

Calendaring extensions	N. Jenkins
Internet-Draft	R. Stepanek
Intended status: Standards Track	FastMail
Expires: January 30, 2021	M. Douglass
	Bedework
	July 29, 2020

JSCalendar: Converting from and to iCalendar

draft-ietf-calext-jscalendar-icalendar-02

Abstract

This document provides the required methods for converting JSCalendar from and to iCalendar.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 30, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. **Introduction**
 - 1.1. **Motivation**
 - 1.2. **Scope and caveats**
 - 1.3. **Notational Conventions**
2. **New iCalendar parameters**
 - 2.1. **FRACTIONAL parameter**
3. **iCalendar pre-processing**
4. **Translating iCalendar components to JSCalendar**
 - 4.1. **VALARM**
 - 4.2. **VCALENDAR**
 - 4.3. **VEVENT**
 - 4.4. **VTIMEZONE, STANDARD, DAYLIGHT**
 - 4.5. **VTODO**
5. **Translating iCalendar properties to JSCalendar**
 - 5.1. **ATTACH**
 - 5.2. **ATTENDEE**
 - 5.3. **CALSCALE**
 - 5.4. **CATEGORIES**
 - 5.5. **COMMENT**
 - 5.6. **COMPLETED**
 - 5.7. **CONTACT**
 - 5.8. **CREATED**
 - 5.9. **DESCRIPTION**
 - 5.10. **DTEND, DTSTART, DUE, DURATION**
 - 5.11. **EXDATE**
 - 5.12. **DTSTAMP and LAST-MODIFIED**
 - 5.13. **PERCENT-COMPLETE**
 - 5.14. **PRIORITY**
 - 5.15. **PRODID**
 - 5.16. **RELATED-TO**
 - 5.17. **RESOURCES**
 - 5.18. **RDATE**
 - 5.19. **RRULE**
 - 5.20. **SEQUENCE**
 - 5.21. **STATUS**
 - 5.22. **SUMMARY**
 - 5.23. **TRANSP**
 - 5.24. **UID**
 - 5.25. **URL**
6. **Translating iCalendar: Further examples**
 - 6.1. **Recurring event with ATTACH**
 - 6.2. **Simple event with CONTACT**
 - 6.3. **Simple event with RESOURCES**
 - 6.4. **Recurring event. Attendees only in overrides**
7. **Translating JSCalendar properties to iCalendar components**
 - 7.1. **jsevent**
 - 7.2. **jsgroup**
 - 7.3. **jstask**
 - 7.4. **timezones**
8. **Some notes**
 - 8.1. **Alerts**

- 8.2. [Preserving icalendar](#)
- 8.3. [JSCalendar changes](#)
- 9. [JSEvent](#)
- 10. [JSTask](#)
- 11. [JSGroup](#)
- 12. [Common properties](#)
 - 12.1. [Time properties and types](#)
 - 12.2. [Locations](#)
 - 12.3. [Participants](#)
- 13. [Custom properties](#)
- 14. [Security Considerations](#)
- 15. [IANA Considerations](#)
- 16. [Acknowledgments](#)
- 17. [References](#)
 - 17.1. [Normative References](#)
 - 17.2. [Informative References](#)
- [Authors' Addresses](#)

1. Introduction

1.1. Motivation

The JSCalendar [\[draft-ietf-calext-jscalendar\]](#) data format is used to represent calendar data, and is meant as an alternative to the widely deployed iCalendar [\[RFC5545\]](#) data format.

While new calendaring services and applications might use JSCalendar as their main data format to exchange calendaring data, they are likely to interoperate with services and clients that just support iCalendar. Similarly, existing calendaring data is stored in iCalendar format in databases and other calendar stores, and providers and users might want to represent this data also in JSCalendar. Lastly, there is a requirement to preserve custom iCalendar properties that have no equivalent in JSCalendar when converting between these formats.

To support these use cases, this document provides the required approach when converting JSCalendar data from and to iCalendar.

1.2. Scope and caveats

JSCalendar and iCalendar have a lot of semantics in common, but they are not interchangeable formats:

- JSCalendar contains a richer data model to express calendar information such as event locations and participants. While future iCalendar extensions may allow a direct mapping, for now there may be no representation directly in iCalendar of some properties. These values may have to be extracted from a full copy of the iCalendar format provided as a property in the JSCalendar data.
- iCalendar may contain arbitrary, non-standardised data with custom properties/attributes. These will be translated using the same approach as jCal.
- iCalendar has some obsolete features that have been removed from JSCalendar due to not being useful and/or supported in the real world (e.g. custom email alerts to send to random people). Translating these may lose some of the original fidelity.
- Implementations may use a custom property to store data that could not be mapped directly in either direction in the original or a custom format, however this is not interoperable.

Accordingly, this document defines a canonical translation between iCalendar and JSCalendar, and implementations **MUST** follow the approaches specified here when iCalendar data is represented in JSCalendar and vice-versa.

1.3. Notational Conventions

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 [\[RFC2119\]](#).

2. New iCalendar parameters

2.1. FRACTIONAL parameter

This parameter is defined by the following notation:

```
fractional-param = DATE-TIME or DURATION
```

```
DTSTART;FRACTIONAL=20190605T133015.03:20190605T133015
```

Parameter name:

FRACTIONAL

Purpose:

This parameter is used to contain a value with fractional seconds for time values and durations. FRACTIONAL MUST NOT be used in date-time calculations or comparisons in iCalendar. It is meant to preserve time precision on time values and duration with sub-second precision, without increasing the time value range within iCalendar.

Description:

This parameter MAY be specified on properties of type DATE-TIME or DURATION. It MUST be a valid iCalendar DATE-TIME or DURATION value with the addition of fractional seconds. The value MUST NOT be negative for durations but MAY be negative for alarm triggers. iCalendar implementations SHOULD ignore this parameter in date time arithmetic. Implementations MUST ignore presence of the FRACTIONAL parameter on RECURRENCE-ID properties when determining recurrence overrides.

Applications receiving a property with a FRACTIONAL parameter MUST ensure its value is consistent with the value of the property. The property value must match:

- a positive FRACTIONAL value rounded up to the next non-fractional second or
- a negative FRACTIONAL value rounded down the next non-fractional second

If the values do not match the the application MUST assume that the property value has been updated by an application that is unaware of the FRACTIONAL parameter. The parameter should be ignored in this case.

