUEST

The "REQUEST" method in a "VEVENT" component provides the following scheduling functions:

- . Invite "Attendees" to an event;
- . Reschedule an existing event;
- . Update the details of an existing event, without rescheduling it;
- . Update the status of "Attendees" of an existing event, without rescheduling it;
- . Reconfirm an existing event, without rescheduling it;
- . To forward a VEVENT to another uninvited CU.
- . For an existing "VEVENT" calendar component, delegate the role of "Attendee" to another CU;
- . For an existing "VEVENT" calendar component, changing the role of "Organizer" to another CU.

The "Organizer" originates the "REQUEST". The recipients of the "REQUEST" method are the CUs invited to the event, the "Attendees". "Attendees" use the "REPLY" method to convey attendance status to the "Organizer".

The "UID" and "SEQUENCE" properties are used to distinguish the various uses of the "REQUEST" method. If the "UID" property value in the "REQUEST" is not found on the recipient's calendar, then the "REQUEST" is for a new "VEVENT" calendar component. If the "UID" property value is found on the recipient's calendar, then the "REQUEST" is for a rescheduling, an update, or a reconfirm of the "VEVENT" calendar component.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "REQUEST"

Silverberg/Mansour/Dawson/Hopson 15 Expires September 1998

Internet Draft iTIP March 13, 1998

VEVENT	MUST
ATTENDEE	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it

must NOT be present.

SEQUENCE	MUST be present if non-zero, may be present if zero
SUMMARY	MUST be present, may be NULL
UID	MUST

ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY be present and may be NULL
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY
PRIORITY	MAY
RDATE	MAY
RELATED-TO	MAY
RESOURCES	MAY
RRULE	MAY
STATUS	MAY be one of TENTATIVE/CONFIRMED
TRANSP	MAY
URL	MAY
DURATION	NOT
REQUEST-STATUS	NOT

VTOD0	NOT
VJOURNAL	NOT
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
VFREEBUSY	NOT
X-TOKENS	MAY

3.2.2.1 Rescheduling an Event

The "REQUEST" method may be used to reschedule an event. A rescheduled event involves a change to the existing event in terms of its time or recurrence intervals and possibly the location or description. If the recipient CUA of a "REQUEST" method finds that the "UID" property value already exists on the calendar, but that the "SEQUENCE" (or "DTSTAMP") property value in the "REQUEST" method is greater than the value for the

existing event, then the "REQUEST" method describes a rescheduling of the event.

3.2.2.2 Updating or Reconfirmation of an Event

The "REQUEST" method may be used to update or reconfirm an event. An update to an existing event does not involve changes to the time or recurrence intervals, and might not involve a change to the location or description for the event. If the recipient CUA of a "REQUEST" method finds that the "UID" property value already exists on the calendar and that the "SEQUENCE" property value in the "REQUEST" is the same as the value for the existing event, then the "REQUEST" method describes an update of the event details, but no rescheduling of the event.

The update "REQUEST" method is the appropriate response to a "REFRESH" method sent from an "Attendee" to the "Organizer" of an event.

The "Organizer" of an event may also send unsolicited "REQUEST" methods. The unsolicited "REQUEST" methods may be used to update the details of the event without rescheduling it, to update the "attstat" parameter of "Attendees", or to reconfirm the event.

3.2.2.3 Delegating an Event to another CU

Some calendar and scheduling systems allow "Attendees" to delegate their presence at an event to another calendar user. ITIP supports this concept using the following workflow. Any "Attendee" may delegate their right to participate in a calendar VEVENT to another CU. The implication is that the delegate participates in lieu of the original "Attendee"; NOT in addition to the "Attendee". The delegator MUST notify the "Organizer" of this action using the steps outlined below. Implementations may support or restrict delegation as they see fit. For instance, some implementations may restrict a delegate from delegating a "REQUEST" to another CU.

The "Delegator" of an event forwards the existing "REQUEST" to the "Delegate". The "REQUEST" method MUST include an "ATTENDEE" property with the calendar address of the "Delegate". The "Delegator" MUST also send a "REPLY" method to the "Organizer" with the "Delegator's" "ATTENDEE" property "attstat" parameter value set to "delegated". In addition, the "delegated-to" parameter MUST be included with the calendar address of the "Delegate".

In response to the request, the "Delegate" MUST send a "REPLY" method to the "Organizer" and optionally, to the "Delegator". The "REPLY" method "SHOULD" include the "ATTENDEE" property with the "delegated-from" parameter value of the "Delegator's" calendar address.

[3.2.2.4](#) Changing the Organizer

The situation may arise where the "Organizer" of a VEVENT is no longer able to perform the "Organizer" role and abdicates without passing on the "Organizer" role to someone else. When this occurs the "Attendees" of the VEVENT may use out-of-band mechanisms to communicate the situation and agree upon a new "Organizer". The new "Organizer" should then send out a new "REQUEST" with a modified version of the VEVENT in which the "SEQUENCE" number has been incremented and value of the "ORGANIZER" property has been changed to the calendar address of the new "Organizer".

[3.2.2.5](#) Sending on Behalf of the Organizer

There are a number of scenarios that support the need for a calendar user to act on behalf of the "Organizer" without explicit role changing. This might be the case if the CU designated as "Organizer" was sick or unable to perform duties associated with that function. In these cases iTIP supports the notion of one CU acting on behalf of another. Using the "sent-by" parameter, a calendar user could send an updated "VEVENT" REQUEST. In the case where one CU sends on behalf of another CU, the "Attendee" responses are still directed back towards the CU designated as "Organizer".

[3.2.2.6](#) Forwarding to An Uninvited CU

An "Attendee" invited to an event may invite another uninvited CU to the event. The invited "Attendee" accomplishes this by forwarding the original "REQUEST" method to the uninvited CU. Whether the uninvited CU is added to the attendee list, and thus informed of changes to the "VEVENT" calendar component, is an issue left to individual implementations. It is not required. It is important to note that when forwarding a "REQUEST" to another CU, the forwarding "Attendee" MUST NOT make changes to the VEVENT property set.

[3.2.2.7](#) Updating Attendee Status

The "Organizer" of an event may also request updated status from one or more "Attendees. The "Organizer" sends a "REQUEST" method to the "Attendee" and sets the "ATTENDEE;RSVP=TRUE" property parameter. The "SEQUENCE" property for the event is not changed from its previous value. A recipient will determine that the only change in the "REQUEST" is that their "RSVP" property parameter indicates a request for updated status. The recipient SHOULD respond with a "REPLY" method indicating their current status with respect to the "REQUEST".

[3.2.3](#) **REPLY**

The "REPLY" method in a "VEVENT" calendar component is used to respond (e.g., accept or decline) to a "REQUEST" or to reply to a delegation

Silverberg/Mansour/Dawson/Hopson 18

Expires September 1998

Internet Draft

iTIP

March 13, 1998

"REQUEST". When used to provide a delegation response, the "Delegator" SHOULD include the calendar address of the "Delegate" on the "delegated-to" property parameter of the "Delegator's" "ATTENDEE" property. The "Delegate" SHOULD include the calendar address of the "Delegator" on the "delegated-from" property parameter of the "Delegate's" "ATTENDEE" property.

The "REPLY" method may also be used to respond to an unsuccessful "REQUEST" method. Depending on the value of the "REQUEST-STATUS" property no scheduling action may have been performed.

The "Organizer" of an event may receive the "REPLY" method from a CU not in the original "REQUEST". For example, a "REPLY" may be received from a "Delegate" to an event. In addition, the "REPLY" method may be received from an unknown CU (a "Party Crasher"). This uninvited "Attendee" may be accepted, or the "Organizer" may cancel the event for the uninvited "Attendee" by sending a "CANCEL" method to the uninvited "Attendee".

The "Organizer" may also receive a "REPLY" from one CU on behalf of another. Like the scenario enumerated above for the "Organizer", "Attendees" may have another CU respond on their behalf. This is done using the "sent-by" parameter.

The optional properties listed in the table below (those listed as "MAY") MUST NOT be changed from those of the original request. If property changes are desired the COUNTER message must be used.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "REPLY"
VEVENT	MUST
ATTENDEE	MUST be the address of the Attendee replying.
DTSTAMP	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it must NOT be present.
SEQUENCE	MUST if non-zero
UID	MUST be the UID of the original REQUEST
ATTACH	MAY
COMMENT	MAY
CONTACT	MAY
EXDATE	MAY
EXRULE	MAY
REQUEST-STATUS	MAY
CATEGORIES	MAY

Silverberg/Mansour/Dawson/Hopson 19

Expires September 1998

Internet Draft

iTIP

March 13, 1998

CLASS	MAY
CREATED	MAY
DESCRIPTION	MAY
DTSTART	MAY
DTEND	MAY
DURATION	MAY
GEO	MAY
LAST-MODIFIED	MAY
PRIORITY	MAY
RELATED-TO	MAY
RESOURCES	MAY
RDATE	MAY
RRULE	MAY
STATUS	MAY
SUMMARY	MAY
TRANSP	MAY
URL	MAY

VTIMEZONE MUST be present if any date/time refers to a
timezone

X-TOKENS	NOT
VTODO	NOT
VJOURNAL	NOT
VFREEBUSY	NOT
VALARM	NOT

3.2.4 ADD

The "ADD" method in a "VEVENT" calendar component is used to add one or more instances to an existing "VEVENT". Unlike the "REQUEST" method, when using issuing an "ADD" method, the "Organizer" does not send the full "VEVENT" description; only the new instance(s). The "ADD" method is especially useful if there are instance-specific properties to be preserved in a recurring "VEVENT". For instance, an "Organizer" may have originally scheduled a weekly Thursday meeting. At some point, several instances changed. Location or start time may have changed, or some instances may have unique "DESCRIPTION" properties. The "ADD" method allows the "Organizer" to add new instances to an existing event using a single ITIP message without redefining the entire recurring pattern.

The "UID" must be that of the existing event. If the "UID" property value in the "ADD" is not found on the recipient's calendar, then the recipient SHOULD send a "REFRESH" to the "Organizer" in order to be updated with the latest version of the "VEVENT". If an "Attendee" implementation does not support the "ADD" method it should respond with a "REQUEST-STATUS" value of 5.3 and ask for a "REFRESH".

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST

Silverberg/Mansour/Dawson/Hopson	20	Expires September 1998
----------------------------------	----	------------------------

Internet Draft	iTIP	March 13, 1998
----------------	------	----------------

VERSION	MUST be "2.0"
METHOD	MUST be "ADD"
VEVENT	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST
SEQUENCE	MUST be greater than 0
SUMMARY	MUST be present, may be NULL
UID	MUST match that of the original event

specified in the "CANCEL" method. In order to cancel an individual instance of the event, the "RECURRENCE-ID" property value for the event MUST be specified in the "CANCEL" method.

There are two options for canceling a sequence of instances of a recurring "VEVENT" calendar component:

- (a) the "RECURRENCE-ID" property for an instance in the sequence MUST be specified with the "RANGE" property parameter value of THISANDPRIOR (or THISANDFUTURE) to indicate cancellation of the specified "VEVENT" calendar component and all instances before (or after); or
- (b) individual recurrence instances may be cancelled by specifying multiple "RECURRENCE-ID" properties corresponding to the instances to be cancelled.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "CANCEL"
VEVENT	MUST
DTSTAMP	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to one or more instances of a recurring calendar component. Otherwise it MUST NOT be present.
SEQUENCE	MUST
UID	MUST be the UID of the original REQUEST
COMMENT	MAY
STATUS	Must be set to "Cancelled". If uninviting specific "Attendees" then MUST NOT be included.
ATTACH	MAY
ATTENDEE	MAY
CATEGORIES	MAY
CLASS	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY
DTSTART	MAY
DURATION	MAY
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY

LAST-MODIFIED	MAY
PRIORITY	MAY
RELATED-TO	MAY

Silverberg/Mansour/Dawson/Hopson 22

Expires September 1998

Internet Draft

iTIP

March 13, 1998

REQUEST-STATUS	MAY
RESOURCES	MAY
RDATE	MAY
RRULE	MAY
STATUS	MAY
SUMMARY	MAY
TRANSP	MAY
URL	MAY

VTIMEZONE	MUST be present if any date/time refers to a timezone
-----------	--

X-TOKENS	MAY
VTODO	NOT
VJOURNAL	NOT
VFREEBUSY	NOT
VALARM	NOT

3.2.6 REFRESH

The "REFRESH" method in a "VEVENT" calendar component is used by "Attendees" of an existing event to request an updated description from the event "Organizer". The "REFRESH" method must specify the "UID" property of the event to update. A recurrence instance of an event may be requested by specifying the "RECURRENCE-ID" property corresponding to the associated event. The "Organizer" responds with the latest description and version of the event.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence

PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "REFRESH"
VEVENT	MUST
ATTENDEE	MUST be the address of requestor
DTSTAMP	MUST

RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it must NOT be present.
UID	MUST be the UID associated with original REQUEST
COMMENT	MAY
ATTACH	NOT
ATEGORIES	NOT
CLASS	NOT
CONTACT	MAY
CREATED	NOT

Silverberg/Mansour/Dawson/Hopson 23

Expires September 1998

Internet Draft

iTIP

March 13, 1998

DESCRIPTION	NOT
DTSTART	NOT
DTEND	NOT
DURATION	NOT
EXDATE	NOT
EXRULE	NOT
GEO	NOT
LAST-MODIFIED	NOT
ORGANIZER	NOT
PRIORITY	NOT
RDATE	NOT
RELATED-TO	NOT
RESOURCES	NOT
REQUEST-STATUS	NOT
RRULE	NOT
SEQUENCE	NOT
SUMMARY	NOT
STATUS	NOT
TRANSP	NOT
URL	NOT

VTIMEZONE	MUST be present if any date/time refers to a timezone
-----------	---

X-TOKENS	MAY
VTODO	NOT
VJOURNAL	NOT
VFREEBUSY	NOT
VALARM	NOT

3.2.7 COUNTER

The "COUNTER" method for a "VEVENT" calendar component is used by an "Attendee" of an existing event to submit to the "Organizer" a counter proposal to the event description. The "Attendee" sends this message to the "Organizer" of the event.

The counter proposal is an iCalendar object consisting of a VEVENT calendar component describing the complete description of the alternate event.

The "Organizer" rejects the counter proposal by sending the "Attendee" a VEVENT "DECLINE-COUNTER" method. The "Organizer" accepts the counter proposal by sending all of the "Attendees" to the event a VEVENT "REQUEST" method rescheduling the event. In the latter case, the "Organizer" SHOULD reset the individual "RSVP" property parameter values to TRUE on each "ATTENDEE" property in order to force a response by the "Attendees".

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property Presence

Silverberg/Mansour/Dawson/Hopson 24

Expires September 1998

Internet Draft

iTIP

March 13, 1998

PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "COUNTER"
VEVENT	MUST
DTSTAMP	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it must NOT be present.
SEQUENCE	MUST be present if non-zero, MAY be present if zero
SUMMARY	MUST be present may be NULL
UID	MUST be the UID associated with the REQUEST being countered
ATTACH	MAY
ATTENDEE	MAY be used to propose other "Attendees"
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY

X-TOKENS	MAY
VTODO	NOT
VJOURNAL	NOT
VFREEBUSY	NOT
VALARM	NOT

3.3 Methods For VFREEBUSY Components

This section defines the property set for the methods that are applicable to the "VFREEBUSY" calendar component. Each of the methods is defined using a restriction table.

