

IETF Review of the RFC 2446bis-07 draft: iCalendar Transport-Independent Interoperability Protocol (iTIP)

Reviewer: Reinhold Kainhofer, reinhold@kainhofer.com

September 7, 2008

Contents

About this review	2
General observations	2
1 Introduction and overview	2
2 Interoperability Models	2
3 Application Protocol Elements	4
3.1 Common Component Restriction Tables	4
3.2 Methods for VEVENT Calendar Components	4
3.3 Methods for VFREEBUSY Components	7
3.4 Methods for VTODO Components	7
3.5 Methods for VJOURNAL Components	8
3.6 Status Replies	9
3.7 Implementation Considerations	9
4 Examples	9
4.1 Published Event Examples	9
4.2 Group Event Examples	9
4.3 Busy Time Examples	10
4.4 Recurring Event and Time Zone Examples	10
4.5 Group To-do Examples	12
4.6 Journal Examples	12
4.7 Other Examples	12
5 Application Protocol Fallbacks	13
6 Security Considerations	14
7 IANA Considerations	14
8 Acknowledgments	14
9 Normative References	14
A Appendices	14
Issues in rfc2445bis-08	15

About this review

Since this draft is an update to an already existing standard, I will not comment on its usefulness in general, but rather concentrate on technical details. In particular, I looked at possible contradictions to rfc2445bis-08, unclear explanations, formulations that might possibly be misunderstood, and typos. The page-breaking of the draft is also not ideal, but I suppose this will be fixed in the final release, so I will not further comment on it. Even though several of the issues I found were already detailed in Lisa Dusseault's review, I will still list them here.

I will go through the whole draft section by section, mixing comments of different types, like errors, unclear formulations, other suggestions and questions I still have, etc. I will use one item per issue I found, even when I found multiple issues in a section. This will make it easier to discuss the issues.

Quotes from the draft and rfc2445bis as well as text to be inserted into the draft will be placed in »guillemets« to distinguish them from text in quotations marks in the draft (like all property and method names).

General observations

1. The capitalization of »event« is inconsistent throughout the draft. Sometimes, »Event« is used inside a sentence, at other times »event« is used.

1 Introduction and overview

2. Abstract (p.1): »calendar systems« should be replaced by »calendar systems« to prevent any possible confusion. iTIP is only about the Gregorian calendar system.
3. Sec. 1.3 (p.6): This draft also describes forwarding, where there is a third role involved, namely "Other CUs". Also, when delegating, the delegate is not an »"Attendee" asked to participate by the "Organizer"«, so it does not strictly fulfill the definition of the Attendee role of section 1.3., either.
4. Sec. 1.3 (p.7): The definition of the Attendee uses »participate«, which does not make sense for to-dos.
5. Sec. 1.4 (p.7): In the REQUEST Description: Change »Meeting Requests« to »Meeting requests«
6. Sec. 1.4 (p.8): Why are Attendees not allowed to PUBLISH a group event?

2 Interoperability Models

7. Sec. 2.1.1 (p.9): »When an "Organizer" issues the initial entry, "Attendee" status is unknown.« The Organizer might have had other communication with the Attendee and already know the status from other means than iTIP. I would insert the word »typically« before »unknown«.
8. Sec. 2.1.4 (p.10f): This section heavily refers to the rules for incrementing the "SEQUENCE" number in [LD.ietf-calsify-rfc2445bis]. However, rfc2445bis-08 only says: »It is monotonically incremented by the "Organizer's" CUA each time the "Organizer" makes a significant revision to the calendar component.« The rest of the rules was removed from rfc2445bis in rfc2445bis-08, but has not been added to rfc2446bis. I propose to remove all references to rfc2445bis in this section and instead paste the rules into rfc2446bis:

The "SEQUENCE" property is used by the "Organizer" to indicate revisions to the calendar component. When the "Organizer" makes

changes to one of the following properties, the sequence number MUST be incremented:

- * "DTSTART"
- * "DTEND"
- * "DURATION"
- * "DUE"
- * "RDATE"
- * "RRULE"
- * "EXDATE"
- * "STATUS"

In addition, changes made by the "Organizer" to other properties MAY also require the sequence number to be incremented. The "Organizer" CUA MUST increment the sequence number whenever it makes changes to properties in the calendar component that the "Organizer" deems will jeopardize the validity of the participation status of the "Attendees". For example, changing the location of a meeting from one location to another distant location could effectively impact the participation status of the "Attendees".

Depending on the METHOD, the "SEQUENCE" property MUST follow these rules in the context of iTIP:

- * For the "PUBLISH" and "REQUEST" methods, the "SEQUENCE" property value is incremented according to the rules stated above.
- * The "SEQUENCE" property value MUST be incremented each time the ... [rest is taken as found in rfc2446bis-07] ...