Format Definition:

Example:

3. iCalendar pre-processing

iCalendar uses a line-folding mechanism to limit lines of data to a maximum line length (typically 75 octets) to ensure the maximum likelihood of preserving data integrity as it is transported via various means (e.g., email) -- see Section 3.1 of [\[RFC5545\]](#).

iCalendar data uses an "escape" character sequence for text values and property parameter values. See Sections 3.1 and 3.3 of [\[RFC5545\]](#) as well as [\[RFC6868\]](#).

There is a subtle difference in the number representations between JSON and iCalendar. While in iCalendar, a number may have leading zeros, as well as a leading plus sign; this is not the case in JSON. Numbers

should be represented in whatever way needed for the underlying format.

When converting from iCalendar to JSCalendar: First, iCalendar lines MUST be unfolded. Afterwards, any iCalendar escaping MUST be unescaped. Finally, JSON escaping, as described in Section 7 of [\[RFC7159\]](#), MUST be applied. The reverse order applies when converting from JSCalendar to iCalendar, which is further described in Section ?.

iCalendar uses a base64 encoding for binary data. However, it does not restrict the encoding from being applied to non-binary value types. So, the following rules are applied when processing a property with the "ENCODING" property parameter set to "BASE64":

- If the property value type is "BINARY", the base64 encoding MUST be preserved.
- If the value type is not "BINARY", the "ENCODING" property parameter MUST be removed, and the value MUST be base64 decoded.

When base64 encoding is used, it MUST conform to Section 4 of [\[RFC4648\]](#), which is the base64 method used in [\[RFC5545\]](#).

One key difference in the formatting of values used in iCalendar and JSCalendar is that in JSCalendar, the specification uses date/time values aligned with the extended format of [\[ISO.8601.2004\]](#), which is more commonly used in Internet applications that make use of the JSON format. The sections of this document describing the various date and time formats contain more information on the use of the complete representation, reduced accuracy, or truncated representation.

4. Translating iCalendar components to JSCalendar

This section is an alphabetic list of all [\[RFC5545\]](#) components and how they are mapped to JSCalendar.

At present VFREEBUSY and VJOURNAL are not mapped in jscalendar.

4.1. VALARM

```
BEGIN: VEVENT
```

```
...
```

```
BEGIN: VALARM
```

```
...
```

```
END: VALARM
```

```
BEGIN: VALARM
```

```
...
```

```
END: VALARM
```

```
END: VEVENT
```

maps to

```
{
```

```
  "@type": "jsevent",
```

```
  ...
```

```
  "alerts": {
```

```
    "6ICJGbGlnaHRSZXNlcnZhdGlvb": {
```

```
      "@type": "Alert",
```

```
      ...
```

```
    },
```

```
    "kljogllJYSjM0UCIsCiAglCAgl": {
```

```
      "@type": "Alert",
```

```
      ...
```

```
}  
}  
}
```

A [RFC5545] VALARM component is mapped to a member of a JSCalendar "alerts" object with a type of "Alert" and an arbitrary id.

The [RFC5545] VALARM has a number of problems which are not carried over into JSCalendar. Clients tend to choose how, and in some cases when to notify the user.

For example, if the user has a smart-watch they may get tapped on the wrist. The method of notification may depend on which device is being used and the context, for example, a meeting or driving.

Also, many clients are taking into consideration the travel time and notifying the user earlier if it seems necessary.

Specifying that a client should send emails to all attendees is both annoying and dangerous. Attendees have their own preferences for how and when they should be notified.

Accordingly, the specification only allows for "display" and "email" actions and - other than specifying when - does not allow much else. Clients and/or servers will generally use the associated event or task title as identification. User preferences generally indicate what actions they prefer.

An [RFC5545] ACTION property can take the defined values "AUDIO" / "DISPLAY" / "EMAIL" whereas the JSCalendar "action" property only supports "display" and "email".

An "AUDIO" alarm SHOULD be mapped to a "display" alert. Any attachment MUST be ignored.

```
BEGIN:VEVENT  
...  
BEGIN:VALARM  
TRIGGER;VALUE=DATE-TIME:19970317T133000Z  
REPEAT:4  
DURATION:PT15M  
ACTION:AUDIO  
ATTACH;FMTTYPE=audio/basic:ftp://example.com/pub/  
sounds/bell-01.aud  
END:VALARM  
BEGIN:VALARM  
TRIGGER:-PT30M  
REPEAT:2  
DURATION:PT15M  
ACTION:DISPLAY  
DESCRIPTION:Breakfast meeting with executive\  
team at 8:30 AM EST.  
END:VALARM  
BEGIN:VALARM  
TRIGGER;RELATED=END:-P2D  
ACTION:EMAIL  
ATTENDEE:mailto:john_doe@example.com  
SUMMARY:*** REMINDER: SEND AGENDA FOR WEEKLY STAFF MEETING ***  
DESCRIPTION:A draft agenda needs to be sent out to the attendees  
to the weekly managers meeting (MGR-LIST). Attached is a  
pointer the document template for the agenda file.  
ATTACH;FMTTYPE=application/msword:http://example.com/  
templates/agenda.doc
```

```
END:VALARM
END:VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
  "alerts": {
    "6lCJGbGlnaHRSZXNlcnZhdGlvb": {
      "@type": "Alert",
      "action": "display",
      "trigger": {
        "@type": "AbsoluteTrigger",
        "when": "19970317T133000Z"
      }
    },
    "kljogllJYSjM0UCIsCiAglCAgl": {
      "@type": "Alert",
      "action": "display",
      "trigger": {
        "@type": "OffsetTrigger",
        "offset": "-PT30M"
      }
    }
  },
  "odHRwOi8vc2NoZW1hLm9yZy9SZ": {
    "@type": "Alert",
    "action": "email",
    "trigger": {
      "@type": "OffsetTrigger",
      "offset": "-P2D",
      "relativeTo": "end"
    }
  }
}
}
```

The [RFC5545](#) example VALARMS will be mapped as follows, assuming they are all in the same event:

Note that the ATTACH, ATTENDEE, DESCRIPTION, DURATION, REPEAT and SUMMARY properties have been dropped.

