

# iCalendar Recurrence Problems and Recommendations

Published Report

## **Warning for drafts**

This document is not a CalConnect Standard. It is distributed for review and comment, and is subject to change without notice and may not be referred to as a Standard. Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

The Calendaring and Scheduling Consortium, Inc. 2006

:2006

© 2006 The Calendaring and Scheduling Consortium, Inc.

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from the address below.

The Calendaring and Scheduling Consortium, Inc.

4390 Chaffin Lane  
McKinleyville  
California 95519  
United States of America

[copyright@calconnect.org](mailto:copyright@calconnect.org)  
[www.calconnect.org](http://www.calconnect.org)

# Contents

Foreword.....	iv
Introduction.....	v
1. What are recurrences and why do we need them?.....	1
2. Recurrence Issues with iTIP.....	1
2.1. How should recurrence modifications be handled?.....	1
2.2. When should the ADD method be used?.....	1
2.3. How should the scheduling of an unbounded rule be solved?.....	1
2.4. Can unbounded rules be truncated?.....	2
2.5. How should calendar objects occurring during a daylight time change be handled?.....	2
2.6. If an event is set to repeat on the 31st of each month, what do you do in months with fewer than 31 days?.....	2
2.7. How do we create a rule that excludes weekends and holidays?.....	2
3. Recommendations for iTIP.....	2
3.1. Recurrence properties modifications.....	2
3.2. When should the ADD method be used?.....	3
3.3. How should the scheduling of an unbounded rule be solved?.....	3
3.4. Can unbounded rules be truncated?.....	3
3.5. How should calendar objects occurring during a daylight saving time change be handled?.....	3
3.6. If an event is set to repeat on the 31st of each month, what do you do for months with fewer than 31 days?.....	3
3.7. How do we create a rule that excludes weekends and holidays?.....	3
3.8. Implementations without RRULE support.....	4
4. Recurrence Issues with iCalendar.....	4
4.1. Usefulness vs. Cost of implementation of some properties.....	4
5. Recommendations for iCalendar.....	4
5.1. Usefulness vs. Cost of implementation of some properties.....	4
6. Best Practices.....	5
6.1. When should the SEQUENCE value change?.....	5
6.2.....	5
6.3.....	5
6.4.....	6
7. Conclusion.....	6
Appendix A (normative) RRULE Processing.....	8
Appendix B (normative) Recurrence FREQ Examples.....	9
Bibliography.....	12

:2006

## **Foreword**

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

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

## Introduction

This document contains information about recurrence implementation issues and recommendations on how to resolve these issues. We first explain what recurrences are, justify the need for recurrences, identify recurrence implementation problems and ambiguities, and then offer some guidelines and recommendations to solve these issues.



# iCalendar Recurrence Problems and Recommendations

## 1. What are recurrences and why do we need them?

The word “recurrence” in iCalendar refers to the ability of an event to repeat, i.e. schedule a meeting to occur every Wednesday for the next 3 weeks. This sort of scheduling can be represented in iCalendar using a recurrence rule. The purpose of using the recurrence properties is to have the ability to transmit many related event dates in a single transaction. For example, it is possible to create a recurring meeting that will repeat every day for 5 days in one transaction to the invitee list, instead of sending five transactions (one for each day) to the invitee list. Recurrence properties also offer synchronization from desktop to mobile device, for example, a way to quickly send information about many related event dates in a single transaction.

The receiver expands the rule to display the actual dates during processing.

## 2. Recurrence Issues with iTIP

There are some ambiguities in iCalendar that have led some implementers to interpret the same iCalendar stream in very different ways. This difference in interpretation has led to low interoperability of some of the features of recurrences.

### 2.1. How should recurrence modifications be handled?

#### 2.1.1.

There are interoperability problems when the sender is attempting to reschedule an entire event, but does not send recurrence properties and does not wish to replace the recipients set with the new set. The sender is trying to simply move the set from one time to another (1pm to 2pm), adjust duration, etc. A simple example would be a recurring meeting that is currently from 1pm to 2pm, that is being rescheduled to 4pm to 5pm, for all instances. Implementers might not want to lose the recipient’s changes, personal notes, and updates to the set by replacing their set with what’s received.