9. Sec. 2.1.5 (p.11/12): From this description, it appears to me that all calendar components with the same UID also need to have the SEQUENCE incremented simultaneously (i.e. if the Organizer changes just one occurrence from a recurring sequence, he must increase the SEQUENCE not only of that particular VEVENT, but also of the VEVENT containing the RRULE). This was not clear to me from rfc2445bis and is never clearly spelled out. In particular, rfc2445bis says on page 142 that »SEQUENCE defines the revision sequence number of the calendar component within a sequence of revisions« and defines calendar component on pages 52ff as any VEVENT, VTODO, VJOURNAL, etc. inside the VCALENDAR. In particular, a VEVENT with an RRULE and another VEVENT with the same UID, but using RECURRENCE-ID to modify one single instance of the recurrence, are actually two calendar components, as far as I understand the wording. The description of SEQUENCE can then be understood that both can have different SEQUENCE numbers...

This should probably be clarified in rfc2445bis.

If, however, it is possible for the two components to have different SEQUENCE numbers, then items 3. and 4. of the list in section 2.1.5 need to be changed from »UID« to »UID and RECURRENCE-ID«.

10. Sec. 2.1.5 (p.12): »Hence, CUAs must persist« should rather be »Hence, CUAs MUST persist«
11. Sec. 2.1.5 (p.12): I don't understand what is meant with the sentence »Furthermore, for each "ATTENDEE" property of a component CUAs must persist the "SEQUENCE" and "DT-STAMP" property values associated with the "Attendee's" response.« First, I don't understand the meaning: Does this mean that the Organizer's CUA needs to store for each Attendee the last SEQUENCE number, to which the Attendee has responded? In this case, I don't understand the

reason behind it. Or does it mean that the Attendee's CUA needs to preserve the SEQUENCE of his/her last response, so that it's possible to decide whether a response to a new REQUEST from the Organizer is needed. In that case, I don't think that sentence is needed, either, because it is stated in the draft already that only the Organizer is allowed to make changes to a component. Also, which CUAs are meant: The CUA of the Organizer or of the Attendee?

3 Application Protocol Elements

12. Sec. 3 (p.12, methods table): From the VJOURNAL column, it seems to me that CANCEL and ADD can also be used change components that were PUBLISHED, not only REQUESTED ones. This is never clearly spelled out, in particular not for VEVENT and VTODO.
13. All the restriction tables in this chapter list the VALARM component parallel to VEVENT, VTODO, VJOURNAL and VFREEBUSY. I regard this as quite unfortunate, because VALARM is not a top-level calendar component, but can only appear inside other components. As the description of the tables says that component properties are indented, I think it would make sense to also indent VALARM, which is a component component.
14. Several of the constraint tables for VEVENT in chapter 3 either disallow DTSTART at all or allow DTSTART to be absent from a component. In particular, the VEVENT REFRESH (Sec. 3.2.6, p.28) and DECLINECOUNTER (Sec. 3.2.8, p.31) tables disallow DTSTART at all, while VEVENT CANCEL (Sec. 3.2.5, p.27), allow $\gg 0$ or $1 \ll$ DTSTART. In contrast, rfc2445bis (Sec. 3.6.1, p.54) REQUIRES DTSTART to be present in all VEVENT components, so that e.g. the contents of a REFRESH are NOT VALID iCalendar! Notice, that this restriction in rfc2445bis was not in rfc2445, but was added during the update to rfc2445bis.

3.1 Common Component Restriction Tables

15. Sec. 3.1.2 (p.14, VTIMEZONE table): This table states that RRULE and RDATE MUST NOT both be present at the same time. rfc2445bis does not have such a restriction, so that some VTIMEZONE objects, which are valid in rfc2445bis, cannot be used for group scheduling. I propose to remove that restriction.

3.2 Methods for VEVENT Calendar Components

16. Sec. 3.2 (p.15, methods table, REQUEST): \gg Event requests \ll instead of \gg Event Requests \ll . Also, I think it should be \gg MAY degrade \ll instead of \gg may degrade \ll .
17. Sec. 3.2.1 (p.16): I have never understood why a PUBLISHED event MUST NOT have any Attendees. In particular, I think it might be useful for several types of published events to contain Attendees and their participation status: panel discussions, hearings, etc.
18. Sec. 3.2.2 (p.16ff): Is there any requirement that the VEVENT REQUEST sent out to one particular Attendee MUST/SHOULD contain all other Attendees? Cf. in particular Sec. 3.2.5 (see issue 33), which states that a cancelled event MUST include all Attendees. In my opinion, there should not be such a requirement, mainly for privacy reasons. The attendees' email addresses are explicitly listed in the ATTENDEE property, so I don't regard it a good idea that all attendees automatically learn about the email addresses of all other attendees, which might be well-known persons and quite reluctant to give out their private contact details. In any case, a sentence should be inserted clarifying whether all attendees MUST/SHOULD be given in the VEVENT, or whether it is possible to send individual requests to each attendee containing only the one ATTENDEE property for this particular person.
19. Sec. 3.2.2 (p.17): The paragraph after the list is not true for delegated and forwarded events. In particular, in both cases the recipients are not the "Attendees", but "Other CUs".