4.2. VCALENDAR

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
...
END: VCALENDAR
```

maps to

```
{
  "@type": "jsgroup",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
```

```
...
}
```

A [\[RFC5545\]](#) VCALENDAR component may be mapped to a JSCalendar object with a type of "jsgroup".

Note that a single JSEvent or JSTask MAY be converted without a surrounding JSGroup if the VCALENDAR component only contains PRODID and CALSCALE properties. In this case the prddid can go in the JSEvent or JSTask. The CALSCALE property is dropped - there is no equivalence in JSCalendar.

4.3. VEVENT

```
BEGIN: VEVENT
```

```
...
```

```
END: VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
}
```

A [\[RFC5545\]](#) VEVENT component is mapped to a JSCalendar object with a type of "jsevent".

4.4. VTIMEZONE, STANDARD, DAYLIGHT

A [\[RFC5545\]](#) VTIMEZONE component is mapped to a member of a JSCalendar "timezones" object with a type of "TimeZone" and an id which follows the restrictions specified.

The STANDARD and DAYLIGHT components map to JSCalendar TimeZoneRule objects as members of the

```
BEGIN: VTIMEZONE
```

```
TZID: Example/Somewhere
```

```
...
```

```
END: VTIMEZONE
```

```
BEGIN: VTIMEZONE
```

```
TZID: Example/Somewhere-else
```

```
...
```

```
END: VTIMEZONE
```

```
BEGIN: VEVENT
```

```
...
```

```
END: VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
  "timezones": {
    "/Example/Somewhere": {
      "@type": "TimeZone",
      ...
    },
    "/Example/Somewhere-else": {
      "@type": "TimeZone",
```

```
...
}
}
}
```

Note that

- There is no current approach for defining standalone sets of timezones.
- Timezones defined in the IANA timezone database SHOULD NOT be redefined in the object. Only custom timezones will be defined.

4.5. VTODO

```
BEGIN: VTODO
```

```
...
```

```
END: VTODO
```

maps to

```
{
  "@type": "jstask",
  ...
}
```

A [\[RFC5545\]](#) VTODO component is mapped to a JSCalendar object with a type of "jstask".

5. Translating iCalendar properties to JSCalendar

This section is an alphabetic list of all [\[RFC5545\]](#) properties and how they are mapped to JSCalendar.

5.1. ATTACH

A [\[RFC5545\]](#) ATTACH allows for two types of attachment:

- A uri value
- A binary value

Both map to a JSCalendar "link" object with a "rel" of "enclosure" and the "href" set to the value of the property.

If the FMTTYPE parameter is set then add a JSCalendar "contentType" property to the link object.

For a binary value use a base64 data uri.

For an example of a recurring event with ATTACH see [Section 6.1](#)

5.2. ATTENDEE

A [\[RFC5545\]](#) ATTENDEE maps to the JSCalendar "participant" property with a JSCalendar "role" of "attendee". The value for role should always be set.

In the simplest case a JSCalendar "participant" property will be created and added to the JSCalendar "participants" property.

The value of the ATTENDEE property is used to add an "imip" method to the JSCalendar "sendTo" property. The value of the entry will be the ATTENDEE property value.

```

...
ATTENDEE:mailto:user01@example.org
...

maps to

{
...
  "participants": {
    "be450b70-9bf7-4f6e-8f65-971ede566ce3": {
      "@type": "Participant",
      "sendTo": {
        "imip": "user01@example.org"
      },
    },
  },
...
}

```

For example:

The attendee parameters are mapped to JSCalendar "participant" properties as follows:

CN:

The value of the CN parameter is used to set the JSCalendar "name" property.

CUTYPE:

This maps on to the JSCalendar "kind" property as follows:

```

INDIVIDUAL
  "individual"
GROUP
  "group"
RESOURCE
  "resource"
ROOM
  "location"
UNKNOWN
  No value

```

Any other value should be converted to lower case and assigned to the JSCalendar "kind" property.

DELEGATED-FROM:

Split the value at any commas and add each resulting element to the JSCalendar "delegatedFrom" property

DELEGATED-TO:

Split the value at any commas and add each resulting element to the JSCalendar "delegatedFrom" property

DIR:

If non-null look in the JSCalendar "links" property for a JSCalendar "link" property with an href with the same value as the DIR parameter. You may need to search the current override and the master.

If none is found create a new one with the JSCalendar "href" property set to the value of the DIR parameter and the JSCalendar "rel" property set to "alternate"

Add the id of the located or created link to the JSCalendar "linkIds" property of the participant.

LANG:

set the JSCalendar "language" property to the value of the LANG parameter.

MEMBER:

If this is set there should be a corresponding ATTENDEE object with a value equal to the value of the member parameter. If not it is appropriate to skip this parameter.

If there is a corresponding ATTENDEE then there should be a corresponding JSCalendar "participant" property. This suggests that CUTYPE=GROUP ATTENDEE properties should be processed ahead of the others.

Locate the JSCalendar "participant" property for the group. This may be in the current override or in the master. Add the id to the current participants JSCalendar "memberOf" property.

PARTSTAT:

If the PARTSTAT parameter is set and is not "NEEDS-ACTION" then set the JSCalendar "participationStatus" property to the lower-cased value of the PARTSTAT.

ROLE:

This is mapped to the JSCalendar "roles" property as follows:

CHAIR	"attendee" and "chair"
REQ-PARTICIPANT	"attendee"
OPT-PARTICIPANT	"attendee" and "optional"
NON-PARTICIPANT	"informational"

Any other value should be converted to lower case and added to the JSCalendar "roles" property.

RSVP:

If the value of the RSVP parameter is TRUE set the JSCalendar "expectReply" property to "true" otherwise omit it.

SCHEDULE-AGENT:

If the value is "CLIENT" (ignoring case) set the JSCalendar "scheduleAgent" property to "client" otherwise omit it.

SCHEDULE-FORCE-SEND:

Set the JSCalendar "scheduleForceSend" property to the lower-cased value of the [RFC6638](#) SCHEDULE-FORCE-SEND parameter.

SCHEDULE-STATUS:

Split the value at any commas and add each resulting element to the JSCalendar "scheduleStatus" property.

SENT-BY:

The value of the SENT-BY parameter is used to set the JSCalendar "invitedBy" property.

5.3. CALSCALE

A [RFC5545](#) CALSCALE has no equivalence in JSCalendar. It is ignored.

5.4. CATEGORIES

```
...
CATEGORIES:APPOINTMENT,EDUCATION
CATEGORIES:MEETING
```

```

...
maps to

...
"keywords": {
  "APPOINTMENT": true,
  "EDUCATION": true,
  "MEETING": true
},
...

```

These map on to the JSCalendar "keywords" property with each category being the key to an entry.

5.5. COMMENT

There is no direct mapping for this property which may appear multiple times in [\[RFC5545\]](#).

For a scheduling reply it is presumably a message by the participant so the value or values should be used to set the JSCalendar "participantComment" property.

5.6. COMPLETED