#### 2.1.2.

There are additional interoperability problems when a sender attempts to reschedule a recurrence set with a new rule, which should replace the existing rule. An example would be a recurring meeting that follows one rule pattern, say daily for 5 days, and is replaced with another rule pattern that may or may not match the original number of instance, say weekly for 3 weeks.

### 2.2. When should the ADD method be used?

#### 2.2.1.

There’s some ambiguity in iTIP involving when these properties: RRULE, EXRULE, RDATE, EXDATE and some others of the master instance should be sent to the recipient. It’s unclear if the master instance properties should be sent to the recipient during an ADD.

### 2.3. How should the scheduling of an unbounded rule be solved?

#### 2.3.1.

Since there is no limit to the set of recurrences, it’s unclear what is expected to happen when scheduling, e.g. how can an attendee verify that they are free or busy for each event in the unbounded (infinite) set?

## **2.4. Can unbounded rules be truncated?**

### **2.4.1.**

It's unclear what an implementer should do if unbounded rules do not fit the business case the application was written for, i.e. if the iCalendar stream contains an unbounded rule that repeats every second forever. If the implementer's application expands RRULEs upon receiving them, it's unclear if it's acceptable to truncate and what response, if any, should go back to the sender.

## **2.5. How should calendar objects occurring during a daylight time change be handled?**

### **2.5.1.**

It's unclear what should happen if an iCalendar object occurs during a daylight time change, like April 4, 2005 at 2am EST, what should happen?

## **2.6. If an event is set to repeat on the 31st of each month, what do you do in months with fewer than 31 days?**

### **2.6.1.**

Should the instance roll to the next day or be dropped?

## **2.7. How do we create a rule that excludes weekends and holidays?**

### **2.7.1.**

It's unclear how to describe the sender's idea of holidays (say US holidays) when the recipient could have a different holiday set (say Israeli holidays).

## **3. Recommendations for iTIP**

### **3.1. Recurrence properties modifications**

#### **3.1.1.**

To reschedule an event to a set of explicit dates/times but not replace the set with a new rule pattern, the sender should send a set of components with RECURRENCE-ID being set in each to handle the explicit override of the rule's date/time.

#### **3.1.2.**

If an RRULE is changed, then the entire iCalendar object should be resent in the reschedule. This tells the receiver that all exceptions should be removed and replaced with this new rule. The new RRULE properties are required in order for the recipient to know that they need to replace the set they have with the incoming set.

##### **3.1.2.1.**

In iTIP there is currently the possibility of adding an RRULE to an event using the ADD method, discussions about this should be brought to a public forum. See next topic for details on the ADD method.



## **3.2. When should the ADD method be used?**

### **3.2.1.**

The ADD method should only be used to add occurrences to an existing calendar object, the supplied VEVENT/VJOURNAL/VTODO component data must not contain these properties: RRULE, EXRULE, RDATE, EXDATE (and all other properties that can only be in the master instance). Currently iTIP does allow recurrence properties in an ADD method's data.

## **3.3. How should the scheduling of an unbounded rule be solved?**

### **3.3.1.**

A free busy request must always have a start and end date, only the occurrences of the unbounded rule that match the free busy request timerange should be expanded.

## **3.4. Can unbounded rules be truncated?**

### **3.4.1.**

The implementer can truncate an unbounded rule if unbounded rules do not fit the business case the application was written for. The implementer must however make sure that when that calendar object is exported, the original rule is preserved; a warning could also be issued indicating that the calendar object was truncated.

### **3.4.2.**

The application truncating an incoming calendar object could implement an internal mechanism that would send an iTIP refresh request to the organizer requesting additional occurrences as time goes by.

## **3.5. How should calendar objects occurring during a daylight saving time change be handled?**

### **3.5.1.**

An explicit instance specified using an RDATE with a PERIOD value should be used to fix the required duration for the event spanning the daylight saving time change.

## **3.6. If an event is set to repeat on the 31st of each month, what do you do for months with fewer than 31 days?**

### **3.6.1.**

The instance needs to be dropped as specified in [IETF RFC 2445, Page 44](#).

## **3.7. How do we create a rule that excludes weekends and holidays?**

### **3.7.1.**

Use BYDAY without specifying Saturday and Sunday and use EXDATES to remove holidays.