20. Sec. 3.2.2.3 (p.21): In the second-to-last paragraph, there is a spurious " between method and SHOULD.
21. Sec. 3.2.2.3 (p.20/21): The delegator forwards the REQUEST to the delegate, including the delegate as a new ATTENDEE. The delegator also sends a REPLY to the Organizer with the DELEGATED-TO parameter set. However, it is not clear to me whether the REPLY to the organizer MUST/SHOULD also contain the delegate as a new ATTENDEE or not. As the comment in Sec. 4.2.5 says this is a MUST, it should be added here, too, since here is the section where delegating is actually defined. The examples – helpful as they are – should not include any new requirements not described before.
22. Sec. 3.2.2.4 (p.21): In the last sentence, it should read »... incremented and the value of ...«.
23. Sec. 3.2.2.4 (p.21): The last sentence says that »the "ORGANIZER" property has been changed to the *calendar address* of the new "Organizer".« However, rfc2445bis defines the calendar address as only the URI and does not include property parameters like CN, which should clearly be changed, too. I think it would suffice to simply say »and the value of the "ORGANIZER" property has been changed to the new "Organizer".«
24. Sec. 3.2.2.6 (p.21): There is one step missing in the description: The Attendee forwards to the uninvited CU, but there is no mentioning of how the Organizer learns of this.
25. Sec. 3.2.2.6 (p.21): How does the uninvited CU find out whether the Organizer added him/her or not. The draft only says that the Organizer MAY send a CANCEL if he rejects the new Attendee. However, if he does not, then the uninvited Attendee has no way of knowing whether he was added or not. A sentence should be added clarifying this situation. Should the uninvited CU simply send a REFRESH to the organizer? If he does not get a response there either, he can deduce that he was not added (see Sec. 6.1.6).
26. Sec. 3.2.2.6 (p.22): The last sentence says that the Attendee MUST NOT make changes to the VEVENT property set. However, this would not prohibit changes to other calendar components, like VTIMEZONE, VALARM, etc. I don't think such changes should be allowed, either.
27. Sec. 3.2.3 (p.22): Should »An "Attendee" can include« rather be »An "Attendee" MAY include«?
28. Sec. 3.2.3 (p.23): This section says that the optional properties listed in the table MUST NOT be changed. However, the rest of the draft and the comments in the table make it clear that some required parameters (DTSTAMP, UID, ORGANIZER) MUST also NOT be changed.
29. Sec. 3.2.3 (p.24): The table lists VALARM as »0«. If the original REQUEST contained a VALARM, shouldn't it be included in the REPLY, too?
30. Sec. 3.2.5 (p.24f): I don't understand how ADD should work exactly. rfc2445bis says that an instance of an event is identified by the UID and the RECURRENCE-ID. If we don't allow a RECURRENCE-ID, how does the resulting sequence of events look in iCalendar notation (i.e. what would the Organizer send in response to a REFRESH)?

I suppose it would work to add the additional date as an RDATE or a second RRULE, but only if none of the other properties is different from the original event. Sec. 4.4.7 gives an example of this case (but since it does not point out that an RDATE was added, it is very hard to figure out how the ADD was handled). If some properties are different for a single ADDED event, this can still be represented as an added RDATE and a second VEVENT using RECURRENCE-ID to specify the added instance. However, if an ADD message tries to add a recurring event with some property values different from the original (already recurring) event, then I simply don't see how this can be represented in iCalendar. RECURRENCE-ID;RANGE=THISANDFUTURE can not be used, because the change should only apply to the instances of one RRULE, not of both.

An example of this problem is the first example in Sec. 4.4.7: The original RRULE recurs on TU in »The White Room«. The ADD message adds a second rule on TH, but the location is now »The usual conference room« (see also issue 95). I don't see how these two combined can be properly represented in the same event in iCalendar. The example says that they are equivalent to the REQUEST given in Sec. 4.4.7, but that REQUEST seems to ignore the different LOCATION property value of the ADD. I think it should be clarified, how such colliding messages should be handled. In particular, do all other property values except DTSTART, RDATE and RRULE have to coincide with the original REQUEST? If not, we should clarify how that case can be handled (i.e. either discard all information from the ADD, which does not coincide? Or try to merge them as much as possible? Or simply disallow adding recurring events to an already recurring event with different properties.)

The same clarification is needed for VTODO (p.47)

31. Sec. 3.2.5 (p.26): The first sentence talks about sending a notice »to the "Attendees"«. To me it is not clear if this means sending a CANCEL message to ALL Attendees (even when just uninviting one Attendee) or only to the affected (i.e. uninvited) Attendees. Both would make sense...

The same goes for VTODO on p.47.

32. Sec. 3.2.5 (p.26): In the ATTENDEE comment in the table, »from« is missing in »being removed from the event.«

33. Sec. 3.2.5 (p.26): Why is it a MUST that all Attendees have to be included if the whole event is cancelled? The STATUS property already tells the CU(A) whether the whole event was cancelled or only some Attendees uninvited. Can't a CUA also send out individual CANCEL messages to each of the Attendees with only that one Attendee included in the VEVENT? This would prevent privacy issues, because sending out the email addresses of all Attendees is not always desirable (see also issue 18).

The same holds for VTODO (p.49)

34. Sec. 3.2.5 (p.27): The description of STATUS in the table is badly worded (Isn't this a contradiction: MUST be set to CANCELLED, and MUST NOT be present, if ...). I propose to change the first part of the description to

MUST be set to CANCELLED to cancel the entire event.