```

...
COMPLETED: "20101010T101010Z"
...

maps to

...
"progressUpdated": "2010-10-10T10:10:10Z",
"progress": "completed",
...

```

Set the JSCalendar "progress" property to "completed" and the "progressUpdated" property to the reformatted date/time.

5.7. CONTACT

The CONTACT property is mapped on to a participant object with a "roles" property of "contact" and an "order" property of 1 (one). This defines the participant as a primary contact.

Mapping parameters:

ALTREP

Use the same process as for the ATTENDEE DIR parameter: create a link property with the "rel" property set to "alternate" and the "href" property set to the value of the ALTREP parameter. Then add the id of the link to the participants "linkIds" property.

LANG

Set the participants "language" property.

For an example see [Section 6.2](#)

5.8. CREATED

```
BEGIN:VEVENT
...
CREATED:19960329T133000Z
...
END:VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
  "created": "1996-03-29T13:30"00Z",
  ...
}
```

The CREATED property is mapped on to a "created" property with a json formatted form of the date.
Example:

5.9. DESCRIPTION

Copy the value, preprocessed according to [Section 3](#) into the "description" property.

Mapping parameters:

ALTREP

No mapping.

LANG

Use the "locale" property.

```
BEGIN:VEVENT
...
DESCRIPTION:We are having a meeting all this week at 12 pm fo
r one hour\, with an additional meeting on the first day 2 h
ours long.\nPlease bring your own lunch for the 12 pm meetin
gs.
...
END:VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
  "description": // Note: comments and string concatenation are not
    // allowed per the JSON specification and is used here
    // to avoid long lines.
    "We are having a meeting all this week at 12 pm for one " +
    "hour, with an additional meeting on the first day 2 " +
    "hours long.\nPlease bring your own lunch for the 12 pm " +
    "meetings.",
  ...
}
```

Example:

5.10. DTEND, DTSTART, DUE, DURATION

If the DTSTART is a DATE only property then add the JSCalendar showWithoutTime property with the value set to "true". The JSCalendar "start" property is set with zero time values.

If the DTSTART has a TZID parameter then set the JSCalendar "timeZone" property to the value of TZID.

If the DTSTART has a UTC value then set the JSCalendar "timeZone" property to the value "Etc/UTC". The JSCalendar "start" property is set without any UTC indicator.

JSCalendar has no equivalent to DTEND. If the component has a DTEND then calculate a value for "DURATION" from that property and DTSTART and proceed as below.

If the DTEND has a TZID parameter with a value that differs from the DTSTART TZID parameter then a "location" object should be created with a "relativeTo" property set to "end" and a "timezone" property set to the value of the "TZID" parameter.

Note that a task is not required to have a DTSTART so the JSCalendar "timezone" property needs to be set from the DUE property.

Convert a DURATION property to the JSCalendar duration.

```
BEGIN:VEVENT
...
DTSTART;TZID=America/New_York:20170315T150000
DTEND;TZID=America/New_York:20170315T160000
```

```
...
END:VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
  "start": "2017-03-15T15:00:00",
  "timeZone": "America/New_York",
  "duration": "PT1H"
  ...
}
```

Example - DTSTART and DTEND in same timezone:

```
BEGIN:VEVENT
...
DTSTART;TZID=America/New_York:20170315T150000
DTEND;TZID=America/LosAngeles:20170315T190000
```

```
...
END:VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
```

```

"start": "2017-03-15T15:00:00",
"timeZone": "America/New_York",
"duration": "PT7H"
...
"locations": {
  "lbGxlcil6IHsKICAgI": {
    "@type": "location",
    "relatedTo": "end",
    "timeZone": "America/Los_Angeles"
  }
}
}

```

Example - DTSTART and DTEND in different timezone:

```

BEGIN:VEVENT
...
DTSTART;VALUE=DATE:20210315
DTEND;VALUE=DATE:20210318
...
END:VEVENT

maps to

{
  "@type": "jsevent",
  ...
  "start": "2017-03-15T00:00:00",
  "duration": "P3D",
  "showWithoutTime": true,
  ...
}

```

Example - 3 day event:

5.11. EXDATE

Create a patch object with the recurrence id set from the EXDATE value. Add a single JSCalendar "excluded" property with the value set to true. There MUST NOT be any other properties set - other than "@type".

5.12. DTSTAMP and LAST-MODIFIED

The mapping depends on whether or not the component is a scheduling entity.

Not a scheduling entity:

The [\[RFC5545\]](#) DTSTAMP and LAST-MODIFIED properties have essentially the same meaning. If both are present use the value of the latest for the "updated" property. Otherwise set from whichever is present.

Is a scheduling entity:

DTSTAMP should be used to set the "ScheduleUpdated" property in the "participant" object for the attendee.

If present LAST-MODIFIED should be used to set the "updated" property - otherwise set it from the

DTSTAMP.

5.13. PERCENT-COMPLETE

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
METHOD:PUBLISH
BEGIN:VTODO
...
PERCENT-COMPLETE:39
END:VTODO
END: VCALENDAR
```

maps to

```
{
  "@type": "jstask",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
  ...
  "percentComplete": 39
}
```

For all methods other than REPLY (or no method), the PERCENT-COMPLETE applies to the VTODO as a whole. In this case it the value is used to set the JSCalendar "percentComplete" property in the task object.

PERCENT-COMPLETE in a REPLY is used to indicate the level of completeness of the ATTENDEE. There should only be a single ATTENDEE in the VTODO object.

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
METHOD:REPLY
BEGIN:VTODO
...
ATTENDEE:mailto:douglm@example.org
PERCENT-COMPLETE:39
END:VTODO
END: VCALENDAR
```

maps to

```
{
  "@type": "jstask",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
  ...
  "participants": {
    "be450b70-9bf7-4f6e-8f65-971ede566ce3": {
      "@type": "Participant",
      "sendTo": {
        "imip": "mailto:douglm@example.org"
      },
      "percentComplete": 39,
      "roles": {
        "attendee": true
      }
    }
  }
}
```

```
},  
...  
}
```

As ever recurrences complicate matters. For a non-recurring event or an override that contains the single participant, set the JSCalendar "percentComplete" property in the JSCalendar "participant" object representing the attendee.