This document only addresses the transfer of busy time information. Applications desiring free time information MUST infer this from available busy time information.

The busy time information within the iCalendar object MAY be grouped into more than one "VFREEBUSY" calendar component. This capability allows busy time periods to be grouped according to some common periodicity, such as a calendar week, month, or year. In this case, each "VFREEBUSY" calendar component MUST include the "ATTENDEE", "DTSTART" and "DTEND" properties in order to specify the source of the busy time information and the date and time interval over which the busy time information covers.

The "FREEBUSY" property value MAY include a list of values, separated by the COMMA character (ASCII decimal 44). Alternately, multiple busy time periods MAY be specified with multiple instances of the "FREEBUSY" property. Both forms MUST be supported by implementations conforming to this document. Duplicate busy time periods SHOULD NOT be specified in an iCalendar object. However, two different busy time periods MAY overlap.

"FREEBUSY" properties should be sorted such that their values are in ascending order, based on the start time, and then the end time, with the earliest periods first. For example, today's busy time information should appear after yesterday's busy time information. And the busy time for this half-hour should appear after the busy time for earlier today.

Since events may span a day boundary, free busy time period may also span a day boundary. Individual "A" requests busy time from individuals "B", "C" and "D". Individual "B" and "C" replies with busy time data to individual "A". Individual "D" does not support busy time requests and does not reply with any data. If the transport binding supports exception messages, then individual "D" returns an "unsupported capability" message to individual "A".

The following summarizes the methods that are defined for the "VFREEBUSY" calendar component.

|=====|=====|

Silverberg/Mansour/Dawson/Hopson 27

Expires September 1998

Internet Draft

iTIP

March 13, 1998

Method	Description
PUBLISH	Publish unsolicited busy time data.
REQUEST	Request busy time data.
REPLY	Reply to a busy time request.

[3.3.1](#) PUBLISH

The "PUBLISH" method in a "VFREEBUSY" calendar component is used to publish busy time data. The method may be sent from one CU to any other. The purpose of the method is to provide a message for sending unsolicited busy time data. That is, the busy time data is not being sent as a "REPLY" to the receipt of a "REQUEST" method.

The "ATTENDEE" property must be specified in the busy time information. The value is the CU address of the originator of the busy time information.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "PUBLISH"
VFREEBUSY	MUST
ATTENDEE	MUST contain the address of originator of busy time data
DTSTAMP	MUST
FREEBUSY	MUST be BUSYTIME. Multiple instances are allowed. Multiple instances must be sorted in ascending order.
ATTACH	MAY
RELATED-TO	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY specifies when the busy time data was Created

DTSTART	MAY be present to represent start of the interval for busy time data
DTEND	MAY be present to represent the end of the interval for busy time data
GEO	MAY
ORGANIZER	MAY
URL	MAY (specifies busy time URL)
CATEGORIES	NOT
CLASS	NOT
DESCRIPTION	NOT

Silverberg/Mansour/Dawson/Hopson 28

Expires September 1998

Internet Draft

iTIP

March 13, 1998

DURATION	NOT
EXDATE	NOT
EXRULE	NOT
LAST-MODIFIED	NOT
PRIORITY	NOT
RDATE	NOT
RECURRENCE-ID	NOT
REQUEST-STATUS	NOT
RESOURCES	NOT
RRULE	NOT
SEQUENCE	NOT
STATUS	NOT
SUMMARY	NOT
TRANSP	NOT
UID	NOT

VTIMEZONE	MUST be present if any date/time refers to a timezone
-----------	---

X-TOKENS	NOT
VEVENT	NOT
VTODO	NOT
VJOURNAL	NOT
VALARM	NOT

[3.3.2](#) REQUEST

The "REQUEST" method in a "VFREEBUSY" calendar component is used to ask a "Calendar User" for their busy time information. The request may be for a busy time information bounded by a specific date and time interval.

This message only permits requests for busy time information. The

message is sent from a "Calendar User" requesting the busy time information to one or more intended recipients.

If the originator of the "REQUEST" method is not authorized to make a busy time request on the recipient's calendar system, then an exception message SHOULD be returned in a "REPLY" method, but no busy time data need be returned.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "REQUEST"
VFREEBUSY	MUST
ATTENDEE	MUST contain the address of the calendar store for which busy time is being requested.

Silverberg/Mansour/Dawson/Hopson 29

Expires September 1998

Internet Draft

iTIP

March 13, 1998

DTEND	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST be the request originator's address
ATTACH	MAY
COMMENT	MAY
CONTACT	MAY
GEO	MAY
URL	MAY
CATEGORIES	NOT
CLASS	NOT
CREATED	NOT
DESCRIPTION	NOT
DURATION	NOT
EXDATE	NOT
EXRULE	NOT
FREEBUSY	NOT
LAST-MODIFIED	NOT
PRIORITY	NOT
RDATE	NOT
RECURRENCE-ID	NOT

	type. Multiple instances are allowed. Multiple instances MUST be sorted in ascending order. Values MAY NOT overlap)
ORGANIZER	MUST be the request originator's address
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it must NOT be present.
UID	MUST
ATTACH	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY specifies when the busy time data was created)
DTSTART	MAY (represents start of interval for busy time data}, DTEND{represents end of interval for busy time data)
GEO	MAY
RELATED-TO	MAY
REQUEST-STATUS	MAY
SEQUENCE	MAY be present if non-zero
URL	MAY (specifies busy time URL)
CATEGORIES	NOT
CLASS	NOT
DESCRIPTION	NOT
DURATION	NOT
EXDATE	NOT
EXRULE	NOT
FREEBUSY	NOT
LAST-MODIFIED	NOT
PRIORITY	NOT
RDATE	NOT
RESOURCES	NOT
RRULE	NOT
SEQUENCE	NOT
STATUS	NOT
SUMMARY	NOT
TRANSP	NOT
VTIMEZONE	MUST be present if any date/time refers to a

Silverberg/Mansour/Dawson/Hopson 31

Expires September 1998

Internet Draft

iTIP

March 13, 1998

timezone

X-TOKENS

MAY

VEVENT	NOT
VTOD0	NOT
VJOURNAL	NOT
VALARM	NOT

3.4 Methods For VTOD0 Components

This section defines the property set for the methods that are applicable to the "VTOD0" calendar component. Each of the methods is defined using a restriction table that specifies the property constraints that define the particular method.

The following summarizes the methods that are defined for the "VTOD0" calendar component.

Method	Description
PUBLISH	Post notification of a VTOD0. Used primarily as a method of advertising the existence of a VTOD0.
REQUEST	Assign a VTOD0. This is an explicit assignment to one or more Calendar Users. The REQUEST method is also used to update or change an existing VTOD0. Clients that cannot handle REQUEST MAY degrade the method to treat it as a PUBLISH.
REPLY	Reply to a VTOD0 request. Attendees MAY set ATTSTAT to ACCEPTED, DECLINED, TENTATIVE, DELEGATED, PARTIAL, and COMPLETED.
ADD	Add one or more instances to an existing to-do.
CANCEL	Cancel one or more instances of an existing to-do.
REFRESH	A request sent to a VTOD0 "Organizer" asking for the latest version of a VTOD0.
COUNTER	Counter a REQUEST with an alternative proposal.
DECLINECOUNTER	Decline a counter proposal by an "Attendee."

Internet Draft

iTIP

March 13, 1998

3.4.1 PUBLISH

The "PUBLISH" method in a "VTODO" calendar component has no associated response. It is simply a posting of an iCalendar object that maybe added to a calendar by a "Calendar User". Its expected usage is for encapsulating an arbitrary "VTODO" calendar component as an iCalendar object. The "Organizer" MAY subsequently update (with another "PUBLISH" method), add instances to (win an "ADD" method), or cancel (with a "CANCEL" method) a previously published "VTODO" calendar component.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "PUBLISH"
VTODO	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it must NOT be present.
SEQUENCE	MUST be present if not 0, may be present if 0
SUMMARY	MUST; may be null.
UID	MUST
ATTENDEE	MAY
ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY be present and MAY be NULL
DURATION	MAY
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY

PERCENT-COMPLETE	MAY
PRIORITY	MAY
RELATED-TO	MAY
RDATE	MAY
RESOURCES	MAY
RRULE	MAY
STATUS	MAY be one of COMPLETED/NEEDS ACTION/IN-PROCESS
TRANSP	MAY
URL	MAY

Silverberg/Mansour/Dawson/Hopson 33

Expires September 1998

Internet Draft

iTIP

March 13, 1998

VEVENT	NOT
VJOURNAL	NOT
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
VFREEBUSY	NOT
X-TOKENS	MAY

3.4.2 REQUEST

The "REQUEST" method in a "VTOD0" calendar component provides the following scheduling functions:

- . Assign a to-do to one or more "Calendar Users";
- . Reschedule an existing to-do;
- . Update the details of an existing to-do, without rescheduling it;
- . Update the completion status of "Attendees" of an existing to-do, without rescheduling it;
- . Reconfirm an existing to-do, without rescheduling it;
- . Delegate/reassign an existing to-do to another "Calendar User".

The assigned "Calendar Users" are identified in the "VTOD0" calendar component by individual "ATTENDEE;ROLE=REQ-PARTICIPANT" property value sequences.

The originator of a "REQUEST" is the "Organizer" of the to-do. The recipient of a "REQUEST" is the "Calendar User" assigned the to-do-the "Attendee". The "Attendee" uses the "REPLY" method to convey their acceptance and completion status to the "Organizer" of the "REQUEST".

The "UID", "SEQUENCE", and "DTSTAMP" properties are used to distinguish the various uses of the "REQUEST" method. If the "UID" property value in

the "REQUEST" is not found on the recipient's calendar, then the "REQUEST" is for a new to-do. If the "UID" property value is found on the recipient's calendar, then the "REQUEST" is a rescheduling, an update, or a reconfirm of the "VTODO" calendar object.

If the "Organizer" of the "REQUEST" method is not authorized to make a to-do request on the "Attendee's" calendar system, then an exception is returned in the "REQUEST-STATUS" property of a subsequent "REPLY" method, but no scheduling action is performed.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property Presence

PROPID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "REQUEST"

Silverberg/Mansour/Dawson/Hopson 34

Expires September 1998

Internet Draft

iTIP

March 13, 1998

VTODO	MUST
ATTENDEE	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it must NOT be present.
SEQUENCE	MUST be present if not 0, may be present if 0
SUMMARY	MUST; may be null.
UID	MUST
ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY be present and MAY be NULL
DURATION	MAY
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY

"REQUEST" is the same as the value for the existing event, then the "REQUEST" method describes an update of the "VTODO" calendar component details, but not a rescheduling of the "VTODO" calendar component.

The update "REQUEST" is the appropriate response to a "REFRESH" method sent from an "Attendee" to the "Organizer" of a "VTODO" calendar component.

Unsolicited "REQUEST" methods MAY be sent by the "Organizer" of a "VTODO" calendar component. The unsolicited "REQUEST" methods are used to update the details of the "VTODO" (without rescheduling it or updating the completion status of "Attendees") or the "VTODO" calendar component itself (i.e., reconfirm the "VTODO").

3.4.2.3 REQUEST for Delegating a VTODO

The "REQUEST" method is also used to delegate or reassign ownership of a "VTODO" calendar component to another "Calendar User". For example, it may be used to delegate an "Attendee's" role (i.e. "chair", or "participant") for a "VTODO" calendar component. The "REQUEST" method is sent by one of the "Attendees" of an existing "VTODO" calendar component to some other individual. An "Attendee" of a "VTODO" calendar component MUST NOT delegate to the "Organizer" of the event.

For the purposes of this description, the "Attendee" delegating the "VTODO" calendar component is referred to as the "Delegator". The "Attendee" receiving the delegation request is referred to as the "Delegate".

The "Delegator" of a "VTODO" calendar component MUST forward the existing "REQUEST" method for a "VTODO" calendar component to the "Delegate". The "VTODO" calendar component description MUST include the "Delegator's" up-to-date "VTODO" calendar component definition. The "REQUEST" method MUST also include an "ATTENDEE" property with the calendar address of the "Delegate". The "Delegator" MUST also send a "REPLY" method back to the "Organizer" with the "Delegator's" "Attendee" property "attstat" parameter value set to "DELEGATED". In addition, the "delegated-to" parameter SHOULD be included with the calendar address of

Silverberg/Mansour/Dawson/Hopson 36

Expires September 1998

Internet Draft

iTIP

March 13, 1998

the "Delegate". A response to the delegation "REQUEST" is sent from the "Delegate" to the "Organizer" and optionally, to the "Delegator". The "REPLY" method from the "Delegate" SHOULD include the "ATTENDEE" property with their calendar address and the "delegated-from" parameter with the value of the "Delegator's" calendar address.

The delegation "REQUEST" method MUST assign a value for the "RSVP" property parameter associated with the "Delegator's" "Attendee" property to that of the "Delegate's" "ATTENDEE" property. For example if the "Delegator's" "ATTENDEE" property specifies "RSVP=TRUE", then the "Delegate's" "ATTENDEE" property MUST specify "RSVP=TRUE".

3.4.2.4 REQUEST Forwarded To An Uninvited Calendar User

An "Attendee" assigned a "VTODO" calendar component may assign the "VTODO" calendar component to another new "Calendar User", not previously associated with the "VTODO" calendar component. The current "Attendee" assigned the "VTODO" calendar component accomplishes this scheduling action by forwarding the original "REQUEST" method to the new "Calendar User".

3.4.2.5 REQUEST Updated Attendee Status

An "Organizer" of a "VTODO" may request an updated status from one or more "Attendees". The "Organizer" sends a "REQUEST" method to the "Attendee" with the "ATTENDEE;RSVP=TRUE" property sequence. The "SEQUENCE" property for the "VTODO" is not changed from its previous value. A recipient determines that the only change in the "REQUEST" is that their "RSVP" property parameter indicates a request for an updated status. The recipient SHOULD respond with a "REPLY" method indicating their current status with respect to the "REQUEST".

3.4.3 REPLY

The "REPLY" method in a "VTODO" calendar component is used to respond (e.g., accept or decline) to a request or to reply to a delegation request. It is also used by an "Attendee" to update their completion status. When used to provide a delegation response, the "Delegator" SHOULD include the calendar address of the "Delegate" in the "delegated-to" parameter of the "Delegator's" "ATTENDEE" property. The "Delegate" SHOULD include the calendar address of the "Delegator" on the "delegated-from" parameter of the "Delegate's" "ATTENDEE" property.

The "REPLY" method MAY also be used to respond to an unsuccessful "VTODO" calendar component "REQUEST" method. Depending on the "REQUEST-STATUS" value, no scheduling action may have been performed.