35. Sec. 3.2.6 (p.28): I think DTSTAMP and ORGANIZER also MUST be the same value as in the original REQUEST...

36. Sec. 3.2.6 (p.29): Since RECURRENCE-ID is a date-time value, it might include a reference to a time zone. In this case, a VTIMEZONE is REQUIRED (instead of the current »0«)

37. Sec. 3.2.7 (p.29, first paragraph): The COUNTER is used not only for counter proposals to the event description! I would simply delete »description« so the sentence reads »... a counter proposal to the event.«

38. Sec. 3.2.7 (p.30): The SEQUENCE value MUST NOT be changed from the original event. I would include this in the comment for SEQUENCE (it is mentioned in the examples in Sec. 4.2.4, though).

39. Sec. 3.2.7 (p.30): In the comment for STATUS, remove the space before CANCELLED.

40. Sec. 3.2.7 (p.31): At the end of this section, it should be clarified how the Organizer reacts to the COUNTER. Declining it is clear, but if he accepts it, does he have to send out a new REQUEST to all Attendees or can he simply do nothing, too? In particular, if an Attendee

sends a COUNTER with significant changes and does not receive a DECLINECOUNTER, can he assume that the Organizer has accepted it or does he have to treat it like receiving a DECLINECOUNTER? How does he learn about it other than sending a REFRESH?

41. Sec. 3.2.8 (p.32): VTIMEZONE is required if the RECURRENCE-ID contains a reference to a timezone! (like issue 36).

3.3 Methods for VFREEBUSY Components

42. Sec. 3.3 (p.32): The third paragraph says that the VFREEBUSY MUST include the ATTENDEE in this case, while the PUBLISH table in Sec. 3.3.1 expressively forbid this.
43. Sec. 3.3 (p.33): This section says »However, two different busy time periods MAY overlap«, while the restriction table in REPLY (sec 3.33, p.36) says that they MUST NOT overlap.
44. Sec. 3.3 (p.33): Shouldn't it be a SHOULD in »"FREEBUSY" properties should be sorted...«?
45. Sec. 3.3 (p.33): The paragraph starting with »Since events may span a day boundary...« is confusing at best. The free/busy periods do not necessarily coincide with events, so this is no argument. Also, the following sentence sounds a bit out of order, since it clarifies the REQUEST method. Instead of giving such a lengthy example (talking about "Individuals" not supporting busy time requests, while in fact their CUAs don't support it), wouldn't it be simpler to reformulate the whole paragraph?

I propose to append

Busy time periods may also span a day boundary.

to the previous paragraph and remove the rest of the paragraph, since it deals with a fallback and only explains what the methods fallback table in Sec. 5.1.2 already says much better.

46. Sec. 3.3.1 (p.33): Should it really be »...to provide a message for sending unsolicited...« or should it rather be »... to provide a method for sending unsolicited...«?
47. Sec. 3.3.1 (p.33): Shouldn't it be »The "ORGANIZER" property MUST be specified...«?
48. Sec. 3.3.2 (p.35): The comment for ATTENDEE sounds wrong. What is it supposed to mean?
49. Sec. 3.3.3 (p.36): Remove the parentheses in the comment for ATTENDEE and URL.
50. Sec. 3.3.3 (p.36): Remove »Multiple instances are allowed.« in the FREEBUSY comment, since »0+« already says this.
51. Sec. 3.3.3 (p.36): Why is the sorting a MUST for PUBLISH, while the introduction in Sec. 3.3 only speaks of SHOULD?
52. Sec. 3.3.3 (p.36): Add comment for UID (like for VEVENT): »MUST be the UID of the original REQUEST«

3.4 Methods for VTODO Components

53. All restriction tables list the PRIORITY property as required for to-dos, while rfc2445bis does not mention any such restriction. I don't think PRIORITY is important enough to warrant a MUST in iTIP.
54. Sec. 3.4.1 (p.38): Why are Attendees forbidden with PUBLISH?
55. Sec. 3.4.1 (p.39), Sec. 3.4.2 (p.41), Sec. 3.4.4 (p.47), Sec. 3.4.7 (p.53) and Sec. 3.4.8 (p.55): In the STATUS comment, it should be IN-PROCESS (delete space), similarly it should be NEEDS-ACTION (insert hyphen)