```
BEGIN: VCALENDAR  
PRODID:-//ABC Corporation//NONSGML My Product//EN  
METHOD:REPLY  
BEGIN:VTODO  
...  
ATTENDEE:mailto:douglm@example.org  
END:VTODO  
BEGIN:VTODO  
...  
RECURRENCE-ID:20200523T120000  
...  
ATTENDEE:mailto:douglm@example.org  
PERCENT-COMPLETE:39  
END:VTODO  
END: VCALENDAR
```

maps to

```
{  
  "@type": "jstask",  
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",  
  ...  
  "participants": {  
    "be450b70-9bf7-4f6e-8f65-971ede566ce3": {  
      "@type": "Participant",  
      "sendTo": {  
        "imip": "mailto:douglm@example.org"  
      },  
      "roles": {  
        "attendee": true  
      }  
    },  
    "recurrenceOverrides": {  
      "2020-05-23T12:00:00": {  
        "participants/be4...6ce3/percentComplete": 39  
      },  
      ...  
    }  
  }  
}
```

In the case of an override with the participant appearing in the master then add a patch to the override.

5.14. PRIORITY

Simply copy value into the JSCalendar "priority" property.

5.15. PRODID

For a vcalendar JSGroup object with multiple JSEvent and/or JSTask object the [\[RFC5545\]](#) VCALENDAR PRODID is mapped to a JSCalendar "prodid" property in the group.

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT
...
END:VEVENT
END: VCALENDAR
```

maps to

```
{
  "@type": "jsevent",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
  ...
}
```

When mapping to a single JSEvent and/or JSTask object the [\[RFC5545\]](#) VCALENDAR PRODID is mapped to a JSCalendar "prodid" property in the group

5.16. RELATED-TO

This is mapped to the JSCalendar "relatedTo" property which is a map of relations with the target UID as the keys. The iCalendar relation is by default a PARENT relationship. There is no default for JSCalendar so the relationship must be explicitly specified.

```
...
RELATED-TO:jsmith.part7.19960817T083000.xyzMail@example.com
RELATED-TO;RELTYPE=SIBLING:
19960401-080045-4000F192713-0052@example.com
...
```

maps to

```
"relatedTo" : {
  "jsmith.part7.19960817T083000.xyzMail@example.com" : {
    "@type" : "Relation",
    "relation" : {
      "parent" : true
    }
  },
  "19960401-080045-4000F192713-0052@example.com" : {
    "@type" : "Relation",
    "relation" : {
      "next" : true
    }
  },
},
{
  "@type": "jsevent",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
```

```
...
}
```

Also note that the iCalendar relationship types are not identical. CHILD and PARENT map to JSCalendar "child" and "parent" but the best match for iCalendar SIBLING is "next"

5.17. RESOURCES

The RESOURCES property value is a comma-separated list of resources. First split this into the separate resource names and then each resource is mapped on a participant object with a "kind" property of "resource" and the "name" property set to the resource name.

Mapping parameters:

ALTREP

Use the same process as for the ATTENDEE DIR parameter: create a link property with the "rel" property set to "alternate" and the "href" property set to the value of the ALTREP parameter. The add the id of the link to the participants "linkIds" property.

LANG

Set the participants "language" property.

For an example see [Section 6.3](#)

5.18. RDATE

Create a patch object with the recurrence id set from the RDATE value. If the instance has overrides the differences will also be set in the object.

5.19. RRULE

```
...
RRULE:...
...

maps to

...
"recurrenceRules" : [{
  "@type" : "RecurrenceRule",
  ...
}],
...
```

Each RRULE is converted to an object in the JSCalendar "recurrenceRules" property. Each entry has the type "RecurrenceRule".

```
...
RRULE:...,BYDAY=-1MO
...

maps to

...
"recurrenceRules" : [{
  "@type" : "RecurrenceRule",
```

```

...
"byday": [{
  "day": "mo",
  "nthOfPeriod": -1
}]
...
}],
...

```

The recurrence rule object has one property for each element of the recurrence rule. The iCalendar rule has to be parsed out and the individual jscalendar property values set. Most take the same type but there are exceptions.

FREQ (mandatory)

Copy into the jscalendar "frequency" property converted to lowercase.

INTERVAL

If present and not 1 copy into the jscalendar "interval" property.

RSCALE

If present copy into the jscalendar "rscale" property converted to lowercase.

SKIP

If present copy into the jscalendar "skip" property converted to lowercase.

WKST

If present copy into the jscalendar "firstDayOfWeek" property converted to lowercase.

BYDAY

If present each element becomes an entry in the jsCalendar "byDay" property. This is an array of NDay objects which may have 2 properties:

day

The two character weekday abbreviation.

nthOfPeriod

If the weekday abbreviation is preceded by a signed integer value set the jscalendar "nthOfPeriod" property.

BYMONTHDAY

If present each element will be an element in the jscalendar "byMonthDay" property.

BYMONTH

If present each element will be an element in the jscalendar "byMonth" property.

Note that the iCalendar values are numeric but the JSCalendar values are strings. This is because of the possible "L" suffix for leap months.

BYYEARDAY

If present each element will be an element in the jscalendar "byYearDay" property.

BYWEEKNO

If present each element will be an element in the jscalendar "byWeekNo" property.

BYHOUR

If present each element will be an element in the jscalendar "byHour" property.

BYMINUTE

If present each element will be an element in the jscalendar "byMinute" property.

BYSECOND

If present each element will be an element in the jscalendar "bySecond" property.

BYSETPOS

If present each element will be an element in the jscalendar "bySetPosition" property.

COUNT

If present set in the jscalendar "count" property.

UNTIL

If present set the jscalendar "until" property with the appropriately reformatted value. If there is no time part append a 0 time and reformat as a jscalendar local date/time.

```
...  
RRULE:FREQ=DAILY;COUNT=10
```

```
...
```

maps to

```
...  
"recurrenceRules" : [{  
  "@type" : "RecurrenceRule",  
  "frequency": "daily",  
  "count": 10  
}],  
...
```

```
...  
RRULE:FREQ=YEARLY;UNTIL=20220512T140000Z;  
BYMONTH=1;BYDAY=SU,MO,TU,WE,TH,FR,SA
```

```
...
```

maps to

```
...  
"recurrenceRules" : [{  
  "@type" : "RecurrenceRule",  
  "frequency": "yearly",  
  "byMonth": ["1"],  
  "byDay": [{  
    "day": "su"  
  }],  
  "day": "mo"  
},  
  {  
    "day": "tu"  
  }],  
  {  
    "day": "we"  
  }],  
  {  
    "day": "th"  
  }],  
  {  
    "day": "fr"  
  }],  
  {  
    "day": "sa"
```

```
    }},  
    "until": "2022-05-12T10:00:00"  
  }},  
  ...
```

```
...  
RRULE:FREQ=MONTHLY;COUNT=6;BYDAY=-2MO  
...
```

maps to

```
...  
"recurrenceRules" : [{  
  "@type" : "RecurrenceRule",  
  "frequency": "monthly",  
  "byDay": [{  
    "day": "mo",  
    "nthOfPeriod": -2  
  }],  
  "count": 6  
}],  
...  
...
```

Some examples:

5.20. SEQUENCE

Copy the value into the JSCalendar "sequence" property.