The "Organizer" of a "VTODO" calendar component MAY receive a "REPLY" method from a "Calendar User" not in the original "REQUEST". For example, a "REPLY" method MAY be received from a "Delegate" of a "VTODO" calendar component. In addition, the "REPLY" method MAY be received from

an unknown "Calendar User", having been forwarded the "REQUEST" by an original "Attendee" of the "VTODO" calendar component. This uninvited "Attendee" MAY be accepted, or the "Organizer" MAY cancel the "VTODO" calendar component for the uninvited "Attendee" by sending them a "CANCEL" method.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "REPLY"
VTODO	MUST
ATTENDEE	MUST
UID	MUST must be the address of the replying attendee
DTSTAMP	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a Recurring calendar component. Otherwise it Must NOT be present
SEQUENCE	MUST
COMMENT	MAY
PERCENT-COMPLETE	MAY
REQUEST-STATUS	MAY
DTSTART	NOT
CREATED	NOT
DESCRIPTION	NOT
PRIORITY	NOT
CLASS	NOT
CATEGORIES	NOT
CONTACT	NOT
CREATED	NOT
DTEND	NOT
GEO	NOT
LAST-MODIFIED	NOT
LOCATION	NOT
RDATE	NOT
RRULE	NOT
RELATED-TO	NOT
RESOURCE	NOT
STATUS	NOT

TRANSP	NOT
URL	NOT
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	NOT
VEVENT	NOT
VFREEBUSY	NOT
X-TOKENS	NOT

Silverberg/Mansour/Dawson/Hopson 38

Expires September 1998

Internet Draft

iTIP

March 13, 1998

[3.4.4](#) **ADD**

The "ADD" method in a "VTODO" calendar component is used to add one or more instances to an existing to-do.

If the "UID" property value in the "ADD" is not found on the recipient's calendar, then the recipient SHOULD send a "REFRESH" to the "Organizer" in order to be updated with the latest version of the "VTODO". If an "Attendee" implementation does not support the "ADD" method it should respond with a "REQUEST-STATUS" value of 5.3 and ask for a "REFRESH".

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property Presence

PROPID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "ADD"
VTODO	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST
SEQUENCE	MUST be greater than 0
SUMMARY	MUST; may be null.
UID	MUST match that of the original to-do
ATTENDEE	MAY
ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY

DESCRIPTION	MAY be present and MAY be NULL
DTEND	MAY
DURATION	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY
PERCENT-COMPLETE	MAY
PRIORITY	MAY
RELATED-TO	MAY
RDATE	MAY
RESOURCES	MAY
RRULE	MAY
STATUS	MAY be one of COMPLETED/NEEDS ACTION/IN-PROCESS
TRANSP	MAY
URL	MAY

Silverberg/Mansour/Dawson/Hopson 39

Expires September 1998

Internet Draft

iTIP

March 13, 1998

RECURRENCE-ID	NOT
REQUEST-STATUS	NOT
VEVENT	NOT
VJOURNAL	NOT
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
VFREEBUSY	NOT
X-TOKENS	MAY

[3.4.5](#) CANCEL

The "CANCEL" method in a "VTODO" calendar component is used to send a cancellation notice of an existing "VTODO" calendar request to the "Attendees". The message is sent by the "Organizer" of a "VTODO" calendar component to the "Attendees" of the "VTODO" calendar component. For a recurring "VTODO" calendar component, either the whole "VTODO" calendar component or instances of a "VTODO" calendar component may be cancelled. To cancel the complete range of a recurring "VTODO" calendar component, the "UID" property value for the "VTODO" calendar component MUST be specified and a "RECURRENCE-ID" MUST NOT be specified in the "CANCEL" method. In order to cancel an individual instance of a recurring "VTODO" calendar component, the "RECURRENCE-ID" property value for the "VTODO" calendar component MUST be specified in the "CANCEL"

method.

There are two options for canceling a sequence of instances of a recurring "VTODO" calendar component:

- (a) the "RECURRENCE-ID" property for an instance in the sequence MUST be specified with the "RANGE" property parameter value of THISANDPRIOR (or THISANDFUTURE) to indicate cancellation of the specified "VTODO" calendar component and all instances before (or after); or
- (b) individual recurrence instances may be cancelled by specifying multiple "RECURRENCE-ID" properties corresponding to the instances to be cancelled.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "CANCEL"
VTODO	MUST
UID	MUST must echo original UID
DTSTAMP	MUST
ORGANIZER	MUST
SEQUENCE	MUST

Silverberg/Mansour/Dawson/Hopson 40

Expires September 1998

Internet Draft

iTIP

March 13, 1998

RECURRENCE-ID	MUST only if referring to one or more instances of a recurring calendar component. Otherwise it MUST NOT be present
COMMENT	MAY
ATTENDEE	MAY
DTSTART	MAY
CREATED	MAY
DESCRIPTION	MAY
ORGANIZER	MAY
PRIORITY	MAY
CLASS	MAY
CATEGORIES	MAY
CONTACT	MAY
CREATED	MAY
DTEND	MAY
GEO	MAY

LAST-MODIFIED	MAY
LOCATION	MAY
PERCENT-COMPLETE	MAY
PRIORITY	MAY
RDATE	MAY
RRULE	MAY
RELATED-TO	MAY
REQUEST-STATUS	MAY
RESOURCE	MAY
STATUS	MAY If present it must be set to "CANCELLED". MUST NOT be used if purpose is to remove "ATTENDEES" rather than cancel the entire VTODO.
TRANSP	MAY
URL	MAY
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
VEVENT	MAY
VFREEBUSY	MAY
X-TOKENS	MAY

3.4.6 REFRESH

The "REFRESH" method in a "VTODO" calendar component is used by "Attendees" of an existing "VTODO" calendar component to request an updated description from the "Organizer" of the "VTODO" calendar component. The "Organizer" of the "VTODO" calendar component MAY use this method to request an updated status from the "Attendees". The "REFRESH" method MUST specify the "UID" property corresponding to the "VTODO" calendar component needing update.

A refresh of a recurrence instance of a "VTODO" calendar component may be requested by specifying the "RECURRENCE-ID" property corresponding to the associated "VTODO" calendar component. The "Organizer" responds with the latest description and rendition of the "VTODO" calendar component.

Silverberg/Mansour/Dawson/Hopson 41

Expires September 1998

Internet Draft

iTIP

March 13, 1998

This method is intended to facilitate machine processing of requests for updates to a "VTODO" calendar component.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PROID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "REFRESH"
VTODO	MUST
ATTENDEE	MUST
DTSTAMP	MUST
RECURRENCE-ID	MUST only if referring to an instance of a Recurring calendar component. Otherwise it MUST NOT be present
UID	MUST echo original UID
COMMENT	MAY
REQUEST-STATUS	MAY
DTSTART	MAY
CREATED	MAY
DESCRIPTION	MAY
PRIORITY	MAY
CLASS	MAY
CATEGORIES	MAY
CONTACT	MAY
CREATED	MAY
DTEND	MAY
GEO	MAY
LAST-MODIFIED	MAY
PERCENT-COMPLETE	MAY
ORGANIZER	MAY
PRIORITY	MAY
LOCATION	MAY
RDATE	MAY
RRULE	MAY
RELATED-TO	MAY
RESOURCE	MAY
SEQUENCE	MAY
STATUS	MAY
TRANSP	MAY
URL	MAY
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
VEVENT	MAY
VFREEBUSY	MAY
X-TOKENS	MAY

3.4.7 COUNTER

The "COUNTER" method in a "VTODO" calendar component is used by an "Attendee" of an existing "VTODO" calendar component to submit to the "Organizer" a counter proposal for the "VTODO" calendar component. The "Attendee" sends the message to the "Organizer" of the "VTODO" calendar component.

The counter proposal is an iCalendar object consisting of a "VTODO" calendar component describing the complete description of the alternate "VTODO" calendar component.

The "Organizer" rejects the counter proposal by sending the "Attendee" a "DECLINE-COUNTER" method. The "Organizer" accepts the counter proposal by sending all of the "Attendees" of the "VTODO" calendar component a "REQUEST" method rescheduling the "VTODO" calendar component. In the latter case, the "Organizer" SHOULD reset the individual "RSVP" property parameter values to TRUE on each "ATTENDEE" property; in order to force a response by the "Attendees".

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "COUNTER"
VTODO	MUST
DTSTAMP	MUST
DTSTART	MAY
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it MUST NOT be present.
SEQUENCE	MUST be present if NOT 0, MAY be present if 0
SUMMARY	MUST be present; MAY be NULL
UID	MUST
ATTENDEE	MAY
ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY

VTODO	MUST
ATTENDEE	MUST for all attendees
UID	MUST must echo original UID
DTSTAMP	MUST
SEQUENCE	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it MUST NOT be present.
COMMENT	MAY
PERCENT-COMPLETE	MAY
REQUEST-STATUS	MAY
DTSTART	MAY
CREATED	MAY
DESCRIPTION	MAY
ORGANIZER	MAY
PRIORITY	MAY

Silverberg/Mansour/Dawson/Hopson 44

Expires September 1998

Internet Draft

iTIP

March 13, 1998

CLASS	MAY
CATEGORIES	MAY
CONTACT	MAY
CREATED	MAY
DTEND	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY
RDATE	MAY
RRULE	MAY
RELATED-TO	MAY
RESOURCE	MAY
STATUS	MAY be one of COMPLETED/NEEDS ACTION/IN-PROCESS
TRANSP	MAY
URL	MAY

VTIMEZONE	MUST be present if any date/time refers to a timezone
-----------	--

VALARM	MAY
VEVENT	MAY
VTODO	MAY
VFREEBUSY	MAY
X-TOKENS	MAY

[3.5](#) Methods For VJOURNAL Components

This section defines the property set for the methods that are

applicable to the "VJOURNAL" calendar component.

The following summarizes the methods that are defined for the "VJOURNAL" calendar component.

Method	Description
PUBLISH	Post a journal entry. Used primarily as a method of advertising the existence of a journal entry.
ADD	Add one or more instances to an existing journal entry.
CANCEL	Cancel one or more instances of an existing journal entry.

[3.5.1](#) PUBLISH

The "PUBLISH" method in a "VJOURNAL" calendar component has no associated response. It is simply a posting of an iCalendar object that may be added to a calendar by a CUA. There is no response to the "Organizer". The expected usage is for encapsulating an arbitrary journal entry as an iCalendar object. The "Organizer" MAY subsequently

Silverberg/Mansour/Dawson/Hopson 45

Expires September 1998

Internet Draft

iTIP

March 13, 1998

update (with another "PUBLISH" method) or cancel (with a "CANCEL" method) a previously published journal entry.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
PROPID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "PUBLISH"
VJOURNAL	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it

	MUST NOT be present.
SEQUENCE	MUST be present if not 0, MAY be present if 0
SUMMARY	MUST be present and MAY be NULL
UID	MUST
ATTENDEE	MAY
ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY be present; MAY be NULL.
DURATION	MAY
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY
PRIORITY	MAY
RELATED-TO	MAY
RDATE	MAY
RESOURCES	MAY
RRULE	MAY
STATUS	MAY be one of DRAFT/FINAL/CANCELLED
TRANSP	MAY
URL	MAY
VEVENT	MAY
VTODO	MAY
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
VFREEBUSY	MAY
X-TOKENS	MAY

Silverberg/Mansour/Dawson/Hopson 46

Expires September 1998

Internet Draft

iTIP

March 13, 1998

[3.5.2](#) **ADD**

The "ADD" method in a "VJOURNAL" calendar component is used to add one or more instances to an existing "VJOURNAL" entry. There is no response to the "Organizer".

If the "UID" property value in the "ADD" is not found on the recipient's

calendar, then the recipient MAY treat the "ADD" as a "PUBLISH".

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "ADD"
VJOURNAL	MUST
DTSTAMP	MUST
DTSTART	MUST
ORGANIZER	MUST
SEQUENCE	MUST be greater than 0
SUMMARY	MUST be present and MAY be NULL
UID	MUST match that of the original journal
ATTACH	MAY
CATEGORIES	MAY
CLASS	MAY
COMMENT	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY be present; may be null.
DURATION	MAY
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
LOCATION	MAY
PRIORITY	MAY
RELATED-TO	MAY
RDATE	MAY
RESOURCES	MAY
RRULE	MAY
STATUS	MAY be one of DRAFT/FINAL/CANCELLED
TRANSP	MAY
URL	MAY
ATTENDEE	NOT
RECURRENCE-ID	NOT
VEVENT	MAY
VTODO	MAY
VTIMEZONE	MUST be present if any date/time refers to a

	timezone
VALARM	MAY
VFREEBUSY	MAY
X-TOKENS	MAY

[3.5.3](#) CANCEL

The "CANCEL" method in a "VJOURNAL" calendar component is used to send a cancellation notice of an existing journal entry. The message is sent by the "Organizer" of a journal entry. For a recurring journal entry, either the whole journal entry or instances of a journal entry may be cancelled. To cancel the complete range of a recurring journal entry, the "UID" property value for the journal entry **MUST** be specified and a "RECURRENCE-ID" property **MUST NOT** be specified in the "CANCEL" method. In order to cancel an individual instance of the journal entry, the "RECURRENCE-ID" property value for the journal entry **MUST** be specified in the "CANCEL" method.

There are two options for canceling a sequence of instances of a recurring "VJOURNAL" calendar component:

- (a) the "RECURRENCE-ID" property for an instance in the sequence **MUST** be specified with the "RANGE" property parameter value of **THISANDPRIOR** (or **THISANDFUTURE**) to indicate cancellation of the specified "VTODO" calendar component and all instances before (or after); or
- (b) individual recurrence instances may be cancelled by specifying multiple "RECURRENCE-ID" properties corresponding to the instances to be cancelled.

This method type is an iCalendar object that conforms to the following property constraints:

Component/Property	Presence
-----	-----
PRODID	MUST
VERSION	MUST be "2.0"
METHOD	MUST be "CANCEL"
VJOURNAL	MUST
DTSTAMP	MUST
ORGANIZER	MUST
RECURRENCE-ID	MUST only if referring to an instance of a recurring calendar component. Otherwise it MUST NOT be present.
SEQUENCE	MUST
UID	MUST be the UID of the original REQUEST

COMMENT	MAY
STATUS	MAY be present, must be "CANCELLED" if present
ATTACH	MAY
ATTENDEE	MAY
CATEGORIES	MAY

Silverberg/Mansour/Dawson/Hopson 48

Expires September 1998

Internet Draft

iTIP

March 13, 1998

CLASS	MAY
CONTACT	MAY
CREATED	MAY
DESCRIPTION	MAY
DTSTART	MAY
DTEND	MAY
EXDATE	MAY
EXRULE	MAY
GEO	MAY
LAST-MODIFIED	MAY
%-COMPLETE	MAY
PRIORITY	MAY
RELATED-TO	MAY
REQUEST-STATUS	MAY
RESOURCES	MAY
RDATE	MAY
RRULE	MAY
STATUS	MAY
SUMMARY	MAY
TRANSP	MAY
URL	MAY

VTODO	MAY
VEVENT	MAY
VFREEBUSY	MAY
VTIMEZONE	MUST be present if any date/time refers to a timezone
VALARM	MAY
X-TOKENS	MAY

3.6 Status Replies

The "REQUEST-STATUS" property may include the following values:

Short Return	Longer Return Status	Offending Data

Status Code	Description	
2.0	Success.	None.
2.1	Success but fallback taken on one or more property values.	Property name and value MAY be specified.
2.2	Success, invalid property ignored.	Property name MAY be specified.
2.3	Success, invalid property parameter ignored.	Property parameter name and value MAY be specified.
2.4	Success, unknown non-	Non-standard property

Silverberg/Mansour/Dawson/Hopson 49

Expires September 1998

Internet Draft

iTIP

March 13, 1998

	standard property ignored.	name MAY be specified.
2.5	Success, unknown non standard property value ignored.	Property and non-standard value MAY be specified.
2.6	Success, invalid calendar component ignored.	Calendar component sentinel (e.g., "BEGIN: ALARM") MAY be specified.
2.7	Success, request forwarded to Calendar User.	Original and forwarded caluser addresses MAY be specified.
2.8	Success, repeating event ignored. Scheduled as a single event.	RRULE or RDATE property name and value MAY be specified.
2.9	Success, truncated end date time to date boundary.	DTEND property value MAY be specified.
2.10	Success, repeating VTOD0 ignored. Scheduled as a single VTOD0.	RRULE or RDATE property name and value MAY be specified.
2.11	Success, RRULE or EXRULE	RRULE or EXRULE property

	too complex. Scheduled as a single event.	name and value MAY be specified.
2.12	Success, unbounded RRULE clipped at some finite number of instances	RRULE property name and value MAY be specified. Number of instances MAY also be specified.
3.0	Invalid property name.	Property name MAY be specified.
3.1	Invalid property value.	Property name and value MAY be specified.
3.2	Invalid property parameter.	Property parameter name and value MAY be specified.
3.3	Invalid property parameter value.	Property parameter name and value MAY be specified.
3.4	Invalid calendar component sequence.	Calendar component sentinel MAY be specified (e.g., BEGIN: VTIMEZONE).