## 3.8. Implementations without RRULE support

### 3.8.1.

A new iTIP response could be created indicating to a sender that the recipient does not support RRULEs, the sender could then decide to “explode” the calendar object into RDATEs and send that RRULE-less calendar object. A link to RRULE processing reference table and FREQ example streams can be found at the end of this document.

## 4. Recurrence Issues with iCalendar

### 4.1. Usefulness vs. Cost of implementation of some properties

#### 4.1.1.

Some of the RRULE grammar is cumbersome and difficult to use.

#### 4.1.2.

Some of the RRULE grammar is not needed for human-interaction systems.

## 5. Recommendations for iCalendar

### 5.1. Usefulness vs. Cost of implementation of some properties

#### 5.1.1.

Further discussions on public forums are needed about the usefulness of the following properties and if there could be simpler ways to solve the use cases they were made for:

##### 5.1.1.1.

EXRULE — This recurrence property is cumbersome to use and the equivalent can be generated with a list of EXDATEs. This property could be removed for better interoperability.

##### 5.1.1.2.

BYSECOND — This recurrence rule part is not useful in a human-interaction system and since that is our target, not automated systems, this rule part should be removed for better interoperability.

##### 5.1.1.2.1.

How to handle seconds if they are received? If a client receives an RRULE with a DTSTART that has non-zero seconds, the client can ignore the seconds without having to round up, which might have pushed the time into the next hour or day.

##### 5.1.1.3.

BYSETPOS — This RRULE property is useful in that it has the “payday” use case, ie. last workday of the month, but can be complicated to implement. The sender could use RDATEs as well but could be a lengthy list if this goes on yearly, etc. It is better to send a list of RDATEs with exceptions already taken into account, and refresh this at appropriate intervals to extend the set. If that is recommended in the RFC, then this property could be removed. Recommend going to the CALSIFY list to see if this is deemed a workable solution.

#### 5.1.1.4.

Multiple EXRULEs and RRULEs — These properties combined are complicated to implement, are not supported by many implementers, so support for multiple EXRULEs and RRULEs should be removed from the iCalendar RFC and related memos.

## 6. Best Practices

### 6.1. When should the SEQUENCE value change?

#### 6.1.1.

The SEQUENCE value should be changed when the date, time, or duration of one or more instances, of the master instance, change. This refers to a scheduling time change, say a meeting that was from 1pm to 2pm is being rescheduled to 3pm to 4pm.

#### 6.1.2.

The SEQUENCE value should not be changed when the only properties that are changed are those not having to do with meeting date, time, or duration change. This refers to changing SUMMARY or LOCATION, for example. When changing these properties, it's best not to change the SEQUENCE value if a meeting time change is not also involved in the update. The SEQUENCE value is used to denote a date, time, or duration change; not a change in other properties.

#### 6.1.2.1.

#### EXAMPLE

Chair sends out a 3 day recurring meeting that repeats Monday through Wednesday. Chair later changes the LOCATION for all the instances, but did not change the date, time, or duration. The receiver will note the SEQUENCE value has not changed, and can simply apply the other properties sent to the recurrence set. If the sender had changed the SEQUENCE value, the receiver could believe this to mean a date, time, or duration change, and attempt to apply a reschedule to the set when one did not occur.

### 6.2.

If you want to reschedule the first instance, you'd send DTSTART and DTEND, RECURRENCE-ID, and UID. (iTIP)

### 6.3.

Reschedules occur in two different varieties (iTIP), rescheduling where the RECURRENCE-ID is supposed to be changed and when RECURRENCE-ID is not supposed to change.

#### 6.3.1.

Rescheduling of one or more instances where RECURRENCE-ID is not subject to change. In this case, recurrence properties are not sent, so the receiver is expected to keep the current recurrence set and simply reschedule what's already on the calendar to different dates/times.

:2006

### 6.3.1.1.

Single Instance example: Chair sends out a 5 day recurring meeting that repeats Monday through Friday. Chair later reschedules only Wednesday to a different time. The Wednesday instance should retain its original RECURRENCE-ID; this property should not be updated.