5.21. STATUS

For a VEVENT copy the lower-cased value into the JSCalendar "status" property.

For a VTODO copy the lower-cased value into the JSCalendar "progress" property.

5.22. SUMMARY

Copy the value into the JSCalendar "title" property.

5.23. TRANSP

If the value of the TRANSP property (ignoring case) is "opaque" set the JSCalendar "freeBusyStatus" property to the value "busy".

Otherwise set the JSCalendar "freeBusyStatus" property to the value "free".

5.24. UID

Copy the value into the JSCalendar "uid" property.

5.25. URL

TBD.

6. Translating iCalendar: Further examples

This section provides more complete examples of translating from [\[RFC5545\]](#) to JSCalendar.

As usual note that json string values may be split because of line width limits. This is not legal json.

6.1. Recurring event with ATTACH

```
BEGIN:VCALENDAR
CALSCALE:GREGORIAN
PRODID: -//example.org//EN
VERSION:2.0
BEGIN:VEVENT
DTSTAMP:20200522T142047Z
DTSTART;TZID=America/New_York:20200522T120000
DURATION:PT1H
RRULE:FREQ=DAILY;COUNT=8
SUMMARY:recurring daily 8 times
UID:6252D6C40A8308BFE25BBDErecur-1
ATTACH;FMTTYPE=text/plain:http://example.org/doc1.txt
ATTACH;FMTTYPE=text/plain:http://example.org/doc2.txt
ATTACH;FMTTYPE=text/plain:http://example.org/doc3.txt
END:VEVENT
BEGIN:VEVENT
DTSTAMP:20200522T142047Z
DTSTART;TZID=America/New_York:20200523T120000
DURATION:PT1H
RECURRENCE-ID;TZID=America/New_York:20200523T120000
SUMMARY:recurring daily 8 times
UID:6252D6C40A8308BFE25BBDErecur-1
ATTACH;FMTTYPE=text/plain:http://example.org/doc2.txt
ATTACH;FMTTYPE=text/plain:http://example.org/doc3.txt
ATTACH;FMTTYPE=text/plain:http://example.org/doc4.txt
END:VEVENT
BEGIN:VEVENT
DTSTAMP:20200522T142047Z
DTSTART;TZID=America/New_York:20200524T120000
DURATION:PT1H
RECURRENCE-ID;TZID=America/New_York:20200524T120000
SUMMARY:recurring daily 8 times
UID:6252D6C40A8308BFE25BBDErecur-1
END:VEVENT
END:VCALENDAR
```

maps to

```
{
  "prodId": "-//example.org//EN",
  "entries": [
    {
      "links": {
        "d4a618d4-929c-4c81-ae5b-322afe407a00": {
          "@type": "Link",
          "rel": "enclosure",
          "contentType": "text/plain",
          "href": "http://example.org/doc1.txt"
```

```

},
"6c54e72e-3413-487c-ae14-fb318a90db43": {
  "@type": "Link",
  "rel": "enclosure",
  "contentType": "text/plain",
  "href": "http://example.org/doc2.txt"
},
"44087e9a-132c-4a5d-b25d-4ce580edb004": {
  "@type": "Link",
  "rel": "enclosure",
  "contentType": "text/plain",
  "href": "http://example.org/doc3.txt"
}
},
"created": "2020-05-23T17:04:50Z",
"start": "2020-05-22T12:00:00",
"timeZone": "America/New_York",
"duration": "PT1H",
"title": "recurring daily 8 times",
"uid": "6252D6C40A8308BFE25BBDErecur-1",
"recurrenceRules": [
  {
    "@type": "RecurrenceRule",
    "frequency": "daily",
    "count": 8
  }
],
"recurrenceOverrides": {
  "2020-05-23T12:00:00": {
    "recurrenceId": "2020-05-23T12:00:00",
    "links/d4a618d4-929c-4c81-ae5b-322afe407a00": null,
    "links/fb75b76a-a159-4a86-bd3d-7ace6b39c6c3": {
      "@type": "Link",
      "rel": "enclosure",
      "contentType": "text/plain",
      "href": "http://example.org/doc4.txt"
    }
  },
  "2020-05-24T12:00:00": {
    "recurrenceId": "2020-05-24T12:00:00",
    "links/d4a618d4-929c-4c81-ae5b-322afe407a00": null,
    "links/6c54e72e-3413-487c-ae14-fb318a90db43": null,
    "links/44087e9a-132c-4a5d-b25d-4ce580edb004": null
  }
}
}
]
}

```

This is an example of a recurring event with overrides. The first override removes an ATTACH property and adds an ATTACH property. The second override removes all ATTACH properties.

6.2. Simple event with CONTACT

```

BEGIN:VCALENDAR
CALSCALE:GREGORIAN
PROPID:../Example//EN
VERSION:2.0
BEGIN:VEVENT
DTSTAMP:20200522T142047Z
DTSTART;TZID=America/New_York:20200622T120000
DURATION:PT1H
SUMMARY:event with contact
UID:6252D6C40A8308BFE25BBEFcontact-1
CONTACT;ALTREP="ldap://example.com:6666/o=ABC%20Industries\,
c=US???(cn=Jim%20Dolittle)":Jim Dolittle\, ABC Industries\,
+1-919-555-1234
END:VEVENT
END:VCALENDAR

```

translates to

```

{
  "@type": "jsgroup",
  "proId": "../Example.org//Example V3.13.2//EN",
  "entries": [
    {
      "@type": "jsevent",
      "participants": {
        "40288108-733187c1-0173-3188007b-00000001": {
          "@type": "Participant",
          "roles": {
            "contact": true
          },
          "description": "Jim Dolittle, ABC Industries,\
            +1-919-555-1234",
          "linkIds": {
            "2e1025a2-1f1a-44f3-a867-75e19ce38b46": true
          }
        }
      },
      "links": {
        "2e1025a2-1f1a-44f3-a867-75e19ce38b46": {
          "@type": "Link",
          "href": "ldap://example.com:6666/o=ABC%20Industries\,
            c=US???(cn=Jim%20Dolittle)",
          "rel": "alternate"
        }
      },
      "created": "2020-07-09T03:04:23Z",
      "start": "2020-06-22T12:00:00",
      "timeZone": "America/New_York",
      "duration": "PT1H",
      "title": "event with contact",
      "uid": "6252D6C40A8308BFE25BBEFcontact-1"
    }
  ]
}

```

```
}
```

This example shows the conversion of a simple event with a single CONTACT property in JSCalendar.