Silverberg/Mansour/Dawson/Hopson 50

Expires September 1998

Internet Draft

iTIP

March 13, 1998

3.5	Invalid date or time.	Date/time value(s) MAY be specified.
3.6	Invalid rule.	Rule value MAY be specified.
3.7	Invalid Calendar User.	Attendee property value MAY be specified.
3.8	No authority.	PROFILE and Attendee property values MAY be specified.
3.9	Unsupported version.	VERSION property name and value MAY be specified.

3.10	Request entity too large.	None.
3.11	Required component or property missing.	Component or property name MAY be specified.
4.0	Event conflict. Date/time is busy.	DTSTART and DTEND property name and values MAY be specified.
5.0	Request MAY supported.	Method property value MAY be specified.
5.1	Service unavailable.	ATTENDEE property value MAY be specified.
5.2	Invalid calendar service.	ATTENDEE property value MAY be specified.
5.3	No scheduling support for user.	ATTENDEE property value MAY be specified.

3.7 Implementation Considerations

3.7.1 Working With Recurrence Instances

iCalendar includes a recurrence grammar to represent recurring events. The benefit of such a grammar is the ability to represent a number of events in a single object. However, while this simplifies creation of a recurring event, meeting instances still need to be referenced. For instance, an "Attendee" may decline the third instance of a recurring Friday event. Similarly, the "Organizer" may change the time or location to a single instance of the recurring event.

Silverberg/Mansour/Dawson/Hopson 51

Expires September 1998

Internet Draft

iTIP

March 13, 1998

Since implementations may elect to store recurring events as either a single event object or a collection of discreet, related event objects, the protocol is designed so that each recurring instance may be both referenced and versioned. Hence, implementations that choose to maintain per-instance properties (such as "ATTENDEE" property "attstat" parameter) may do so. However, the protocol does not require per-

instance recognition unless the instance itself must be renegotiated.

The scenarios for recurrence instance referencing are listed below. For purposes of simplification a change to an event refers to a "trigger property." That is, a property that has a substantive effect on the meeting itself such as start time, location, due date (for "VTOD0" calendar component components) and possibly description.

"Organizer" initiated actions:

- . "Organizer" deletes or changes a single instance of a recurring event
- . "Organizer" makes changes that affect all future instances
- . "Organizer" makes changes that affect all previous instances
- . "Organizer" deletes or modifies a previously changed instance

"Attendee" initiated actions:

- . "Attendee" changes status for a particular recurrence instance
- . "Attendee" sends Event-Counter for a particular recurrence instance

An instance of a recurring event is assigned a unique identification, "RECURRENCE-ID" property, when that instance is renegotiated. Negotiation may be necessary when a substantive change to the event or to-do has been made (such as changing the start time, end time, due date or location). The "Organizer" can identify a specific recurrence instance using the "RECURRENCE-ID" property. The property value is equal to the date/time of the instance. If the "Organizer" wishes to change the "DTSTART", the original "DTSTART" value is used for "RECURRENCE-ID" property and the new "DTSTART" and "DTEND" values reflect the change. Note that after the change has occurred, the "RECURRENCE-ID" has changed to the new "DTSTART" value.

3.7.2 Attendee Property Considerations

The "ORGANIZER" property is required on published events, to-dos, and journal entries for two reasons. First, only the "Organizer" is allowed to update and redistribute an event or to-do component. It follows that the "ORGANIZER" property MUST be present in the event, to-do, or journal entry component so that the CUA has a basis for authorizing an update. Second, it is prudent to provide a point of contact for anyone who receives a published component in case of problems.

There are valid [RFC 822](#) addresses that represent groups. Sending email to such an address results in mail being sent to multiple recipients. Such an address may be used as the value of an "ATTENDEE" property.

Thus, it is possible that the recipient of a "REQUEST" does not appear explicitly in the list.

It is recommended that the general approach to finding a "Calendar User" in an attendee list be as follows:

1. Search for the "Calendar User" in the attendee list where "TYPE=INDIVIDUAL"
2. Failing (1) look for attendees where "TYPE=GROUP" or "TYPE=UNKNOWN". The CUA then determines if the "Calendar User" is a member of one of these groups. If so, the "REPLY" method sent to the "Organizer" MUST contain a new "ATTENDEE" property in which the "TYPE" property parameter is set to INDIVIDUAL and the "group" property parameter is set to the name of the group.
3. Failing (2) the CUA MAY ignore or accept the request as the "Calendar User" wishes.

3.7.3 X-Tokens

To make iCalendar objects extensible, new property types MAY be inserted into components. These properties are called X-Tokens as they are prefixed with "X-". A client is not required to make sense of X-Tokens. Clients are not required to save X-Tokens or use them in replies.

4 Examples

4.1 Published Event Examples

In the calendaring and scheduling context, publication refers to the one way transfer of event information. Consumers of published events simply incorporate the event into a calendar. No reply is expected. Individual "A" publishes an event. Individual "B" reads the event and incorporates it into their calendar. Events are published in several ways including: embedding the event as an object in a web page, e-mailing the event to a distribution list, and posting the event to a newsgroup.

The table below illustrates the sequence of events between the publisher and the consumers of a published event.

Action	"Organizer"
Publish an event	"A" sends or posts a PUBLISH

	message	
+-----+	+-----+	+-----+
"B" reads a published event		
+-----+	+-----+	+-----+
Publish an updated event	"A" sends or posts a PUBLISH	
	message	

Silverberg/Mansour/Dawson/Hopson 53

Expires September 1998

Internet Draft

iTIP

March 13, 1998

+-----+	+-----+	+-----+
"B" reads the updated event		
+-----+	+-----+	+-----+
Cancel a published event	"A" sends or posts a CANCEL	
	message	
+-----+	+-----+	+-----+
"B" reads the canceled event		
publication		
+-----+	+-----+	+-----+

[4.1.1](#) A Minimal Published Event

The iCalendar object below describes a single event that begins on July 1, 1997 at 20:00 UTC. This event contains the minimum set of properties for a "PUBLISH" for a "VEVENT" calendar component.

```

BEGIN:VCALENDAR
METHOD:PUBLISH
PROID:-//ACME/DesktopCalendar//EN
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:mailto:a@example.com
DTSTART:19970701T200000Z
DTSTAMP:19970611T190000Z
SUMMARY:ST. PAUL SAINTS -VS- DULUTH-SUPERIOR DUKES
UID:0981234-1234234-23@example.com
END:VEVENT
END:VCALENDAR

```

[4.1.2](#) Changing A Published Event

The iCalendar object below describes an update to the event described in 4.1.1, the time has been changed, an end time has been added, and the sequence number has been adjusted.


```
BEGIN:VCALENDAR
METHOD:PUBLISH
VERSION:2.0
PRODID:-//ACME/DesktopCalendar//EN
BEGIN:VEVENT
ORGANIZER:mailto:a@example.com
DTSTAMP:19970612T190000Z
DTSTART:19970701T210000Z
DTEND:19970701T230000Z
SEQUENCE:2
UID:0981234-1234234-23@example.com
SUMMARY:ST. PAUL SAINTS -VS- DULUTH-SUPERIOR DUKES
END:VEVENT
END:VCALENDAR
```

Silverberg/Mansour/Dawson/Hopson 54

Expires September 1998

Internet Draft

iTIP

March 13, 1998

The "UID" property is used by the client to identify the event. The "SEQUENCE" property indicates that this is the second change to the event. Events with sequence numbers 0 and 1 are superseded by this event.

The "SEQUENCE" property provides a reliable way to distinguish different versions of the same event. Each time an event is published, its sequence number is incremented. If a client receives an event with a sequence number 5 and finds it has the same event with sequence number 2, the event SHOULD be updated. However, if the client received an event with sequence number 2 and finds it already has sequence number 5 of the same event, the event MUST NOT be updated.

4.1.3 Canceling A Published Event

The iCalendar object below cancels the event described in 4.1.1. This cancels the event with "SEQUENCE" property of 0, 1, and 2.

```
BEGIN:VCALENDAR
METHOD:CANCEL
VERSION:2.0
PRODID:-//ACME/DesktopCalendar//EN
BEGIN:VEVENT
ORGANIZER:mailto:a@example.com
COMMENT:DUKES forfeit the game
SEQUENCE:2
UID:0981234-1234234-23@example.com
```

DTSTAMP:19970613T190000Z
END:VEVENT
END:VCALENDAR

4.1.4 **A Rich Published Event**

This example describes the same event as in 4.1.1, but in much greater detail.

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:PUBLISH
SCALE:GREGORIAN
SOURCE:http://www.midwaystadium.com/stadium-cal/1997-events.or4
VERSION:2.0
BEGIN:VTIMEZONE
TZID:America-Chicago
TZURL:http://zones.stds_r_us.net/tz/America-Chicago
LAST-MODIFIED:19870101T000000Z
BEGIN:STANDARD
DTSTART:19671029T020000
RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10
TZOFFSETFROM:-0500
TZOFFSETTO:-0600
TZNAME:CST

Silverberg/Mansour/Dawson/Hopson 55

Expires September 1998

Internet Draft

iTIP

March 13, 1998

END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19870405T020000
RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4
TZOFFSETFROM:-0600
TZOFFSETTO:-0500
TZNAME:CDT
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
ORGANIZER:mailto:a@example.com
ATTACH:http://www.midwaystadium.com
CATEGORIES:SPORTS EVENT,ENTERTAINMENT
CLASS:PRIVATE
CREATED:19970415T194319Z
DESCRIPTION:MIDWAY STADIUM\n
Big time game. MUST see.\n
Expected duration:2 hours\n

DTEND;TZID=America-Chicago:19970701T180000
DTSTART;TZID=America-Chicago:19970702T160000
DTSTAMP:19970614T190000Z
STATUS:CONFIRMED
LAST-MODIFIED:19970416T162421Z
LOCATION;VALUE=URL:http://www.midwaystadium.com/
PRIORITY:2
RESOURCES:SCOREBOARD
SEQUENCE:3
SUMMARY:ST. PAUL SAINTS -VS- DULUTH-SUPERIOR DUKES
UID:0981234-1234234-23@example.com
RELATED-TO:0981234-1234234-14@example.com

BEGIN:VALARM
TRIGGER:PT2H
ALARM-TYPE:DISPLAY
DESCRIPTION:It's almost game time
END:VALARM

BEGIN:VALARM
TRIGGER:PT30M
ALARM-TYPE:AUDIO
DESCRIPTION:You SHOULD leave now. Game starts in 30 minutes!
END:VALARM
END:VEVENT
END:VCALENDAR

The "RELATED-TO" field contains the "UID" property of a related calendar event. The "SEQUENCE" property 3 indicates that this event supersedes versions 0, 1, and 2.

4.1.5 Anniversaries or Events attached to entire days

This example demonstrates the use of the "value" parameter to tie a VEVENT to day rather than a specific time.

Silverberg/Mansour/Dawson/Hopson 56

Expires September 1998

Internet Draft

iTIP

March 13, 1998

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:PUBLISH
VERSION:2.0
BEGIN:VEVENT
DTSTAMP:19970614T190000Z
UID:0981234-1234234-23@example.com
DTSTART;VALUE=DATE:19970714

```
RRULE:FREQ=YEARLY;INTERVAL=1
SUMMARY: Bastille Day
END:VEVENT
END:VCALENDAR
```

4.2 Group Event Examples

Group events are distinguished from published events in that they have "Attendees" and that there is interaction between the "Attendees" and the "Organizer" with respect to the event. Individual "A" requests a meeting between individuals "A", "B", "C" and "D". Individual "B" confirms attendance to the meeting. Individual "C" declines attendance. Individual "D" tentatively confirms attendance. The following table illustrates the the message flow between these individuals. A, the CU scheduling the meeting, is referenced as the "Organizer".

Action	"Organizer"	Attendee
Initiate a meeting request	"A" sends a REQUEST message to "B", "C", and "D"	
Accept the meeting request		"B" sends a REPLY message to "A" with its ATTENDEE "attstat" parameter set to "accepted"
Decline the meeting request		"C" sends a REPLY message to "A" with its ATTENDEE "attstat" parameter set to "declined"
Tentatively accept the meeting request		"D" sends a REPLY message to "A" with its ATTENDEE "attstat" parameter set to "tentative"
Confirm meeting status with attendees	"A" sends a REQUEST message to "B" and "C" with updated information.	

4.2.1 **A Group Event Request**

A sample meeting request is sent from "A" to "B", "C", and "D". _E_ is also sent a copy of the request but is not expected to attend and need not reply. "E" illustrates how CUAs might implement a "CC" type feature. Note the use of the "role" parameter. The default value for the "role" parameter is "req-participant" and it need not be enumerated. In this case we are using the value "non-participant" to indicate "E" is a non-attending CU. The parameter is not needed on other "Attendees" since "participant" is the default value.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED;CN=BIG A:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL;CN=B:Mailto:B@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL;CN=C:Mailto:C@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL;CN=Hal:Mailto:D@example.com
ATTENDEE;RSVP=FALSE;TYPE=ROOM:conf_Big@example.com
ATTENDEE;ROLE=NON-PARTICIPANT;RSVP=FALSE:Mailto:E@example.com
DTSTAMP:19970611T190000Z
DTSTART:19970701T200000Z
DTEND:19970701T203000Z
SUMMARY:Phone Conference
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR
```

4.2.2 **Reply To A Group Event Request**

Attendee "B" accepts the meeting.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VEVENT
ATTENDEE;ATTSTAT=ACCEPTED:Mailto:B@example.com
ORGANIZER:MAILTO:A@example.com
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
REQUEST-STATUS:2.0;Success
```

DTSTAMP:19970612T190000Z
END:VEVENT
END:VCALENDAR

Silverberg/Mansour/Dawson/Hopson 58

Expires September 1998

Internet Draft

iTIP

March 13, 1998

"B" could have declined the meeting or indicated tentative acceptance by setting the ATTENDEE "attstat" parameter to "declined" or "tentative", respectively. Also, "REQUEST-STATUS" is not required on a successful transactions.