56. Sec. 3.4.2 (p.39): Only the meaning of REQ-PARTICIPANT is explained. Are other values allowed? If so, what do they mean in connection with to-dos?
57. Sec. 3.4.2.2 (p.42): If the REQUEST is only used to "reconfirm" the to-do, do the Attendees still have to REPLY, even if nothing has changed? Or should they simply use the value of their RSVP ATTENDEE parameter to decide?
58. Sec. 3.4.2.3 (p.42): Why can't a to-do be delegated to the Organizer?
59. Sec. 3.4.2.4 (p.43): Same as 25 for VEVENT.
60. Sec. 3.4.3 (p.44): What exactly is an »unsuccessful "VTODO" calendar component "REQUEST" method«?
61. Sec. 3.4.3 (p.43): The same comment about properties that MUST NOT be changed should be inserted as for VEVENT (issue 28) on p.23 right before the table.
62. Sec. 3.4.3 (p.44f): The UID comment should be moved to the ATTENDEE property, which it actually describes, and ATTENDEE should be changed from »1+« to »1« (see also the VEVENT REPLY table on p.23). The new UID comment should be »MUST be the UID of the original REQUEST«
63. Sec. 3.4.3 (p.45): If the VTODO in the REQUEST contained a VALARM, should this not be included in the REPLY, too?
64. Sec. 3.4.5 (p.48): »When a VTODO is cancelled« Does this mean only when the whole VTODO (with all its recurrence instances) is cancelled, or does it mean whenever a CANCEL is sent, in particular also when only a single ATTENDEE is unassigned? The same holds for VEVENT (p.26).
65. Sec. 3.4.5 (p.52): Since the RECURRENCE-ID is a date-time, it can include a reference to a timezone. In this case, a VTIMEZONE is REQUIRED.
66. Sec. 3.4.7 (p.52): The second sentence of the first paragraph is redundant, since the first sentence already says that the counterproposal is submitted to the Organizer.
67. Sec. 3.4.7 (p.52): If the counterproposal only contains some minor modifications (e.g. correcting typos in the description), does the Organizer still have to send the REQUEST to all Attendees and set RSVP=TRUE?
68. Sec. 3.4.7 (p.52): Remove the »;« after "ATTENDEE" property
69. Sec. 3.4.7 (p.53): In the comment for RECURRENCE-ID remove the spurious »3.5«
70. Sec. 3.4.8 (p.55): What does the comment for ATTENDEE mean? Does it mean that in a DECLINECOUNTER the Organizer has to insert all original Attendees?

3.5 Methods for VJOURNAL Components

71. Sec. 3.5.1 (p.57): What is the »original SEQUENCE number« in the comment for SEQUENCE for a PUBLISH of a VTODO? I would delete that first sentence.
72. Sec. 3.5.3 (p.61): VJOURNAL does not allow ATTENDEE, so the presence should be changed to »0« from »0+«.

3.6 Status Replies

73. Sec. 3.6 (p.62, item 2. A. and B.) I propose to exchange rules A. and B., delete the last sentence from the current B., and start the current A. with »Otherwise, if any one component«. This makes the logic much clearer.
74. Sec. 3.6 (p.62ff, column »Offending Data«): The comments in the »Offending Data« column speak about specifying the offending data, but don't make it clear where to specify it. I suppose the idea is to give the offending data in the extdata part of the status message, right? Maybe this could be clarified in a sentence before the table.
75. Sec. 3.6 (p.64, Code 5.0): What does the Return Status Description »Request MAY supported.« mean? (It is grammatically wrong, too.) In particular, the Offending Data column speaks about the METHOD (should be written in all-capitals) property

3.7 Implementation Considerations

76. Sec. 3.7.1 (p.65): In the second paragraph it should be »discrete« instead of »discreet«
77. Sec. 3.7.1 (p.66): In the last paragraph of this section it should be »or to-do has been made« (instead of »has be made«)
78. Sec. 3.7.1 (p.66): The last three sentences of this section (discussing RECURRENCE-ID) are plain wrong. In particular, they are in contradiction to Sec. 3.8.4.4 of rfc2445bis: The RECURRENCE-ID is always the start date of the original recurrence instance, even after the change has been made. As an example, take an event recurring each Monday at 9:00. One of these meetings is moved to 12:00 and its LOCATION is changed. Even after the change has been propagated to the Attendees, the RECURRENCE-ID of that event will still be YYYYMMDDT090000, otherwise the recurrence rule would generate an additional event at 09:00 and a RECURRENCE-ID of YYYYMMDDT120000 does not appear in the recurrence set either. This is clearly explained in rfc2445bis (Sec. 3.8.4.4, p.116) using the Friday to Thursday change. The RECURRENCE-ID stays on the original date.
79. Sec. 3.7.3 (p.67): This draft allows clients to completely ignore X-Tokens. Granted, for REPLY, REFRESH etc. there is no advantage in preserving them (because the Attendee's CUA doesn't handle them anyway and the Organizer already has their value), but when delegating or forwarding calendar components, the Other CU might be interested in the X-Tokens, because his CUA might know how to handle them.

4 Examples

4.1 Published Event Examples

80. Sec. 4.1 (p.67): In the first paragraph, change »e- mailing« to »e-mailing« (remove space) and replace »and posting« by »or posting« to make it clearer that this enumeration is not exhaustive and PUBLISH can be used in other ways, too.

4.2 Group Event Examples

81. Sec. 4.2 (p.71): In the table change »tentative« to »TENTATIVE«. According to rfc2445bis Sec. 3.1 and 3.2 enumerated property values are case-insensitive, but rfc2445bis and rfc2446bis consistently uses all-capitals for enumerated parameter values.
82. Sec. 4.2.1 (p.72): The explanation details the default value for PARTSTAT and when it can be left out, but does not mention that the same also holds for CUTYPE=INDIVIDUAL, which could also be left out in the example