### 6.3.1.2.

Many instance example: Chair sends out a 5 day recurring meeting that repeats Monday through Friday. Chair later reschedules Wednesday and Thursday to a different time. The Wednesday and Thursday instances should retain their original RECURRENCE-ID; this property should not be updated.

### 6.3.2.

Rescheduling of the entire set, where RECURRENCE-ID is expected to change. In this case, recurrence properties are sent, so that the receiver is expected to replace the existing recurrence set with the incoming set.

#### 6.3.2.1.

#### EXAMPLE

Chair sends out a 5 day recurring meeting that repeats Monday through Friday. Chair later reschedules the entire event to follow a different rule pattern; this time weekly for 5 weeks on Monday. All of the instances should be replaced with new instances containing data from the master that was sent, with new RECURRENCE-ID properties generated for each.

### 6.3.3.

Why not always update RECURRENCE-ID? Since RECURRENCE-ID is our key to find that particular instance, it should not be changed unless the entire set is being replaced. The reason is best explained with an example: Chair sends out a 5 day recurring meeting, Monday thru Friday, from 9am to 10am. Chair later reschedules Wednesday to be from 1pm to 2pm. One of their recipients does not receive the reschedule for Wednesday. Chair reschedules Wednesday again this time from 3pm to 4pm. If the RECURRENCE-ID was changed during the 1-2pm reschedule, then the recipient will not be able process this reschedule or any subsequent reschedules or updates. That RECURRENCE-ID will never match any instances on their calendar.

### 6.4.

When processing a RANGE that is set to THISANDFUTURE for recurrences, the order in which components have been overridden must be used to define which instances are affected by THISANDFUTURE. For example, say an event has three instances A, B and C. If B is overridden with a component with a RANGE=THISANDFUTURE parameter, then both B and C will be affected by the change. However, if C were subsequently changed, the change to C would not incorporate the changes done in B. Alternatively, if C was overridden first, and then B overridden with THISANDFUTURE, then the changes in B would be incorporated into C. Note that this means that the overridden component for C is effectively not used.

## 7. Conclusion

In conclusion, this document has attempted to trim recurrences to a subset of features that are common to implementations in the market, offer real value in the end result product, and would be deemed require functionality to the end user. Additional modifications that could be discussed for the new drafts are:

- 1) Are multiple RRULEs and EXRULEs really useful, could we do without them?
- 2) Are EXRULEs really useful, could we do without them?
- 3) Removal of THISANDPRIOR, since THISANDPRIOR always refers to a finite number of occurrences it could be done with exceptions.

## Appendix A (normative) RRULE Processing

A particular BYxxx rule part may expand or limit the set of date/times generated by the rule. The expand or limit behaviour is governed by the FREQ value used for the rule.

### EXAMPLE

RRULE:FREQ=MONTHLY;BYMONTH=1,3,5;BYDAY=MO,TU

The FREQ=MONTHLY value would match each of the twelve months in a year.

The BYMONTH=1,3,5 rule part limits the matching months to just the 1st, 3rd and 5th in a year.

The BYDAY=MO,TU rule part adds each Monday and Tuesday within the matching months to the recurrence set.

The table below shows the dependency of BYxxx rule part expand or limit behaviour on the FREQ value in the rule. When evaluating a rule, each BYxxx rule part must be evaluated in the order it appears in the table (i.e. BYMONTH evaluated before BYWEEKNO), irrespective of the expand or limit behaviour.

BYDAY has some special behaviour depending on the FREQ value and this is described in separate notes below the table.

	SECONDLY	MINUTELY	HOURLY	DAILY	WEEKLY	MONTHLY	YEARLY
BYMONTH	Limit	Limit	Limit	Limit	Limit	Limit	Expand
BYWEEKNO	Limit	Limit	Limit	Limit	Limit	N/A	Expand
BYYEARDAY	N/A	N/A	N/A	N/A	N/A	N/A	Expand
BYMONTHDAY	Limit	Limit	Limit	Limit	N/A	Expand	Expand
BYDAY	Limit	Limit	Limit	Limit	a	b	c
BYHOUR	Limit	Limit	Limit	Expand	Expand	Expand	Expand
BYMINUTE	Limit	Limit	Expand	Expand	Expand	Expand	Expand
BYSECOND	Limit	Expand	Expand	Expand	Expand	Expand	Expand
BYSETPOS	Limit	Limit	Limit	Limit	Limit	Limit	Limit

<sup>a</sup> Special expand for WEEKLY. A BYDAY rule part cannot have a numeric value in a FREQ=WEEKLY rule (i.e. 'MO', 'TU' etc is allowed, but '1MO', '2TU' is not allowed).