[4.2.3](#) Update An Event

The event is moved to a different time. The combination of the "UID" property (unchanged) and the "SEQUENCE" (bumped to 1) properties indicate the update.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:B@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:C@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:D@example.com
ATTENDEE;RSVP=FALSE;TYPE=ROOM:Mailto:Conf@example.com
ATTENDEE;ROLE=NON-PARTICIPANT;RSVP=FALSE:Mailto:E@example.com
DTSTART:19970701T180000Z
DTEND:19970701T120000Z
SUMMARY:Phone Conference
UID:www.acme.com-873970198738777@example.com
SEQUENCE:1
DTSTAMP:19970613T190000Z
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR
```

[4.2.4](#) Countering an Event Proposal

"A" sends a "REQUEST" to "B" and "C". "B" makes a counter-proposal to "A" to change the time and location.

"A" sends the following "REQUEST":

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:B@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:C@example.com
```

Silverberg/Mansour/Dawson/Hopson 59

Expires September 1998

Internet Draft

iTIP

March 13, 1998

```
DTSTART:19970701T190000Z
DTEND:19970701T200000Z
SUMMARY:Discuss the Merits of the election results
LOCATION:The Big Conference Room
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
DTSTAMP:19970611T190000Z
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR
```

"B" sends "COUNTER" to "A", requesting changes to time and place. "B" uses the "COMMENT" property to communicate a rationale for the change:

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:COUNTER
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:B@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:C@example.com
DTSTART:19970701T160000Z
DTEND:19970701T190000Z
DTSTAMP:19970612T190000Z
SUMMARY:Discuss the Merits of the election results
LOCATION:The Small Conference Room
COMMENT:This time works much better and I think the big conference
```

room is too big
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
DTSTAMP:19970611T190000Z
END:VEVENT
END:VCALENDAR

"A" accepts the changes from "B". To accept a counter-proposal, the
"Organizer" sends a new Event REQUEST with an incremented sequence
number.

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:B@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:C@example.com
DTSTAMP:19970613T190000Z
DTSTART:19970701T160000Z
DTEND:19970701T190000Z

Silverberg/Mansour/Dawson/Hopson 60 Expires September 1998

Internet Draft iTIP March 13, 1998

SUMMARY:Discuss the Merits of the election results - changed to
meet B's schedule
LOCATION:The Small Conference Room
UID:www.acme.com-873970198738777@example.com
SEQUENCE:1
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

Instead, "A" rejects "B's" counter proposal

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:DECLINECOUNTER
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:B@example.com
COMMENT:Sorry, I cannot change this meeting time
UID:www.acme.com-873970198738777@example.com

SEQUENCE:1
DTSTAMP:19970614T190000Z
END:VEVENT

4.2.5 Delegating an Event

When delegating an event request to another "Calendar User", the "Delegator" must both update the "Organizer" with a "REPLY" and send a request to the "Delegate". There is currently no protocol limitation to delegation depth. It is possible for the original delegate to delegate the meeting to someone else, and so on. When a request is delegated from one CUA to another there are a number of responsibilities required of the "Delegator". They MUST:

- . Send an REPLY to the "Organizer" with their "ATTENDEE" property "attstat" parameter set to "delegated"
- . Include the delegate as an additional "Attendee" with the "delegated-from" property parameter set to the value of the delegator
- . Include the original UID of the "REQUEST" method
- . The "Delegator" MUST also send a copy of the original "REQUEST" method to the "Delegate". The delegator modifies the request as follows:
 - . The "ATTENDEE" property "attstat" parameter for the delegator (sender in this case) is set to "delegated"
 - . "ATTENDEE" "delegated-to" parameter is set to the address of the "Delegate"
 - . An "ATTENDEE" property is added for the "Delegate"

It is not required that the "Delegate" include the "Delegator" in their "REPLY" method. However, it is strongly advised since this will inform the "Delegator" whether the "Delegate" plans to attend the meeting. If

Silverberg/Mansour/Dawson/Hopson 61 Expires September 1998

Internet Draft iTIP March 13, 1998

the "Delegate" declines the meeting, the "Delegator" may elect to delegate the "REQUEST" to another CUA. The process is the same.

+-----+-----+-----+-----+-----+-----+					
Action		"Organizer"		Attendee	
+-----+-----+-----+-----+-----+-----+					
Initiate a meeting		"A" sends a REQUEST			
request		message to "B" and			
		"C"			

Delegate: "C" delegates to "E"		"C" sends a REPLY to "A" with the ATTENDEE. "attstat"parameter set to "delegated" and with a new "ATTENDEE" property for "E". "E's" ATTENDEE "delegated-from" param is set to "C". "C's" ATTENDEE "delegated-to" param is set to "E". "C" sends REQUEST message to "E" with the original meeting request information. The "attstat" property parameter for "C" is set to "delegated" and the "delegated-to" parameter is set to the address of "E". An "ATTENDEE" property is added for "E" and the "delegated-from" parameter is set to the address of "C".
Confirm meeting attendance		"E" sends REPLY message to "A" and optionally "C" with its "attstat" property parameter set to "accepted"
Optional: Redistribute meeting to attendees	"A" sends REQUEST message to "B", "C" and "E". SEQUENCE number is now 1.	

"C" responds to the "Organizer".

BEGIN:VCALENDAR
 PRODID:-//ACME/DesktopCalendar//EN
 METHOD:REPLY

Silverberg/Mansour/Dawson/Hopson 62

Expires September 1998

```
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:MAILTO:A@Example.com
ATTENDEE;ATTSTAT=DELEGATED;DELEGATED-
TO=Mailto:E@example.com:Mailto:C@example.com
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
REQUEST-STATUS:2.0;Success
DTSTAMP:19970611T190000Z
END:VEVENT
END:VCALENDAR
```

Attendee "C" delegates presence at the meeting to "E".

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ATTSTAT=DELEGATED;DELEGATED-
TO=Mailto:E@example.com:Mailto:C@example.com
ATTENDEE;ROLE=DELEGATE;RSVP=TRUE;
  DELEGATED-FROM=Mailto:C@example.com:Mailto:E@example.com
DTSTART:19970701T180000Z
DTEND:19970701T120000Z
SUMMARY:Phone Conference
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
STATUS:CONFIRMED
DTSTAMP:19970611T190000Z
END:VEVENT
END:VCALENDAR
```

4.2.6 Delegate Accepts the Meeting

To accept a delegated meeting, the delegate, "E", sends the following message to "A" and "C"

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:MAILTO:A@Example.com
ATTENDEE;ATTSTAT=CONFIRMED;DELEGATED-
FROM=Mailto:C@example.com:Mailto:E@example.com
ATTENDEE;ATTSTAT=DELEGATED;DELEGATED-
```

TO=Mailto:E@example.com:Mailto:C@example.com
UID:www.acme.com-873970198738777@example.com
SEQUENCE:1
REQUEST-STATUS:2.0;Success

Silverberg/Mansour/Dawson/Hopson 63

Expires September 1998

Internet Draft

iTIP

March 13, 1998

DTSTAMP:19970614T190000Z
END:VEVENT
END:VCALENDAR

[4.2.7](#) Delegate Declines the Meeting

In this example the "Delegate" declines the meeting request and sets the "ATTENDEE" property "attstat" parameter to DECLINED. The "Organizer" SHOULD resend the "REQUEST" to "C" with the "attstat" parameter of the "Delegate" set to "declined". This lets the "Delegator" know that the "Delegate" has declined and provides an opportunity to the "Delegator" to either accept the request or delegate it to another CU.

Response from "E" to "A" and "C".

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:MAILTO:A@Example.com
ATTENDEE;ATTSTAT=DECLINED;DELEGATED-
FROM=Mailto:C@example.com:Mailto:E@example.com
UID:www.acme.com-873970198738777@example.com
ATTENDEE;ATTSTAT=DELEGATED;DELEGATED-
TO=Mailto:E@example.com:Mailto:C@example.com
SEQUENCE:1
REQUEST-STATUS:2.0;Success
DTSTAMP:19970614T190000Z
END:VEVENT
END:VCALENDAR

"A" resends the "REQUEST" method to "C". "A" may also wish to express the fact that the item was delegated in the "COMMENT" property.

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN

```

METHOD:REPLY
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:MAILTO:A@Example.com
ATTENDEE;ATTSTAT=DECLINED;DELEGATED-
FROM=Mailto:C@example.com:Mailto:E@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:C@example.com
UID:www.acme.com-873970198738777@example.com
SEQUENCE:2
REQUEST-STATUS:2.0;Success
DTSTAMP:19970614T200000Z
COMMENT:DELEGATE (ATTENDEE Mailto:E@example.com) DECLINED YOUR
INVITATION
END:VEVENT

```

Silverberg/Mansour/Dawson/Hopson 64 Expires September 1998

Internet Draft iTIP March 13, 1998

END:VCALENDAR

[4.2.8](#) Forwarding an Event Request

The protocol does not prevent an "Attendee" from "forwarding" an "VEVENT" calendar component to other "Calendar Users". Forwarding differs from delegation in that the forwarded "Calendar User" (often referred to as a "Party Crasher") does not replace the forwarding "Calendar User". Implementations are not required to add the "Party Crasher" to the "Attendee" list and hence there is no guarantee that a "Party Crasher" will receive additional updates to the Event. The forwarding "Calendar User" SHOULD NOT add the "Party Crasher" to the attendee list. The "Organizer" MAY add the forwarded "Calendar User" to the attendee list.

[4.2.9](#) Cancel A Group Event

Individual "A" requests a meeting between individuals "A", "B", "C", and "D". Individual "B" declines attendance to the meeting. Individual "A" decides to cancel the meeting. The following table illustrates the sequence of messages that would be exchanged between these individuals.

Messages related to a previously canceled event ("SEQUENCE" property value is less than the "SEQUENCE" property value of the "CANCEL" message) MUST be ignored.

+-----+

Action	"Organizer"	"Attendee"
Initiate a meeting request	"A" sends a REQUEST message to "B", "C", and "D"	
Decline the meeting request		"B" sends a "REPLY" message to "A" with its "attstat" parameter set to "declined".
Cancel the meeting	"A" sends a CANCEL message to "B", "C" and "D"	

The example shows how "A" cancels the event.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:CANCEL
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
```

Silverberg/Mansour/Dawson/Hopson 65

Expires September 1998

Internet Draft

iTIP

March 13, 1998

```
ATTENDEE;TYPE=INDIVIDUAL;Mailto:A@example.com
ATTENDEE;TYPE=INDIVIDUAL;Mailto:B@example.com
ATTENDEE;TYPE=INDIVIDUAL;Mailto:C@example.com
ATTENDEE;TYPE=INDIVIDUAL;Mailto:D@example.com
COMMENT:Mr. B cannot attend. It's raining. Lets cancel.
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
STATUS:CANCELLED
DTSTAMP:19970613T190000Z
END:VEVENT
END:VCALENDAR
```

[4.2.10](#) Removing Attendees

"A" wants to remove "B" from a meeting. This is done by sending a "CANCEL" to "B" and removing "B" from the attendee list in the master copy of the event.

+-----+

Action	"Organizer"	"Attendee"
Remove an "B" as an "Attendee"	"A" sends a CANCEL message to "B"	
Update the master copy of the event	"A" sends the updated event to the remaining "Attendees"	

The original meeting includes "A", "B", "C", and "D". The example below shows the "CANCEL" that "A" sends to "B". Note that in the example below the "STATUS" property is omitted. This is used when the meeting itself is cancelled and not when the intent is to remove an "Attendee" from the Event.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:CANCEL
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
COMMENT:You're off the hook for this meeting
UID:www.acme.com-873970198738777@example.com
DTSTAMP:19970613T193000Z
END:VEVENT
END:VCALENDAR
```

The updated master copy of the event is shown below. The "Organizer" MAY resend the updated event to the remaining "Attendees". Note that "B" has been removed.

```
BEGIN:VCALENDAR
```

Silverberg/Mansour/Dawson/Hopson 66

Expires September 1998

Internet Draft

iTIP

March 13, 1998

```
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;TYPE=INDIVIDUAL:Mailto:C@example.com
ATTENDEE;TYPE=INDIVIDUAL:Mailto:D@example.com
ATTENDEE;TYPE=ROOM:CR_Big@example.com
ATTENDEE;ROLE=NON-PARTICIPANT;
```

RSVP=FALSE:Mailto:E@example.com
DTSTAMP:19970611T190000Z
DTSTART:19970701T200000Z
DTEND:19970701T203000Z
SUMMARY:Phone Conference
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

4.2.11 **Replacing the Organizer**

The scenario for this example begins with "A" as the "Organizer" for a recurring meeting with "B", "C", and "D". "A" receives a new job offer in another country and leaves without changing the "Organizer" to this meeting. "A" left no forwarding address or way to be reached. Using out-of-band communication, the other "Attendees" eventually learn what has happened and reach an agreement that "B" should become the new "Organizer" for the meeting. To do this, "B" sends out a new version of the event and the other "Attendees" agree to accept "B" as the new "Organizer". "B" also removes "A" from the event

This is the message "B" sends to "C" and "D"

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:B@example.com
ATTENDEE;ROLE=CHAIR;STATUS=ACCEPTED:Mailto:B@example.com
ATTENDEE;TYPE=INDIVIDUAL:Mailto:C@example.com
ATTENDEE;TYPE=INDIVIDUAL:Mailto:D@example.com
DTSTAMP:19970611T190000Z
DTSTART:19970701T200000Z
DTEND:19970701T203000Z
RRULE:FREQ=WEEKLY
SUMMARY:Phone Conference
UID:123456@example.com
SEQUENCE:1
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

4.3 Busy Time Examples

Busy time objects can be used in several ways. First, a CU may Request busy time from another CU for a specific range of time. That request can be answered with a busy time Reply. Additionally, a CU may simply publish busy their busy time for a given interval and point other CUs to the published location. The following examples outline both scenarios.

Individual "A" publishes Busy time for one week.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//ENVERSION:2.0
METHOD:PUBLISH
BEGIN:VFREEBUSY
DTSTAMP:19980101T124100Z
ATTENDEE:MAILTO:A@Example.com
DTSTART:19980101T124200Z
DTEND:19980107T124200Z
FREEBUSY:19980101T180000Z/19980101T190000Z
FREEBUSY:19980103T020000Z/19980103T050000Z
FREEBUSY:19980107T020000Z/19980107T050000Z
FREEBUSY:19980113T000000Z/19980113T010000Z
FREEBUSY:19980115T190000Z/19980115T200000Z
FREEBUSY:19980115T220000Z/19980115T230000Z
FREEBUSY:19980116T013000Z/19980116T043000Z
END:VFREEBUSY
END:VCALENDAR
```

Individual "A" requests busy time from individuals "B", "C". Individual "B" and "C" replies with busy time data to individual "A". The following table illustrates the sequence of messages that would be exchanged between these individuals.

Action	"Organizer"	Attendee
Initiate a busy time request	"A" sends "REQUEST" message to "B" and "C"	
Reply to the "BUSY" request with "BUSY" time data		"B" sends a "REPLY" message to "A" with busy time data

4.3.1 Request Busy Time

"A" sends a "BUSY-REQUEST" to "B" and "C" for busy time

Silverberg/Mansour/Dawson/Hopson 68

Expires September 1998

Internet Draft

iTIP

March 13, 1998

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VFREEBUSY
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR:Mailto:A@example.com
ATTENDEE:Mailto:B@example.com
ATTENDEE:Mailto:C@example.com
DTSTAMP:19970613T190000Z
DTSTART:19970701T080000Z
DTEND:19970701T200000
UID:www.acme.com-873970198738777@example.com
END:VFREEBUSY
END:VCALENDAR
```

4.3.2 Reply To A Busy Time Request

"B" sends a "REPLY" method type of a "VFREEBUSY" calendar component to "A"

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VFREEBUSY
ORGANIZER:MAILTO:A@example.com
ATTENDEE:Mailto:B@example.com
DTSTART:19970701T080000Z
DTEND:19970701T200000Z
UID:www.acme.com-873970198738777@example.com
FREEBUSY:19970701T090000Z/PT1H,19970701T140000Z/PT30H
DTSTAMP:19970613T190030Z
END:VFREEBUSY
END:VCALENDAR
```

"B" is busy from 09:00 to 10:00 and from 14:00 to 14:30.