83. Sec. 4.2.1 (p.72): The DTSTART and DTEND coincide. Sec. 3.8.2.2 of rfc2445bis says that the DTEND MUST be later in time than DTSTART! Also, how much sense does a Conference of 0 seconds make?
84. Sec. 4.2.2 (p.73) and Sec. 4.2.4 (p.76): As explained in issue 14 above, rfc2445bis REQUIRES DTSTART to be present in a VEVENT, while the REPLY, REFRESH and DECLINECOUNTER methods defined in this draft forbid it. These examples are not valid iCalendar objects as specified by rfc2445bis. The same holds of course for all further REPLY, REFRESH and DECLINECOUNTER examples.
85. Sec. 4.2.4 (p. 74f): The description says that the given counter-proposal also changes the location, but the LOCATION property value is the same in the REQUEST and the COUNTER.
86. Sec. 4.2.4 (p.75): The counter-proposal contains the DTSTAMP property twice. According to the restrictions table in Sec. 3.2.7 and the VEVENT definition in Sec. 3.6.1 on page 54 of rfc2445bis this is not allowed.
87. Sec. 4.2.5 (p.76): The second to fourth items of the list should be further indented, since they are actually subitems of the first list item.
88. Sec. 4.2.5 (p.76): The last item in the list actually contradicts what is said in Sec. 3.2.2.3 (p.20), namely that the delegator MUST add an "ATTENDEE" property describing the Delegate to the REQUEST that he sends to the Delegate. Thus, he definitely MUST NOT send a copy of the original REQUEST, but rather a slightly modified one.
89. Sec. 4.2.10 (p.82): In the table replace »Remove an "B" as« by »Remove "B" as«.
90. Sec. 4.2.10 (p.82): In the table it is not clear to me whether it is a requirement for the Organizer to send the updated event to the remaining Attendees after B was cancelled or not. The table describes just one cause of action, but from the definitions in Chapter 3, it is also not clear to me whether the Organizer MAY/SHOULD/MUST send a new REQUEST to all remaining Attendees after one Attendee was cancelled.

4.3 Busy Time Examples

91. Sec. 4.3 (p.84): Before »Individual A publishes busy time for one week«, a subsection header »4.3.1 Publish Busy Time« should be inserted for consistency (all other METHODS have their own subsection)
92. Sec. 4.3 (p.85): The description of the example says that busy time for one week was published, but the period specified by DTSTART and DTEND actually is only 6 days! Furthermore, the VFREEBUSY contains FREEBUSY property values that start way beyond the DTEND. rfc2445bis says in Sec. 3.6.4 on p.63 that for published busy time the »DTSTART and DTEND properties specify an inclusive time window that surrounds the busy time information«. In other words, all FREEBUSY periods MUST be inside the interval specified by DTSTART and DTEND.

4.4 Recurring Event and Time Zone Examples

93. Sec. 4.4.1 (p.87): The description says that this event is for a weekly phone conference, but the RRULE uses INTERVAL=20, which means that the event recurs every 20 weeks! The correct RRULE would be

```
RRULE:FREQ=WEEKLY;COUNT=20;WKST=SU;BYDAY=TU
```

I blame this on the ambiguous formulation in rfc2445bis: In Sec. 3.3.10 on p.40 it says »The INTERVAL rule part contains a positive integer representing how often the recurrence rule repeats.« Of course, it is supposed to be interpreted in the sense "in which intervals the r.r. repeats", but can admittedly also be read as "how many times the r.r. repeats". Apparently this was not clear to the person who included this example in rfc2446.

94. Sec. 4.4.1 (p.88): The paragraph describing the representation of the times in different time zones contains several issues / confusions:

- (a) The time zone difference for Attendee "b" is expressed as the offset from PDT, while the time zone difference for Attendee "c" is given as the offset from UTC, which makes comparing these two times much harder.
- (b) In the explicit time listings below, "GMT" is used instead of "UTC".
- (c) JST is 9 hours ahead of UTC, not 8 hours as claimed here.
- (d) On November 11, 1997, San Jose is no longer in PDT, but in PST.
- (e) On Wed, September 10, 1997, San Jose is still in PDT, not in PST as claimed.
- (f) The representation of the exception dates in the other time zones are calculated wrong: 14:00 PDT on Sept 9 is NOT 23:00 GMT (=UTC), but rather 23:00 CEST or 21:00 UTC. Similarly, 14:00 PST (=UTC-8) on Oct 28 is 22:00 UTC or 23:00 CET or 07:00 JST (UTC+9), not 06:00 JST!

I would give all times offsets relative to PDT and list the local time zones and all local times mentioned together with their shift from UTC. In the list of exceptions, I would also list the PDT/PST date/times first, as they are the real exception dates, the representation in UTC and JST just helps the user to understand the issues with time zones.