<sup>b</sup> Limit if BYYEARDAY or BYMONTHDAY is present, otherwise special expand for MONTHLY. The numeric value in a BYDAY rule part in a FREQ=MONTHLY rule corresponds to an offset with the month.

<sup>c</sup> Limit if BYYEARDAY or BYMONTHDAY is present, otherwise special expand for WEEKLY if BYWEEKNO present, otherwise special expand for MONTHLY if BYMONTH present, otherwise special expand for YEARLY. A BYDAY rule part cannot have a numeric value in a FREQ=YEARLY rule (i.e. 'MO', 'TU' etc is allowed, but '1MO', '2TU' is not allowed), if BYWEEKNO is specified. The numeric value in a BYDAY rule part in a FREQ=YEARLY rule with a BYMONTH present corresponds to an offset with the month. The numeric value in a BYDAY rule part in a FREQ=YEARLY rule without BYWEEKNO or BYMONTH present corresponds to an offset within the year.

## Appendix B (normative) Recurrence FREQ Examples

### EXAMPLE 1 — Monthly by Day: Every Month on the 1st Monday for 5 months

```

BEGIN:VCALENDAR
X-LOTUS-CHARSET:UTF-8
VERSION:2.0
PROPID:-//Lotus Development Corporation//NONSGML Notes 7.0//EN
METHOD:REQUEST
BEGIN:VTIMEZONE
TZID:Eastern
BEGIN:STANDARD
DTSTART:19501029T020000
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=-1SU;BYMONTH=10
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19500402T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=1SU;BYMONTH=4
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTART;TZID="Eastern":20100802T100000
DTEND;TZID="Eastern":20100802T110000
TRANSP:OPAQUE
RRULE:FREQ=MONTHLY;COUNT=5;BYDAY=1MO
DTSTAMP:20050822T203750Z
SEQUENCE:0
ATTENDEE;ROLE=CHAIR;PARTSTAT=ACCEPTED;CN="Chris Stoner/Westford/IBM"
;RSVP=FALSE:mailto:Chris_Stoner@notesdev.ibm.com
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=NEEDS-ACTION;RSVP=TRUE
:mailto:green_jellybean@mac.com
CLASS:PUBLIC
DESCRIPTION;ALTREP="CID:<FFFF__=0ABBF6DFE2AA148f9e8a93df938690918c0AB@>"
:Every Month on the 1st Monday for 5 months
SUMMARY:FREQ =BYMONTH Limit Example
ORGANIZER;CN="Chris Stoner/Westford/IBM"
:mailto:Chris_Stoner/Westford/IBM@
UID:A8398E9A6BBE453F8525706500712C84-Lotus_Notes_Generated
END:VEVENT
END:VCALENDAR

```

### EXAMPLE 2 — Monthly By Day Expand Example: Every month on the 6th day

```

BEGIN:VCALENDAR
X-LOTUS-CHARSET:UTF-8
VERSION:2.0
PROPID:-//Lotus Development Corporation//NONSGML Notes 7.0//EN
METHOD:REQUEST
BEGIN:VTIMEZONE
TZID:Eastern
BEGIN:STANDARD
DTSTART:19501029T020000
TZOFFSETFROM:-0400
TZOFFSETTO:-0500

```

:2006

```
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=-1SU;BYMONTH=10
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19500402T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=1SU;BYMONTH=4
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTART;TZID="Eastern":20100906T100000
DTEND;TZID="Eastern":20100906T110000
TRANSP:OPAQUE
RRULE:FREQ=MONTHLY;COUNT=5;BYMONTHDAY=6
DTSTAMP:20050822T204655Z
SEQUENCE:0
ATTENDEE;ROLE=CHAIR;PARTSTAT=ACCEPTED;CN="Chris Stoner/Westford/IBM"
;RSVP=FALSE:mailto:Chris_Stoner@notesdev.ibm.com
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=NEEDS-ACTION;RSVP=TRUE
:mailto:gb@foo.com
CLASS:PUBLIC
DESCRIPTION;ALTREP="CID:<FFFF__=0ABBF6DFE19B548f9e8a93df938690918c0AB@>":
Every month on the 6th day
SUMMARY:Every month on the 6th day
ORGANIZER;CN="Chris Stoner/Westford/IBM"
:mailto:Chris_Stoner/Westford/IBM@
UID:9DF697E752368AE78525706500721DC4-Lotus_Notes_Generated
END:VEVENT
END:VCALENDAR
```