4.4 Recurring Event and Time Zone Examples

4.4.1 A Recurring Event Spanning Time Zones

This event describes a weekly phone conference. The "Attendees" are each in a different time zone.

BEGIN:VCALENDAR

PRODID:-//ACME/DesktopCalendar//EN

Silverberg/Mansour/Dawson/Hopson 69

Expires September 1998

Internet Draft

iTIP

March 13, 1998

METHOD:REQUEST

VERSION:2.0

BEGIN:VTIMEZONE

TZID:America-SanJose

TZURL:http://zones.stds_r_us.net/tz/America-SanJose

LAST-MODIFIED:19870101T000000Z

BEGIN:STANDARD

DTSTART:19671029T020000

RRULE:FREQ=YEARLY;BYDAY=-1SU;BYMONTH=10

TZOFFSETFROM:-0700

TZOFFSETTO:-0800

TZNAME:PST

END:STANDARD

BEGIN:DAYLIGHT

DTSTART:19870405T020000

RRULE:FREQ=YEARLY;BYDAY=1SU;BYMONTH=4

TZOFFSETFROM:-0800

TZOFFSETTO:-0700

TZNAME:PDT

END:DAYLIGHT

END:VTIMEZONE

BEGIN:VEVENT

ORGANIZER:Mailto:A@example.com

ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED;TYPE=INDIVIDUAL:A@example.COM

ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:B@example.fr

ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:c@example.jp

DTSTAMP:19970613T190030Z

DTSTART;TZID=America-SanJose:19970701T140000

DTEND;TZID=America-SanJose:19970701T150000

RRULE:FREQ=WEEKLY;INTERVAL=20;WKST=SU;BYDAY=TU

RDATE;TZID=America-SanJose:19970910T140000
EXDATE;TZID=America-SanJose:19970909T140000
EXDATE;TZID=America-SanJose:19971028T140000
SUMMARY:Weekly Phone Conference
UID:www.acme.com-873970198738777@example.com
SEQUENCE:0
STATUS:CONFIRMED
END:VEVENT

END:VCALENDAR

The first two components of this iCalendar object are the time zone components. The "DTSTART" date coincides with the first instance of the RRULE property.

The recurring meeting is defined in a particular time zone, presumably that of the originator. The client for each "Attendee" has the responsibility of determining the recurrence time in the "Attendee's" time zone.

The repeating event starts on Tuesday, July 1, 1997 at 2:00pm PDT.
"Attendee" B@example.fr is in France where the local time on this date

Silverberg/Mansour/Dawson/Hopson 70 Expires September 1998

Internet Draft iTIP March 13, 1998

is 7 hours later than PDT or 21:00. "Attendee" C@example.jp is in Japan where local time is 9 hours ahead of than UTC or Wednesday, July 2 at 07:00. The event repeats weekly on Tuesdays (in PST/PDT). The "RRULE" property results in 20 instances. The last instance falls on Tuesday, November 11, 1997 2:00pm PST. The "RDATE" property adds another instance: WED, 10-SEP-1997 2:00 PM PST.

There are two exceptions to this recurring appointment. The first one is:

TUE, 09-SEP-1997 21:00 GMT
TUE, 09-SEP-1997 14:00 PDT
WED, 10-SEP-1997 07:00 JDT

and the second is:

TUE, 28-OCT-1997 22:00 GMT
TUE, 28-OCT-1997 14:00 PST
WED, 29-OCT-1997 07:00 JST

[4.4.2](#) Modify A Recurring Instance

In this example the "Organizer" issues a recurring meeting. Later the "Organizer" changes an instance of the event by changing the "DTSTART" property. Note the use of "RECURRENCE-ID" property and "SEQUENCE" property in the second request.

Original Request:

```
BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
CREATED:19970526T083000Z
UID:guid-1@host1.com
SEQUENCE:0
RRULE:FREQ=MONTHLY;BYMONTHDAY=1;UNTIL=19980901T210000Z
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE:Mailto:B@example.com
ATTENDEE:Mailto:C@example.com
ATTENDEE:Mailto:D@example.com
DESCRIPTION:IETF-C&S Conference Call
CLASS:PUBLIC
SUMMARY:IETF Calendaring Working Group Meeting
DTSTART:19970601T210000Z
DTEND:19970601T220000Z
LOCATION:Conference Call
DTSTAMP:19970526T083000Z
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR
```

Silverberg/Mansour/Dawson/Hopson 71

Expires September 1998

Internet Draft

iTIP

March 13, 1998

The event request below is to change the time of a specific instance. This changes the July 1st instance to July 3rd.

```
BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
CREATED:19970526T083000Z
UID:guid-1@host1.com
RECURRENCE-ID:19970701T210000Z
```

SEQUENCE:1
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE:Mailto:B@example.com
ATTENDEE:Mailto:C@example.com
ATTENDEE:Mailto:D@example.com
DESCRIPTION:IETF-C&S Conference Call
CLASS:PUBLIC
SUMMARY:IETF Calendaring Working Group Meeting
DTSTART:19970703T210000Z
DTEND:19970703T220000Z
LOCATION:Conference Call
DTSTAMP:19970626T093000Z
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

[4.4.3](#) **Cancel an Instance**

In this example the "Organizer" of a recurring event deletes the August 1st instance.

BEGIN:VCALENDAR
METHOD:CANCEL
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:guid-1@host1.com
ORGANIZER:Mailto:A@example.com
RECURRENCE-ID:19970801T210000Z
SEQUENCE:2
STATUS:CANCELLED
DTSTAMP:19970721T093000Z
END:VEVENT
END:VCALENDAR

Silverberg/Mansour/Dawson/Hopson 72

Expires September 1998

Internet Draft

iTIP

March 13, 1998

[4.4.4](#) **Cancel Recurring Event**

In this example the "Organizer" wishes to cancel the entire recurring event and any exceptions.

```
BEGIN:VCALENDAR
METHOD:CANCEL
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:guid-1@host1.com
ORGANIZER:Mailto:A@example.com
DTSTAMP:19970721T103000Z
SEQUENCE:2
END:VEVENT
END:VCALENDAR
```

4.4.5 Change All Future Instances

This example changes the meeting location from a conference call to Seattle starting September 1 and extends to all future instances.

```
BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
CREATED:19970526T083000Z
UID:guid-1@host1.com
RECURRENCE-ID;THISANDFUTURE:19970901T210000Z
SEQUENCE:3
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
ATTENDEE;RSVP=TRUE:Mailto:C@example.com
ATTENDEE;RSVP=TRUE:Mailto:D@example.com
DESCRIPTION:IETF-C&S Discussion
CLASS:PUBLIC
SUMMARY:IETF Calendaring Working Group Meeting
DTSTART:19970901T210000Z
DTEND:19970901T220000Z
LOCATION:Building 32, Microsoft, Seattle, WA
DTSTAMP:19970526T083000Z
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR
```

4.4.6 Add A New Instance To A Recurring Event

This example adds a one-time additional instance to the recurring event.

"Organizer" adds a second July meeting on the 15th.

Silverberg/Mansour/Dawson/Hopson 73

Expires September 1998

Internet Draft

iTIP

March 13, 1998

```
BEGIN:VCALENDAR
METHOD:ADD
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
CREATED:19970526T083000Z
UID:123456789@host1.com
SEQUENCE:4
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
ATTENDEE;RSVP=TRUE:Mailto:C@example.com
ATTENDEE;RSVP=TRUE:Mailto:D@example.com
DESCRIPTION:IETF-C&S Conference Call
CLASS:PUBLIC
SUMMARY:IETF Calendaring Working Group Meeting
DTSTART:19970715T210000Z
DTEND:19970715T220000Z
LOCATION:Conference Call
DTSTAMP:19970629T093000Z
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR
```

[4.4.7](#) Add A New Series of Instances To A Recurring Event

The scenario for this example involves an ongoing meeting, originally set up to occur every Tuesday. The "Organizer" later decides that the meetings need to be on Tuesdays and Thursdays, but does not want to reschedule the entire meeting or lose any of the per-instance information already collected.

The original event:

```
BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:123456789@host1.com
SEQUENCE:0
RRULE:WKST=SU;BYDAY=TU;FREQ=WEEKLY
ORGANIZER:Mailto:A@example.com
```


ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
SUMMARY:Review Accounts
DTSTART:19980303T210000Z
DTEND:19980303T220000Z
LOCATION:The White Room
DTSTAMP:19980301T093000Z
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

Silverberg/Mansour/Dawson/Hopson 74

Expires September 1998

Internet Draft

iTIP

March 13, 1998

The "Organizer" adds Thursdays to the event:

BEGIN:VCALENDAR
METHOD:ADD
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:123456789@host1.com
SEQUENCE:7
RRULE:WKST=SU;BYDAY=TH;FREQ=WEEKLY
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
SUMMARY:Review Accounts
DTSTART:19980303T210000Z
DTEND:19980303T220000Z
DTSTAMP:19980303T193000Z
LOCATION:The Usual conference room
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

Alternatively, if the "Organizer" is not concerned with per-instance updates, the entire event can be rescheduled using a "REQUEST". This is done by using the "UID" of the event to reschedule and including the modified RRULE.

BEGIN:VCALENDAR
METHOD:ADD
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT

UID:123456789@host1.com
SEQUENCE:7
RRULE:WKST=SU;BYDAY=TU,TH;FREQ=WEEKLY
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
SUMMARY:Review Accounts
DTSTART:19980303T210000Z
DTEND:19980303T220000Z
DTSTAMP:19980303T193000Z
LOCATION:The White Room
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

The next series of examples illustrate how an "Organizer" would respond to a "REFRESH" submitted by an "Attendee" after a series of instance-specific modifications. To convey all instance-specific changes, the "Organizer" must provide the latest event description and the relevant instances. The first three examples show the history including the

Silverberg/Mansour/Dawson/Hopson 75

Expires September 1998

Internet Draft

iTIP

March 13, 1998

initial "VEVENT" request and subsequent instance changes and finally the "REFRESH".

Original Request:

BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:123456789@host1.com
SEQUENCE:0
RDATE:19980304T180000Z
RDATE:19980311T180000Z
RDATE:19980318T180000Z
RDATE:19980325T180000Z
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
SUMMARY:Review Accounts
DTSTART:19980304T180000Z
DTEND:19980304T200000Z
DTSTAMP:19980303T193000Z

LOCATION:Conference Room A
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

Organizer changes 2nd instance Location and Time:

BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:123456789@host1.com
SEQUENCE:1
RECURRENCE-ID:19980311T180000Z
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
SUMMARY:Review Accounts
DTSTART:19980311T160000Z
DTEND:19980304T180000Z
DTSTAMP:19980306T193000Z
LOCATION:The Small conference room
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

Organizer adds a 4th instance of the meeting using the "ADD" method

BEGIN:VCALENDAR
METHOD:ADD

Silverberg/Mansour/Dawson/Hopson 76

Expires September 1998

Internet Draft

iTIP

March 13, 1998

PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:123456789@host1.com
SEQUENCE:2
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
SUMMARY:Review Accounts
DTSTART:19980315T180000Z
DTEND:19980315T200000Z
DTSTAMP:19980307T193000Z
LOCATION:Conference Room A

STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

If "B" requests a "REFRESH", "A" responds with the following to capture all instance-specific data. In this case both the initial request and an additional "VEVENT" that specifies the instance-specific data are included. Because these are both of the same type (they are both "VEVENTS"), they can be conveyed in the same iCalendar object.

BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:123456789@host1.com
SEQUENCE:3
RDATE:19980304T180000Z
RDATE:19980318T180000Z
RDATE:19980315T180000Z
Error! Bookmark not defined.
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
SUMMARY:Review Accounts
DTSTART:19980304T180000Z
DTEND:19980304T200000Z
DTSTAMP:19980303T193000Z
LOCATION:Conference Room A
STATUS:CONFIRMED
END:VEVENT

BEGIN:VEVENT
Error! Bookmark not defined.
SEQUENCE:3
RECURRENCE-ID:19980311T160000Z
Error! Bookmark not defined.
ATTENDEE;ROLE=CHAIR;Error! Bookmark not defined.
ATTENDEE;Error! Bookmark not defined.
SUMMARY:Review Accounts
DTSTART:19980311T160000Z

Silverberg/Mansour/Dawson/Hopson 77

Expires September 1998

Internet Draft

iTIP

March 13, 1998

DTEND:19980304T180000Z
DTSTAMP:19980306T193000Z

LOCATION:The Small conference room
STATUS:CONFIRMED
END:VEVENT

END:VCALENDAR

4.4.8 Counter An Instance Of A Recurring Event

In this example one of the "Attendees" counters the "DTSTART" property of the proposed second July meeting.

BEGIN:VCALENDAR
METHOD:COUNTER
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
CREATED:19970526T083000Z
UID:guid-1@host1.com
RECURRENCE-ID:19970715T210000Z
SEQUENCE:4
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;RSVP=TRUE:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
ATTENDEE;RSVP=TRUE:Mailto:C@example.com
ATTENDEE;RSVP=TRUE:Mailto:D@example.com
DESCRIPTION:IETF-C&S Conference Call
CLASS:PUBLIC
SUMMARY:IETF Calendaring Working Group Meeting
DTSTART:19970715T220000Z
DTEND:19970715T230000Z
LOCATION:Conference Call
COMMENT:May we bump this by an hour? I have a conflict
DTSTAMP:19970629T094000Z
END:VEVENT
END:VCALENDAR

4.4.9 Error Reply To A Request

The following example illustrates a scenario where a meeting is proposed containing an unsupported property and a bad property.

Original Request

BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
CREATED:19970526T083000Z

Silverberg/Mansour/Dawson/Hopson 78

Expires September 1998

Internet Draft

iTIP

March 13, 1998

UID:guid-1@host1.com
SEQUENCE:0
RRULE:FREQ=MONTHLY;BYMONTHDAY=1
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:Mailto:B@example.com
ATTENDEE;RSVP=TRUE:Mailto:C@example.com
ATTENDEE;RSVP=TRUE:Mailto:D@example.com
DESCRIPTION:IETF-C&S Conference Call
CLASS:PUBLIC
SUMMARY:IETF Calendaring Working Group Meeting
DTSTART:19970601T210000Z
DTEND:19970601T220000Z
DTSTAMP:19970602T094000Z
LOCATION:Conference Call
STATUS:CONFIRMED
FOO:BAR
END:VEVENT
END:VCALENDAR

Response from "B" to indicate that RRULE is not supported and an unrecognized property was encountered

BEGIN:VCALENDAR
PRODID:-//RDU Software//NONSGML HandCal//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE:Mailto:B@example.com
REQUEST-STATUS:2.8;Repeating event ignored. Scheduled as a single
event;RRULE
REQUEST-STATUS:3.0;Invalid Property Name;FOO
UID:guid-1@host1.com
SEQUENCE:0
DTSTAMP:19970603T094000Z
END:VEVENT
END:VCALENDAR

4.5 Group To-do Examples

Individual "A" creates a group task in which individuals "A", "B", "C"

and "D" will participate. Individual "B" confirms acceptance of the task. Individual "C" declines the task. Individual "D" tentatively accepts the task. The following table illustrates the sequence of messages that would be exchanged between these individuals. Individual "A" then issues a "REQUEST" method to obtain the status of the to-do from each participant. The response indicates the individual "Attendee's" completion status. The table below illustrates the message flow.