All in all, I think that paragraph should be:

```
The repeating event starts on Tuesday, July 1, 1997 at 2:00pm PDT
(UTC-7). "Attendee" B@example.fr is in France where the local time
on this date is 9 hours ahead of PDT or 23:00 CEST (UTC+2).
"Attendee" C@example.jp is in Japan where local time is 16 hours
ahead of PDT or Wednesday, July 2 at 06:00 JST (UTC+9). 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 PDT.
```

There are also two exception dates to the recurrence rule. The first one is:

```
TUE, 09-SEP-1997 14:00 PDT (UTC-7)
TUE, 09-SEP-1997 23:00 CEST (UTC+2)
WED, 10-SEP-1997 06:00 JST (UTC+9)
```

and the second is:

```
TUE, 28-OCT-1997 14:00 PST (UTC-8)
TUE, 28-OCT-1997 23:00 CET (UTC+1)
WED, 29-OCT-1997 07:00 JST (UTC+9)
```

95. Sec. 4.4.7 (p.93f): It would be very helpful if the comments included an explanation how the added times are added to the original VEVENT. In particular, a single event is added by

adding its DTSTART as an RDATE to the original event, possibly with another VEVENT using RECURRENCE-ID to modify some of its properties if the properties of the ADD differed from the original event. Similarly, another recurring event can be added by either modifying the existing RRULE or adding a second RRULE (however, rfc2445bis deprecates the use of two RRULEs for a calendar component!). However, it is not clear how one should proceed if the added recurring event differs from the original event (e.g. has a different LOCATION like the second example of this section). As explained in 30, the second example adds another recurrence rule to an event with an already existing recurrence rule, but the added event has a different LOCATION than the original event. The PUBLISH example, which is claimed to be equal to the REQUEST and the subsequent ADD, however, ignores this changed LOCATION property value and only uses the LOCATION property value of the original event. It should be clarified if changed property values in the ADD should really be ignored or not. If not, this case with an added recurrence needs to be explained further (whether and how it can be represented in iCalendar at all!).

96. Sec. 4.4.9 (p.100): The REPLY contains a REQUEST-STATUS with status code 3.0 and one with code 2.8. However, the previous explanation of REQUEST-STATUS in Sec. 3.6 expressively forbids any non-3.x code if a 3.x code is already present. Thus, the 2.8 code entry should be deleted from the event.

4.5 Group To-do Examples

97. Sec. 4.5.1 (p.102): The STATUS property value should be NEEDS-ACTION (Hyphen is missing; enumerated property values are case-insensitive, still I think we should use all-capitals consistently)
98. Sec. 4.5.6 (p.104): The STATUS property value should be IN-PROCESS rather than the invalid IN-PROGRESS
99. Sec. 4.5.7 (p.105): Time zones cannot be given as -0700, so I think DTSTART and DUE should be

```
DTSTART:19980101T170000Z
DUE:19980103T170000Z
```

Additionally, the STATUS property value should be NEEDS-ACTION (hyphen missing)

100. Sec. 4.5.7.2 (p.105): This section talks about how recurrence works in general for to-dos, which would much better belong in rfc2445bis rather than rfc2446bis. So I think this whole section should be moved to rfc2445bis. Also, this section speaks of the due date specified in a "REQUEST", while of course the same holds true for PUBLISH, ADD, COUNTER, etc., too. It is not even specific to iTIP, but rather a general issue in iCalendar. The reference to REQUEST should at least be deleted.

4.6 Journal Examples

101. Sec. 4.6 (p. 106): The example is missing the REQUIRED DTSTAMP property.

4.7 Other Examples

102. Sec. 4.7.1 (p.107): In the describing text the UID value is line-broken, which should not happen.
103. Sec. 4.7.1 (p.107): While rfc2445bis allows spaces in the UID, it's not clear to me whether those spaces are actually part of the UID. In particular, are the following two UID property lines equivalent?

UID:guid-1-1234@example.com

UID: guid-1-1234@example.com

rfc2445bis defines the UID in Sec. 3.8.4.7 to be of text type, which allows spaces, but is quiet on how UIDs should be compared. Reading rfc2445bis strictly, everything after the colon is part of the UID, so that the two lines above are NOT equivalent. The space should probably be deleted from this example, but rfc2445bis should also clarify this case whether spaces are allowed in UIDs and whether leading/trailing spaces should be ignored.

104. Sec. 4.7.2 (p.107): The second sentence speaks about the case when »an "Organizer" sends a request«, which is an unfortunate wording, because it applies not only to the REQUEST method, but to any iTIP message sent. I would replace »a request« by »an iTIP message«.
105. Sec. 4.7.2 (p.107): In the explanation for case (1), the case where the SEQUENCE number was incremented by 1 is not a problem and does not mean that the Attendee missed a message or is receiving an out-of-sequence message. Rather is the default case when the Organizer modifies a calendar component: He will increase the SEQUENCE number by 1 and send out a new REQUEST to the Attendees. Thus this case should be not be covered by this case. If the CU receives a REQUEST where the SEQUENCE number is 1 larger than his current version and the RECURRENCE-ID can not be found in his calendar component, that's a worse case, since apparently he received all changes, which do, however, not contain that specific recurrence instance. Only for the CANCEL method the problem arises as described in case (2).
106. Sec. 4.7.2 (p.108): In the explanation for case (3), I think the Attendee SHOULD/MUST send a response with the status code »2.8; Success, repeating event ignored. Scheduled as a single component«. If he should not send such a reply (since he has probably already sent it when he received the original REQUEST that sets the recurrence), it should be mentioned here explicitly.
107. Sec. 4.7.2 (p.108): The table and the example cover only case (2), so it should either be moved to the text for (2) or the text above the table should explicitly mention that it covers case (2).