6.3. Simple event with RESOURCES

TBD

6.4. Recurring event. Attendees only in overrides

In this more complex example there is no ORGANIZER or ATTENDEEs in the master event. There are overrides which invite one or more attendees.

For one override the ORGANIZER is also an ATTENDEE. In the other that is not the case. This is reflected in the "roles" property for the organizer.

```
BEGIN:VCALENDAR
PRODID://Example.org//Example V3.13.2//EN
VERSION:2.0
BEGIN:VEVENT
CREATED:20200704T035515Z
DURATION:PT1H
DTSTAMP:20200704T035706Z
DTSTART;TZID=America/New_York:20200522T120000
LAST-MODIFIED:20200704T035706Z
SUMMARY:recurring daily 8 times
UID:6252D6C40A8308BFE25BBEFrecur1-1
RRULE:FREQ=DAILY;COUNT=8
END:VEVENT
BEGIN:VEVENT
RECURRENCE-ID;TZID=America/New_York:20200523T120000
ATTENDEE:mailto:douglm@example.org
ATTENDEE;RSVP=TRUE;SCHEDULE-STATUS=1.2;DIR="http://example.org/
vcards/vbede.vcf":mailto:vbede@example.org
CREATED:20200704T035515Z
DURATION:PT1H
DTSTAMP:20200704T035706Z
DTSTART;TZID=America/New_York:20200523T120000
LAST-MODIFIED:20200704T035706Z
ORGANIZER:mailto:douglm@example.org
SUMMARY:recurring daily 8 times
UID:6252D6C40A8308BFE25BBEFrecur1-1
END:VEVENT
BEGIN:VEVENT
RECURRENCE-ID;TZID=America/New_York:20200524T120000
ATTENDEE;RSVP=TRUE;SCHEDULE-STATUS=1.2:mailto:user01@example.org
ATTENDEE;RSVP=TRUE;SCHEDULE-STATUS=1.2:mailto:vbede@example.org
CREATED:20200704T035515Z
DURATION:PT1H
DTSTAMP:20200704T035706Z
DTSTART;TZID=America/New_York:20200524T120000
LAST-MODIFIED:20200704T035706Z
ORGANIZER:mailto:douglm@example.org
SUMMARY:recurring daily 8 times
```

UID:6252D6C40A8308BFE25BBEFrecur1-1
END:VEVENT
END:VCALENDAR

translates to

```
{
  "@type": "jsgroup",
  "prodId": "-//Example.org//Example V3.13.2//EN",
  "entries": [
    {
      "@type": "jsevent",
      "created": "2020-07-04T03:57:06Z",
      "start": "2020-05-22T12:00:00",
      "timeZone": "America/New_York",
      "duration": "PT1H",
      "title": "recurring daily 8 times",
      "uid": "6252D6C40A8308BFE25BBEFrecur1-1",
      "recurrenceRules": [
        {
          "@type": "RecurrenceRule",
          "frequency": "daily",
          "count": 8
        }
      ],
      "recurrenceOverrides": {
        "2020-05-23T12:00:00": {
          "participants": {
            "be450b70-9bf7-4f6e-8f65-971ede566ce3": {
              "@type": "Participant",
              "sendTo": {
                "imip": "mailto:douglm@example.org"
              },
              "roles": {
                "attendee": true,
                "owner": true
              }
            },
            "a539dfe3-4463-4f28-b9de-17d3a0e99faf": {
              "@type": "Participant",
              "sendTo": {
                "imip": "mailto:vbede@example.org"
              },
              "expectReply": true,
              "linkIds": {
                "0234793d-201c-4e64-8081-209878c63c41": true
              },
              "roles": {
                "attendee": true
              }
            }
          },
          "scheduleStatus": "1.2"
        }
      ]
    }
  ],
}
```

```
"links": {
  "0234793d-201c-4e64-8081-209878c63c41": {
    "@type": "Link",
    "href": "http://example.org/vcards/vbede.vcf",
    "rel": "alternate"
  }
},
"replyTo": {
  "imip": "mailto:douglm@example.org"
}
},
"2020-05-24T12:00:00": {
  "participants": {
    "daeae4cf-6f6a-4ce3-9f4d-6bd884650d3d": {
      "@type": "Participant",
      "sendTo": {
        "imip": "mailto:user01@example.org"
      },
      "expectReply": true,
      "roles": {
        "attendee": true
      },
      "scheduleStatus": "1.2"
    },
    "a6de6de3-271f-4679-9241-1b3bca6b602d": {
      "@type": "Participant",
      "sendTo": {
        "imip": "mailto:vbede@example.org"
      },
      "expectReply": true,
      "roles": {
        "attendee": true
      },
      "scheduleStatus": "1.2"
    },
    "aaa8483b-b18b-4dbd-b218-77d8db027d35": {
      "@type": "Participant",
      "sendTo": {
        "imip": "mailto:douglm@example.org"
      },
      "roles": {
        "owner": true
      }
    }
  }
},
"replyTo": {
  "imip": "mailto:douglm@example.org"
}
}
}
]
}
```

Note that each override has its own "participants" property and the first has a links property to handle the DIR parameter on one attendee.

7. Translating JSCalendar properties to iCalendar components

This section is an alphabetic list of all JSCalendar property types that map on to [\[RFC5545\]](#) components.

7.1. jsevent

A JSCalendar object with a type of "jsevent" is mapped on to a [\[RFC5545\]](#) VEVENT component.

```
{
  "@type": "jsevent",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
  ...
}
```

maps to

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT
...
END:VEVENT
END: VCALENDAR
```

If it is a single VEVENT then a [\[RFC5545\]](#) VCALENDAR component must surround it and the JSCalendar "prodid" property will be converted to a [\[RFC5545\]](#) PRODID.

When converting multiple JSEvent or JSTask objects the surrounding [\[RFC5545\]](#) VCALENDAR object must have a [\[RFC5545\]](#) PRODID set from either the JSGroup "prodid" or generated.

7.2. jsgroup

```
{
  "@type": "jsgroup",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
  ...
  {
    "@type": "jsevent",
    ...
  }
  {
    "@type": "jsevent",
    ...
  }
}
```

maps to

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VEVENT
...
```

```
END:VEVENT
BEGIN:VEVENT
...
END:VEVENT
END: VCALENDAR
```

A JSCalendar object with a type of "jsgroup" is mapped on to a [\[RFC5545\]](#) VCALENDAR component.