Silverberg/Mansour/Dawson/Hopson 79

Expires September 1998

Internet Draft

iTIP

March 13, 1998

Action	"Organizer"	Attendee
Initiate a to-do request	"A" sends a REQUEST message to "B", "C", and "D"	
Accept the to-do request		"B" sends a "REPLY" message to "A" with its "attstat" parameter set to "accepted".
Decline the to-do request		"C" sends a REPLY message to "A" with its "attstat" parameter set to "declined".
Tentatively accept the to-do request		"D" sends a REPLY message to "A" with its "attstat" parameter set to "tentative".
Check attendee completion status	"A" sends a REQUEST message to "B" and "C" with current information.	
Attendee indicates percent complete		"B" sends a "REPLY" message indicating percent complete
Attendee indicates completion		"C" sends a "REPLY" message indicating

| | | completion |
+-----+

[4.5.1](#) A VTODO Request

A sample "REQUEST" with for a "VTODO" calendar component that "A" sends to "B", "C", and "D".

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VTODO
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR:Mailto:A@example.com
ATTENDEE;RSVP=TRUE:B@example.com
ATTENDEE;RSVP=TRUE:Mailto:C@example.com
ATTENDEE;RSVP=TRUE:Mailto:D@example.com
DTSTART:19970701T170000Z
```

Silverberg/Mansour/Dawson/Hopson 80

Expires September 1998

Internet Draft

iTIP

March 13, 1998

```
DUE:19970722T170000Z
SUMMARY:Create the requirements document
UID:www.acme.com-873970198738777-00@example.com
SEQUENCE:0
DTSTAMP:19970717T200000Z
STATUS:Needs Action
END:VTODO
END:VCALENDAR
```

[4.5.2](#) A VTODO Reply

Attendee "B" accepts the meeting.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VTODO
ORGANIZER:Mailto:A@example.com
ATTENDEE;ATTSTAT=ACCEPTED:Mailto:B@example.com
UID:www.acme.com-873970198738777-00@example.com
COMMENT:I'll send you my input by e-mail
```


SEQUENCE:0
DTSTAMP:19970717T203000Z
REQUEST-STATUS:2.0;Success
END:VTODO
END:VCALENDAR

"B" could have declined the TODO or indicated tentative acceptance by setting the "attstat" property parameter sequence to "declined" or "tentative", respectively.

[4.5.3](#) A VTOD Request for Updated Status

"A" requests status from all "Attendees".

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VTODO
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:B@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:C@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:D@example.com
UID:www.acme.com-873970198738777-00@example.com
SEQUENCE:0
STATUS:IN-PROCESS
DTSTAMP:19970717T230000Z
END:VTODO

Silverberg/Mansour/Dawson/Hopson 81

Expires September 1998

Internet Draft

iTIP

March 13, 1998

END:VCALENDAR

[4.5.4](#) A Reply: Percent-Complete

A reply indicating the task being worked on and that "B" is 75% complete with "B's" part of the assignment.

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VTODO

ORGANIZER:MAILTO:A@example.com
ATTENDEE;ATTSTAT=IN-PROCESS:Mailto:B@example.com
PERCENT-COMPLETE:75
UID:www.acme.com-873970198738777-00@example.com
DTSTAMP:19970717T233000Z
SEQUENCE:0
END:VTODO
END:VCALENDAR

4.5.5 A Reply: Completed

A reply indicating that "C" completed "C's" part of the assignment.

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VTODO
ORGANIZER:MAILTO:A@example.com
ATTENDEE;ATTSTAT=COMPLETED:Mailto:C@example.com
UID:www.acme.com-873970198738777-00@example.com
DTSTAMP:19970717T233000Z
SEQUENCE:0
END:VTODO
END:VCALENDAR

4.5.6 An Updated VTODO Request

Organizer "A" resends the "VTODO" calendar component. "A" sets the overall completion for the to-do at 40%.

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VTODO
ORGANIZER:Mailto:A@example.com

Silverberg/Mansour/Dawson/Hopson 82

Expires September 1998

Internet Draft

iTIP

March 13, 1998

ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE;ATTSTAT=ACCEPTED;TYPE=INDIVIDUAL:Mailto:B@example.com
ATTENDEE;ATTSTAT=COMPLETED;TYPE=INDIVIDUAL:Mailto:C@example.com

ATTENDEE;ATTSTAT=IN-PROCESS;TYPE=INDIVIDUAL:Mailto:D@example.com
DTSTART:19970701T100000-0700
DUE:19970722T100000-0700
SUMMARY:Create the requirements document
UID:www.acme.com-873970198738777-00@example.com
SEQUENCE:1
DTSTAMP:19970718T100000Z
STATUS:IN-PROGRESS
PERCENT-COMPLETE:40
END:VTODO
END:VCALENDAR

[4.5.7](#) **Recurring VTODOs**

The following examples relate to recurring "VTODO" calendar components.

[4.5.7.1](#) **Request for a Recurring VTODO**

In this example "A" sends a recurring "VTODO" calendar component to "B" and "C".

BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REQUEST
VERSION:2.0
BEGIN:VTODO
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR:Mailto:A@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:B@example.com
ATTENDEE;RSVP=TRUE;TYPE=INDIVIDUAL:Mailto:C@example.com
RRULE:FREQ=MONTHLY;COUNT=10;BYDAY=1FR
DTSTART:19980101T100000-0700
DUE:19980103T100000-0700
SUMMARY:Send Status Reports to Area Managers
UID:www.acme.com-873970198738777-00@example.com
SEQUENCE:0
DTSTAMP:19970717T200000Z
STATUS:NEEDS ACTION
END:VTODO
END:VCALENDAR

[4.5.7.2](#) **Calculating due dates in recurring VTODOs**

The due date in a recurring "VTODO" calendar component is either a fixed interval specified in the "REQUEST" method or specified using the "RECURRENCE-ID" property. The former is calculated by applying the difference between "DTSTART" and "DUE" properties and applying it to

Internet Draft

iTIP

March 13, 1998

each of the start of each recurring instance. Hence, if the initial "VTOD0" calendar component specifies a "DTSTART" property value of "19970701T190000Z" and a "DUE" property value of "19970801T190000Z" the interval of one day which is applied to each recurring instance of the "VTOD0" calendar component to determine the "DUE" date of the instance.

[4.5.7.3](#) Replying to an instance of a recurring VTOD0

In this example "B" updates "A" on a single instance of the "VTOD0" calendar component.

```
BEGIN:VCALENDAR
PRODID:-//ACME/DesktopCalendar//EN
METHOD:REPLY
VERSION:2.0
BEGIN:VTOD0
ATTENDEE;ATTSTAT=IN-PROCESS:Mailto:B@example.com
PERCENT-COMPLETE:75
UID:www.acme.com-873970198738777-00@example.com
DTSTAMP:19970717T233000Z
RECURRENCE-ID:19980101T170000Z
SEQUENCE:1
END:VTOD0
END:VCALENDAR
```

[4.6](#) Journal Examples

The iCalendar object below describes a single journal entry for October 2, 1997. The "RELATED-TO" property references the phone conference event for which minutes were taken.

```
BEGIN:VCALENDAR
PROFILE:PUBLISH
PRODID:-//ACME/DesktopCalendar//EN
VERSION:2.0
BEGIN:VJOURNAL
DTSTART:19971002T200000Z
SUMMARY:Phone conference minutes
DESCRIPTION:The editors meeting was held on October 1, 1997.
  Details are in the attached document.
UID:0981234-1234234-2410@example.com
```

RELATED-TO:0981234-1234234-2402-35@example.com
ATTACH:ftp\://ftp.example.com/pub/ed/minutes100197.txt
END:VJOURNAL
END:VCALENDAR

Silverberg/Mansour/Dawson/Hopson 84

Expires September 1998

Internet Draft

iTIP

March 13, 1998

[4.7](#) Other Examples

[4.7.1](#) Event Refresh

Refresh the event with "UID" property value of "guid-1-12345@host1.com":

```
BEGIN:VCALENDAR
PRODID:-//RDU Software//NONSGML HandCal//EN
METHOD:REFRESH
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE:Mailto:B@example.com
ATTENDEE:Mailto:C@example.com
ATTENDEE:Mailto:D@example.com
UID: guid-1-12345@host1.com
DTSTAMP:19970603T094000
END:VEVENT
END:VCALENDAR
```

[4.7.2](#) Bad RECURRENCE-ID

If an "Attendee" receives a request that references a "RECURRENCE-ID" property that cannot be found, the "Attendee" SHOULD send a "REFRESH" method back to the "Organizer" for the latest copy of the event.

+-----+			
Action	"Organizer"	Attendee	
+-----+			
Update an instance	"A" sends "REQUEST"		
request	message to "B"		

Attendee requests		"B" sends a "REFRESH"	
refresh because		message to "A"	
"RECURRENCE-ID" was			
not found			
Refresh the entire	"A" sends the		
Event	latest copy of the		
	Event to "B"		
Attendee handles		"B" updates to the	
the request and		latest copy of the	
updates the		meeting.	
instance			

Request from "A":

Silverberg/Mansour/Dawson/Hopson 85

Expires September 1998

Internet Draft

iTIP

March 13, 1998

```

BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:-//RDU Software//NONSGML HandCal//EN
VERSION:2.0
BEGIN:VEVENT
UID:acme-12345@host1.com
SEQUENCE:3
RRULE:FREQ=WEEKLY
RDATE;VALUE=PERIOD:19970819T210000Z/199700819T220000Z
ORGANIZER:Mailto:A@example.com
ATTENDEE;ROLE=CHAIR;ATTSTAT=ACCEPTED:Mailto:A@example.com
ATTENDEE:Mailto:B@example.com
DESCRIPTION:IETF-C&S Conference Call
SUMMARY:IETF Calendaring Working Group Meeting
DTSTART:19970801T210000Z
DTEND:19970801T220000Z
DTSTAMP:19970726T083000
STATUS:CONFIRMED
END:VEVENT
END:VCALENDAR

```

"B" has the event with "UID" property "acme-12345@host1.com" but the "SEQUENCE" property value is "1" and the event does not have an instance at the specified recurrence time. This means that "B" has missed one update and needs a new copy of the event.

```

BEGIN:VCALENDAR
PRODID:-//RDU Software//NONSGML HandCal//EN
METHOD:REFRESH
VERSION:2.0
BEGIN:VEVENT
ORGANIZER:Mailto:A@example.com
ATTENDEE;ATTSTAT=ACCEPTED:Mailto:B@example.com
UID:acme-12345@host1.com
DTSTAMP:19970603T094000
END:VEVENT
END:VCALENDAR

```

5 Application Protocol Fallbacks

5.1 Partial Implementation

Applications that support this memo are not required to support the entire protocol. The following describes how methods and properties SHOULD "fallback" in applications that do not support the complete protocol. If a method or property is not addressed in this section, it may be ignored.

Silverberg/Mansour/Dawson/Hopson 86

Expires September 1998

Internet Draft

iTIP

March 13, 1998

5.1.1 Event-Related Fallbacks

Method	Fallback
-----	-----
PUBLISH	Required.
CANCEL	Required.
REQUEST	PUBLISH
REPLY	Required.
ADD	Required.
DELEGATE	Reply with Not Supported.
REQUEST	Reply with Not Supported.
REPLY	Reply with Not Supported.
COUNTER	Reply with Not Supported
DECLINECOUNTER	Required if EVENT-COUNTER is implemented; otherwise

reply with Not Supported.

iCalendar Property	Fallback
-----	-----
CALSCALE	Ignore; assume GREGORIAN.
GEO	Ignore.
PRODID	Ignore.
METHOD	Required as described in the Method list above.
SOURCE	Ignore
NAME	Ignore.
VERSION	Ignore.
Event-Related Components	Fallback
-----	-----
VFREEBUSY	Reply with Not Supported.
VALARM	Reply with Not Supported.
VTIMEZONE	Required if RRULE or RDATE is implemented; otherwise ignore and use local time.
Component Property	Fallback
-----	-----
ATTACH	Ignore.
ATTENDEE	Required if EVENT-REQUEST is not implemented; otherwise reply with Not Supported.
CATEGORIES	Ignore.
CLASS	Ignore.
COMMENT	Ignore.
COMPLETED	Ignore.
CREATED	Ignore.
DAYLIGHT	Required if VTIMEZONE is implemented; otherwise Ignore.
DESCRIPTION	Required.
DELEGATED-FROM	Required if EVENT-DELEGATE is implemented; otherwise Ignore.
DELEGATED-TO	Required if EVENT-DELEGATE is implemented; otherwise Ignore.

Silverberg/Mansour/Dawson/Hopson 87

Expires September 1998

Internet Draft

iTIP

March 13, 1998

DUE	Ignore.
DURATION	Reply with Not Supported.
DTSTAMP	Required.
DTSTART	Required.

DTEND	Required.
EXDATE	Ignore.
EXRULE	Ignore.
FREEBUSY	Reply with Not Supported.
LAST-MODIFIED	Ignore.
LOCATION	Required.
PRIORITY	Ignore.
RELATED-TO	Ignore.
RDATE	Ignore. If implemented, VTIMEZONE MUST also be implemented.
RRULE	Ignore. The first instance occurs on the DTStart property.
RECURRENCE-ID	Required if RRULE is implemented; otherwise Ignore.
REQUEST-STATUS	Required.
RESOURCES	Ignore.
SEQUENCE	Required.
STATUS	Ignore.
SUMMARY	Ignore.
TRANSP	Required if FREEBUSY is implemented; otherwise Ignore.
TZNAME	Required if VTIMEZONE is implemented; otherwise Ignore.
TZOFFSET	Required if VTIMEZONE is implemented; otherwise Ignore.
URL	Ignore.
UID	Required.
X-	Ignore.

[5.1.1.2](#) To-Do-Related Fallbacks

Method	Fallback
-----	-----
PUBLISH	Required.
CANCEL	Required.
ADD	Required.
REQUEST	TODO-PUBLISH.
REPLY	Required.
iCalendar	
Property	Fallback
-----	-----
CALSCALE	Ignore; assume GREGORIAN.
GEO	Ignore.
PRODID	Ignore.
METHOD	Required as described in the Method list above.
SOURCE	Ignore
NAME	Ignore.
VERSION	Ignore.

Internet Draft

iTIP

March 13, 1998

To-Do-Related
Components

Fallback

VALARM

Reply with Not Supported.

VTIMEZONE

Required if RRULE or RDATE is implemented;
otherwise ignore and use local time.Component
Property

Fallback

CALSCALE

Ignore; assume GREGORIAN.

GEO

Ignore.

PROID

Ignore.

PROFILE

Required as described in the Method list above.

SOURCE

Ignore.

NAME

Ignore.

VERSION

Assume "2.0".

Property

Fallback

ATTACH

Ignore.

ATTENDEE

Required if REQUEST is not implemented; otherwise
ignore.

CATEGORIES

Ignore.

CLASS

Ignore.

COMMENT

Ignore.

COMPLETED

Required.

CREATED

Ignore.

DAYLIGHT

Required if VTIMEZONE is implemented; otherwise
ignore.

DESCRIPTION

Required.

DUE

Required.

DURATION

Ignore. Reply with Not Supported.

DTSTAMP

Required.

DTSTART

Required.

EXDATE

Ignore. Reply with Not Supported.

EXRULE

Ignore. Reply with Not Supported.

LAST-MODIFIED

Ignore.