### EXAMPLE 3 — Daily every other day for 5 days

```
BEGIN:VCALENDAR
X-LOTUS-CHARSET:UTF-8
VERSION:2.0
PROPID:-//Lotus Development Corporation//NONSGML Notes 7.0//EN
METHOD:REQUEST
BEGIN:VTIMEZONE
TZID:Eastern
BEGIN:STANDARD
DTSTART:19501029T020000
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=-1SU;BYMONTH=10
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19500402T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=1SU;BYMONTH=4
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTART;TZID="Eastern":20100906T100000
DTEND;TZID="Eastern":20100906T110000
TRANSP:OPAQUE
RRULE:FREQ=DAILY;COUNT=3;INTERVAL=2
DTSTAMP:20050822T204514Z
SEQUENCE:0
ATTENDEE;ROLE=CHAIR;PARTSTAT=ACCEPTED;CN="Chris Stoner/Westford/IBM"
;RSVP=FALSE:mailto:Chris_Stoner@notesdev.ibm.com
```



```

ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=NEEDS-ACTION;RSVP=TRUE
:mailto:green_jellybean@foo.com
CLASS:PUBLIC
SUMMARY:Daily every other day
ORGANIZER;CN="Chris Stoner/Westford/IBM"
:mailto:Chris_Stoner/Westford/IBM@
UID:FACE7CA46BE8B3F3852570650071F380-Lotus_Notes_Generated
END:VEVENT
END:VCALENDAR

```

#### EXAMPLE 4 — Daily every day for 5 days

```

BEGIN:VCALENDAR
X-LOTUS-CHARSET:UTF-8
VERSION:2.0
PROPID:-//Lotus Development Corporation//NONSGML Notes 7.0//EN
METHOD:REQUEST
BEGIN:VTIMEZONE
TZID:Eastern
BEGIN:STANDARD
DTSTART:19501029T020000
TZOFFSETFROM:-0400
TZOFFSETTO:-0500
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=-1SU;BYMONTH=10
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:19500402T020000
TZOFFSETFROM:-0500
TZOFFSETTO:-0400
RRULE:FREQ=YEARLY;BYMINUTE=0;BYHOUR=2;BYDAY=1SU;BYMONTH=4
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTART;TZID="Eastern":20100906T100000
DTEND;TZID="Eastern":20100906T110000
TRANSP:OPAQUE
RRULE:FREQ=DAILY;COUNT=5
DTSTAMP:20050822T204315Z
SEQUENCE:0
ATTENDEE;ROLE=CHAIR;PARTSTAT=ACCEPTED;CN="Chris Stoner/Westford/IBM"
;RSVP=FALSE:mailto:Chris_Stoner@notesdev.ibm.com
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=NEEDS-ACTION;RSVP=TRUE
:mailto:green_jellybean@foo.com
CLASS:PUBLIC
SUMMARY:Daily every day for 5 days
ORGANIZER;CN="Chris Stoner/Westford/IBM"
:mailto:Chris_Stoner/Westford/IBM@
UID:217C3BD27E9FDF9E852570650071C753-Lotus_Notes_Generated
END:VEVENT
END:VCALENDAR

```

## Bibliography

- [1] IETF RFC 2445, F. DAWSON and D. STENERSON. *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. 1998. RFC Publisher. <https://www.rfc-editor.org/info/rfc2445>.
- [2] IETF RFC 2446, S. SILVERBERG, S. MANSOUR, F. DAWSON and R. HOPSON. *iCalendar Transport-Independent Interoperability Protocol (iTIP) Scheduling Events, BusyTime, To-dos and Journal Entries*. 1998. RFC Publisher. <https://www.rfc-editor.org/info/rfc2446>.