5 Application Protocol Fallbacks

108. Between an »Ignore« and some following text in all the tables, the punctuation should be consistent. In some places there is a »,« after the »Ignore«, in some places a period ».« and in some places the text continues after only a space » «.
109. Sec. 5.1.1 (p. 110): If a CUA does not support recurrences, it cannot handle ADD by default. There should be a fallback for this case, probably to insert the ADDED event as a separate component and replying with status code »2.8; Success, repeating event ignored. Scheduled as a single component«.
110. Sec. 5.1.1 (p.110): PRODID and VERSION are REQUIRED for rfc2445bis support, which rfc2446bis relies upon. Still, I agree it does not hurt if a CUA ignores PRODID and/or VERSION, so this can be left unchanged, too. I just wanted to point out that it's already a requirement for rfc2445bis.
111. Sec. 5.1.1 (p.111), Sec. 5.1.3 (p.113) and Sec. 5.1.4 (p.116): In the ATTENDEE properties, remove »EVENT-«, »VTODO - « and »JOURNAL - «, since there is no method "EVENT-REQUEST" and the fact that this is for events is clear from the section header anyway. If you think it should be left in, it should at least be changed to »REQUEST for events« etc.
112. Sec. 5.1.1 (p.111) and Sec. 5.1.3 (p.114): In the ATTENDEE properties, change »is not implemented« to the correct »is implemented«.

113. Sec. 5.1.1 (p.111): In the fallbacks for other calendar methods, there is no requirement to reply with »Not Supported« if the ATTENDEE property is not supported. Why is it a requirement for events?
114. Sec. 5.1.1 (p.111): Why is DESCRIPTION required? If an event has a SUMMARY, one can safely ignore the DESCRIPTION. On the other hand, many events I have seen have only a SUMMARY (which can be ignored according to the table), but no DESCRIPTION. It would make sense to me to treat SUMMARY and DESCRIPTION similarly.
115. Sec. 5.1.1 (p.111): I think the ORGANIZER property can only be ignored for PUBLISH, but not for any of the other methods, since that's the address where required replies must be sent to.
116. Sec. 5.1.1 (p.111): In the fallback for RRULE, change "DTStart" to "DTSTART", since the formatting conventions at the beginning of the draft say that all property names are in capitals.
117. Sec. 5.1.2 (p.112): Shouldn't there be a fallback for the case where the ATTENDEE property is not supported?
118. Sec. 5.1.2 (p.112): Why is DURATION support required for VFREEBUSY, when it is not required for VEVENT and VTODO?
119. Sec. 5.1.3 (p.113) and Sec. 5.1.4 (p.116): RECURRENCE-ID should be changed to »Required if RRULE is implemented«, like it is for VEVENT.
120. Sec. 5.2.2 (p.117): This section talks only of delegation, but completely ignores that a REQUEST can be forwarded to another CU, who then replies to the Organizer. Additionally, the last sentence poses a possible privacy issue: If the CUA implements the first strategy lined out in the second-to-last sentence (»accept the reply«) and treats the REPLY as a REFRESH, it might automatically send the current version of an event to a person, who is not authorized to have that information.

6 Security Considerations

121. Sec. 6.1.5 (p.118): Why is this a "MAY" and not "may"?
122. Sec. 6.2 (p.118): Remove one of the duplicated »is subject«.
123. Sec. 6.2.2 (p.119): Procedure alarms were deprecated in rfc2445bis, so the sentence about their security issues can be deleted from rfc2446bis.

7 IANA Considerations

No comments.

8 Acknowledgments

No comments.

9 Normative References

No comments.

A Appendices

No comments.

Issues in rfc2445bis-08

1. Sec. "3.6.1. Event Component" (p.54): Either make the DTSTART optional in rfc2445bis again or make it REQUIRED in the REPLY, REFRESH and DECLINECOUNTER methods in rfc2446bis (currently, it MUST NOT be present for these methods).
2. Sec. "3.6.5. Time Zone Component" (p.71): The text above the example says that this is an example of a time zone using the RDATE property. The example, however, does not contain any RDATE.
3. Sec. "3.8.7.4. Sequence Number" (p.142): It should be clarified whether the SEQUENCE number is supposed to be the same for all different components with the same UID (but differing in their use of RECURRENCE-ID to identify particular instances of a recurring sequence) or whether they can have different SEQUENCE numbers (see also issue 9 above).
4. Sec. "3.3.10. Recurrence Rule" (p.40): As discussed in issue 93 for Sec. 4.4 above, the description of INTERVAL can possibly be misunderstood. The next sentence explains it better, but it also only uses an interval of 1. Furthermore, anyone who misunderstood the first, ambiguous sentence will not be corrected by that second sentence. I propose to change the first sentence to:

The INTERVAL rule part contains a positive integer representing at which intervals the recurrence rule repeats.

Furthermore, I would add one more sentence to that paragraph:

A value of "2" means for example every other day for a DAILY rule, etc.

This explains the use of INTERVAL much better than the default value of "1".

5. Sec. "3.8.4.7. Unique Identifier" (p.120f): As detailed in issue 103 above, it should be clarified how spaces in a UID property value should be handled. In particular, are they allowed at all? Are leading or trailing spaces to be ignored when looking for or comparing UIDs?