LOCATION

Ignore.

PERCENT-COMPLETE

Ignore.

PRIORITY

Required.

RELATED-TO

Ignore.

RDATE

Ignore. If implemented, VTIMEZONE MUST also be
implemented.

RRULE

Ignore. The first instance occurs on the DTSTART
property.

RESOURCES	Ignore.
SEQUENCE	Required.
STATUS	Required.
SUMMARY	Ignore.
TRANSP	Required if FREEBUSY is implemented; otherwise ignore.
TZNAME	Required if VTIMEZONE is implemented; otherwise ignore.
TZOFFSET	Required if VTIMEZONE is implemented; otherwise ignore.
URL	Ignore.

Silverberg/Mansour/Dawson/Hopson 89

Expires September 1998

Internet Draft

iTIP

March 13, 1998

UID	Required.
X-	Ignore.

[5.1.3](#) Journal-Related Fallbacks

Method	Fallback

PUBLISH	Implementations MAY ignore the profile type. The REQUEST-STATUS "302;Request Not supported" MUST be returned.
CANCEL	Implementations MAY ignore the profile type. The REQUEST-STATUS "302;Request Not supported" MUST be returned.
ADD	Implementations MAY ignore the profile type. The REQUEST-STATUS "302;Request Not supported" MUST be returned.
iCalendar Property	Fallback

CALSCALE	Ignore; assume GREGORIAN.
GEO	Ignore.
PRODID	Ignore.
METHOD	Required as described in the Method list above.
SOURCE	Ignore
NAME	Ignore.
VERSION	Ignore.
Journal-Related Components	Fallback

VTIMEZONE	Required if RRULE or RDATE is implemented; otherwise

	ignore and use local time.
CALSCALE	Ignore; assume GREGORIAN.
GEO	Ignore.
PRODID	Ignore.
METHOD	Required as described in the Method section above.
SOURCE	Ignore
NAME	Ignore.
VERSION	Assume "2.0".

Component	
Property	Fallback
-----	-----
ATTACH	Ignore.
ATTENDEE	Required if JOURNAL-REQUEST is implemented; otherwise ignore.
CATEGORIES	Ignore.
CLASS	Ignore.
COMMENT	Ignore.
CREATED	Ignore.
DAYLIGHT	Required if VTIMEZONE is implemented; otherwise

Silverberg/Mansour/Dawson/Hopson 90

Expires September 1998

Internet Draft

iTIP

March 13, 1998

	ignore.
DESCRIPTION	Required.
DTSTAMP	Required.
DTSTART	Required.
DTEND	Required.
EXDATE	Ignore.
EXRULE	Ignore.
LAST-MODIFIED	Ignore.
RELATED-TO	Ignore.
RDATE	Ignore. If implemented, VTIMEZONE MUST also be implemented.
RRULE	Ignore. The first instance occurs on the DTSTART property.
SEQUENCE	Required.
STATUS	Ignore.
TRANSP	Required if FREEBUSY is implemented; otherwise ignore.
TZNAME	Required if VTIMEZONE is implemented; otherwise ignore.
TZOFFSET	Required if VTIMEZONE is implemented; otherwise ignore.
URL	Ignore.
UID	Required.
X-	Ignore.

5.2 Latency Issues

With a store-and-forward transport, it is possible for events to arrive out of sequence. That is, a "CANCEL" method may be received prior to receiving the associated "REQUEST" for the calendar component. This section discusses a few of these scenarios.

5.2.1 Cancellation of an Unknown Calendar Component.

When a "CANCEL" method is received before the original "REQUEST" method the calendar will be unable to correlate the "UID" property of the cancellation with an existing calendar component. It is suggested that messages that can not be correlated that also contain non-zero sequence numbers be held and not discarded. Implementations MAY age them out if no other messages arrive with the same "UID" property value and a lower sequence number.

5.2.2 Unexpected Reply from an Unknown Delegate

When an "Attendee" delegates an item to another CU they MUST send a "REPLY" method to the "Organizer" using the "ATTENDEE" properties to indicate that the request was delegated and to whom. Hence, it is possible for an "Organizer" to receive an "REPLY" from a CU not listed as one of the original "Attendees". The resolution is left to the implementation but it is expected that the calendaring software will either accept the reply or hold it until the related "REPLY" method is received from the "Delegator". If the version of the "REPLY" method is

Silverberg/Mansour/Dawson/Hopson 91

Expires September 1998

Internet Draft

iTIP

March 13, 1998

out of date the "Organizer" SHOULD treat the message as a "STATUS-REQUEST" and update the delegate with the correct version.

5.3 Sequence Number

Under some conditions, a CUA may receive requests and replies with the same "SEQUENCE" property value. The "DTSTAMP" property is utilized as a tie-breaker when two items with the same "SEQUENCE" property value are evaluated. Furthermore, the "SEQUENCE" property SHOULD only be incremented when one or more of the following properties changes:

DTSTART

DTEND
RDATE
RRULE
EXDATE
EXRULE
DUE (for VTOD0 components)
and possibly LOCATION (at the user's discretion)

6 Security Considerations

ITIP is an abstract transport protocol which will be bound to a real-time transport, a store-and-forward transport, and perhaps other transports. The transport protocol will be responsible for providing facilities for authentication and encryption using standard Internet mechanisms that are mutually understood between the sender and receiver.

6.1 Security Threats

6.1.1 Spoofing the "Organizer"

In iTIP, the "Organizer" (or someone working on the "Organizer's" behalf) is the only person authorized to make changes to an existing "VEVENT", "VTOD0", "VJOURNAL" calendar component and republish it or redistribute updates to the "Attendees". An iCalendar object that maliciously changes or cancels an existing "VEVENT", "VTOD0" or "VJOURNAL" calendar component may be constructed by someone other than the "Organizer" and republished or sent to the "Attendees".

6.1.2 Spoofing the "Attendee"

In iTIP, an "Attendee" of a "VEVENT" or "VTOD0" calendar component (or someone working on the "Attendee's" behalf) is the only person authorized to update any parameter associated with their "ATTENDEE" property and send it to the "Organizer". An iCalendar object that maliciously changes the "ATTENDEE" parameters may be constructed by someone other than the real "Attendee" and sent to the "Organizer".

Silverberg/Mansour/Dawson/Hopson 92

Expires September 1998

Internet Draft

iTIP

March 13, 1998

6.1.3 Unauthorized Replacement of the Organizer

There will be circumstances when "Attendees" of an event or to-do

decide, using out-of-band mechanisms, that the "Organizer" must be replaced. When the new "Organizer" sends out the updated "VEVENT" or "VTODO" the "Attendee's" CUA will detect that the "Organizer" has been changed, but it has no way of knowing whether or not the change was mutually upon.

6.1.4 Eavesdropping

The iCalendar object is constructed with human-readable clear text. Any information contained in an iCalendar object may be read and/or changed by unauthorized persons while the object is in transit.

6.1.5 Flooding a Calendar

Implementations MAY provide a means to automatically incorporate "REQUEST" methods into a calendar. This presents the opportunity for a calendar to be flooded with requests, which effectively block all the calendar's free time.

6.1.6 Procedural Alarms

The "REQUEST" methods for "VEVENT" and "VTODO" calendar components MAY contain "VALARM" calendar components. The "VALARM" calendar component may be of type PROCEDURE and MAY have an attachment containing an executable program. Implementations that incorporate these types of alarms are subject to any virus or malicious attack that may occur as a result of executing the attachment.

6.1.7 Unauthorized REFRESH Requests

It is possible for an "Organizer" to receive a "REFRESH" request from someone who is not an "Attendee" of an event or to-do. Only "Attendee's" of an event or to-do are authorized to receive replies to "REFRESH" requests. Replying to such requests to anyone who is not an "Attendee" may be a security problem.

6.2 Recommendations

For an application where the information is sensitive or critical and the network is subject to a high probability of attack, iTIP transactions SHOULD be secured. This may be accomplished using public key technology, specifically Security Multiparts for MIME [[RFC1847](#)] in the iTIP transport binding. This helps mitigate the threats of spoofing, eavesdropping and malicious changes in transit.

Internet Draft

iTIP

March 13, 1998

6.2.1 Use of [RFC1847] to secure iTIP transactions

iTIP transport bindings SHOULD provide a mechanism based on Security Multiparts for MIME [RFC1847] to enable authentication of the sender's identity, and privacy and integrity of the data being transmitted. This allows the receiver of a signed iCalendar object to verify the identity of the sender. This sender may then be correlated to an "ATTENDEE" property in the iCalendar object. If the correlation is made and the sender is authorized to make the requested change or update then the operation may proceed. It also allows the message to be encrypted to prevent unauthorized reading of the message contents in transit. iTIP transport binding documents describe this process in detail.

6.2.2 Implementation Controls

The threat of unauthorized replacement of the "Organizer" SHOULD be mitigated by a calendar system that uses this protocol by providing controls or alerts that make "Calendar Users" aware of such "Organizer" changes and allowing them to decide whether or not the request should be honored.

The threat of flooding a calendar SHOULD be mitigated by a calendar system that uses this protocol by providing controls that may be used to limit the acceptable sources for iTIP transactions, and perhaps the size of messages and volume of traffic, by source.

The threat of malicious procedural alarms SHOULD be mitigated by a calendar system that uses this protocol by providing controls that may be used to disallow procedural alarms in iTIP transactions and/or remove all alarms from the object before delivery to the recipient.

The threat of unauthorized "REFRESH" requests SHOULD be mitigated by a calendar system that uses this protocol by providing controls or alerts that allow "Calendar User" to decide whether or not the request should be honored. An implementation MAY decide to maintain, for audit or historical purposes, "Calendar Users" who were part of an attendee list and who were subsequently uninvited. Similar controls or alerts should be provided when a "REFRESH" request is received from these "Calendar Users" as well.

7 Acknowledgments

A hearty thanks to the following individuals who have participated in the drafting, review and discussion of this memo:

Anik Ganguly, Bruce Kahn, John Noerenberg, Leo Parker, John Rose, Vinod Seraphin, Richard Shusterman, Derik Stenerson, John Sun, Kevin Tsurutome.

Silverberg/Mansour/Dawson/Hopson 94

Expires September 1998

Internet Draft

iTIP

March 13, 1998

8 Bibliography

[ICAL] "Internet Calendaring and Scheduling Core Object Specification - iCalendar", Internet-Draft, October 1997, <ftp://ftp.ietf.org/internet-drafts/draft-ietf-calsch-ical-03.txt>.

[ICMS] "Internet Calendaring Model Specification", Internet-Draft, October 1997, <ftp://ftp.ietf.org/internet-drafts/draft-ietf-calsch-mod-01.txt>.

[ID-UTF8] "UTF-8, a transformation format of Unicode and ISO 10646", Internet-Draft, July 1996, <ftp://ftp.ietf.org/internet-drafts/draft-bergeau-utf8-01.txt>.

[IMIP] "iCalendar Message-Based Interoperability Protocol - iMIP", Internet-Draft, October 1997, <ftp://ftp.ietf.org/internet-drafts/draft-ietf-calsch-imip-01.txt>.

[ISO8601] "Data elements and interchange formats - information interchange - Representation of dates and times", ISO 8601, 1996-06-15, +1 (212) 642-4900 for ANSI Sales.

[VCAL] "vCalendar - The Electronic Calendaring and Scheduling Format - Version 1.0", Versit Consortium, September 18, 1996, <http://www.imc.org/pdi/vcal-10.doc>.

[VCARD] "vCard - The Electronic Business Card Exchange Format - Version 2.1", Versit Consortium, September 18, 1996, <http://www.imc.org/pdi/vcard-21.doc>.

[RFC821] Postel, "Simple Mail Transfer Protocol", [RFC 821](http://www.rfc821.org), organization name, November 1996, <http://ds.internic.net/rfc/rfc821.txt>.

[RFC1983] "Internet Users' Glossary", [RFC 1983](http://www.rfc1983.org), August 1996, <http://ds.internic.net/rfc/rfc1983.txt>.

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels",
[RFC 2119](http://ds.internic.net/rfc/rfc2119.txt), March 1997, <http://ds.internic.net/rfc/rfc2119.txt>.

[RFC2045] N. Freed and N. Borenstein, "Multipurpose Internet Mail
Extensions - Part One - Format of Internet Message Bodies", [RFC 2045](http://ds.internic.net/rfc/rfc2045.txt),
Innosoft, First Virtual, November 1996,
<http://ds.internic.net/rfc/rfc2045.txt>.

[RFC2046] N. Freed and N. Borenstein, "Multipurpose Internet Mail
Extensions - Part Two - Media Types", [RFC 2046](http://ds.internic.net/rfc/rfc2046.txt), Innosoft, First Virtual,
November 1996, <http://ds.internic.net/rfc/rfc2046.txt>.

[UNICODE] The Unicode Consortium, "The Unicode Standard -Version 2.0",
Addison-Wesley Developers Press, July 1996. UTF-8 is described in
section A-2.

[US-ASCII] Coded Character Set--7-bit American Standard Code for
Information Interchange, ANSI X3.4-1986.

Silverberg/Mansour/Dawson/Hopson 95 Expires September 1998

Internet Draft iTIP March 13, 1998

9 Authors Addresses

The following address information is provided in a MIME-VCARD,
Electronic Business Card, format.

The authors of this draft are:

```
BEGIN:VCARD
VERSION:2.1
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:Frank_Dawson@Lotus.com
URL:http://home.earthlink.net/~fdawson
END:VCARD
```

```
BEGIN:VCARD
VERSION:2.1
FN:Ross Hopson
ORG:On Technology, Inc.
ADR;WORK;POSTAL;PARCEL:Suite 1600;;434 Fayetteville St.
```

Mall, Two Hannover Square;Raleigh;NC;27601
TEL;WORK;MSG:+1-919-890-4036
TEL;WORK;FAX:+1-919-890-4100
EMAIL;INTERNET:rhopsn@on.com
END:VCARD

BEGIN:VCARD
VERSION:2.1
FN:Steve Mansour
ORG:Netscape Communications Corporation
ADR;WORK;POSTAL;PARCEL;;;501 East Middlefield Road;Mountain
View;CA;94043;USA
TEL;WORK;MSG:+1-650-937-2378
TEL;WORK;FAX:+1-650-937-2103
EMAIL;INTERNET:sman@netscape.com
END:VCARD

BEGIN:VCARD
VERSION:2.1
FN:Steve Silverberg
ORG:Microsoft Corporation
ADR;WORK;POSTAL;PARCEL;;;One Microsoft Way;
Redmond;WA;98052-6399;USA
TEL;WORK;MSG:+1-425-936-9277
TEL;WORK;FAX:+1-425-936-8019
EMAIL;INTERNET:stevesil@Microsoft.com
END:VCARD

Silverberg/Mansour/Dawson/Hopson 96

Expires September 1998

Internet Draft

iTIP

March 13, 1998

The iCalendar object is a result of the work of the Internet Engineering Task Force Calendaring and scheduling Working Group. The chairman of that working group is:

BEGIN:VCARD
FN:Anik Ganguly
ORG:Campbel Services, Inc.
ADR;WORK;POSTAL;PARCEL:10 Floor;;21700 Northwestern
Highway;Southfield;MI;48075;USA
TEL;WORK;MSG:+1-248-559-5955
TEL;WORK;FAX:+1-248-559-5034
EMAIL;INTERNET:anik@ontime.com
END:VCARD

10 Full Copyright Statement

"Copyright (C) The Internet Society (date). 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 may 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.