7.3. jstask

A JSCalendar object with a type of "jstask" is mapped on to a [\[RFC5545\]](#) VTOD component.

```
{
  "@type": "jstask",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
  ...
}
```

maps to

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
BEGIN:VTOD
...
END:VTOD
END: VCALENDAR
```

If it is a single VTOD then a [\[RFC5545\]](#) VCALENDAR component must surround it and the JSCalendar "prodid" property will be converted to a [\[RFC5545\]](#) PRODID.

When converting multiple JSEvent or JSTask objects the surrounding [\[RFC5545\]](#) VCALENDAR object must have a [\[RFC5545\]](#) PRODID set from either the JSGroup "prodid" or generated.

7.4. timezones

```
{
  "@type": "jsevent",
  ...
  "timezones": {
    "/Example/Somewhere: {
      "@type": "TimeZone",
      ...
    },
    "/Example/Somewhere-else": {
      "@type": "TimeZone",
      ...
    }
  }
}
```

maps to

```
BEGIN: VTIMEZONE
TZID: /Example/Somewhere
```

```
...
END: VTIMEZONE
BEGIN: VTIMEZONE
TZID: /Example/Somewhere-else
...
END: VTIMEZONE
BEGIN: VEVENT
...
END: VEVENT
```

The JSCalendar TimeZone objects within a "timezones" property are mapped on to [\[RFC5545\]](#) VTIMEZONE components within the surrounding VCALENDAR component. Each mapped TimeZone MUST only appear once.

When converting multiple JSEvent or JSTask objects the surrounding [\[RFC5545\]](#) VCALENDAR object must have a [\[RFC5545\]](#) PRODID set from either the JSGroup "prodid" or generated.

8. Some notes

The sections following this one are all the original ones from draft 1 written by Robert/Neil - there for reference.

8.1. Alerts

I've got text which explains some of the justification for the changes. Because it's not trivial to map I think people need to understand the reasoning.

8.2. Preserving icalendar

I don't think we need to preserve the entire iCalendar object to make this round trippable. That also means somebody needs to parse it and extract what they need.

Rather - where it matters we could use the JCal approach. For example - preserve VALARM

8.3. JSCalendar changes

Here I'll note what I think needs changing in the jscalendar spec.

Links with a rel of "enclosure" SHOULD be considered by the client as attachments for download.

```
"participantTypes": {"attendee": true}
"roles": {"optional": true}
```

is easier to understand than

```
"participantTypes": {"optional": true}
```

implying this is an optional attendee.

Changing role "attendee" to "required" or "expected" might make sense

alert - absolute time

I don't think we should promote UTC. A change in rules could result in alarms all going off at the

wrong time. I think we should allow for local time and a timezone.

attach - ATTACH

The text for links says

comments - COMMENT

comments is only specified for timezone rules. It needs to be valid for all or some indication of how to handle them needs to be provided.

There is an issue in 5545 with COMMENT in scheduling e.g. is a comment meant only for the organizer? That could be tightened up.

CONTACT

There's no mapping suggested. CONTACT is vital for event publication. I'd suggest following the approach in eventpub - add a participantType property to participant

I thought this could probably be handled just by changing the wording for participant as I suggested and extending the roles values. However I think that's conceptually more complicated,

GEO

Probably need a well defined location type to bury this in.

participant

Bring into line with eventpub extensions so we can do a consistent mapping between them. Add the following:

order

Positive number.

participantTypes

As defined.

structuredData

with "encoding" and "value" properties. We need to have a place to put the value which may be a phone number. As a fallback converting to JSCalendar may involve encoding the value as a data uri.

uri

To match ALTREP.

Note that eventpub extensions needs an update: PARTICIPANT-TYPE needs to be allowed more than once and ATTENDEE should be added as a value.

PERCENT-COMPLETE

Add this?

PRORITY

Not scheduling or sharing specific. Text for UIs?

privacy

Still think we should change "secret" to "confidential".

relation

No mapping for "first". Define this in relations draft? Is "next" == SIBLING?

RESOURCES

No obvious mapping?

section 3.1 - etags text

Needs correcting.

timezones

I'm completely fine with the default being by reference. It may be good to strengthen the language in section 1.4.9. A concern is we're not allowing standalone timezone defs. I see 2 uses for that:

1. A group containing a lot of events that all use the same custom timezone -

- having to repeat that per event is bulky. (probably not a major case but...). I guess the first event could be considered the definition but that's messy.
2. More important - how do we deliver jscalender format timezones from tzdist?

Could we simply allow a timezones property as another alternative in JSGroup?

relation type

Include the values from the relations draft spec.

URL

presumably a link but need a way to identify it?

9. JSEvent

A JSEvent maps to the the iCalendar VEVENT component type [\[RFC5545\]](#). The following tables maps the JSEvent-specific properties to iCalendar:

Property	iCalendar counterpart
duration	DURATION property. If the VEVENT contains a DTEND property, the this maps to the duration property as the time span between DTSTART and DTEND when converting the respective time points to the UTC time zone. Fractional seconds SHOULD be preserved with the SUBSECOND parameter.

Mapping JSEvent properties

10. JSTask

A JSTask object maps to the iCalendar VTODO component type [\[RFC5545\]](#). The following tables maps the JSTask-specific properties to iCalendar:

Property	iCalendar counterpart
due	Maps to the DUE property. See Section 12.1 .
estimatedDuration	ESTIMATED-DURATION property in the RFC draft [draft-apthorp-ical-tasks] , or the DURATION property otherwise. Fractional seconds SHOULD be preserved with the SUBSECOND parameter.
statusUpdatedAt	COMPLETED property. The JSTask status property MUST have value completed. Fractional seconds SHOULD be preserved with the SUBSECOND parameter.
progress	PARTSTAT and COMPLETED properties, including the definitions in the RFC draft [draft-apthorp-ical-tasks] .
status	STATUS property, including the definitions in the RFC draft [draft-apthorp-ical-tasks] .

Mapping JSTask properties

11. JSGroup

A JSGroup maps to a iCalendar VCALENDAR containing VEVENT or VTODO components.

Property	iCalendar counterpart
entries	VEVENT and VTODO components embedded in a VCALENDAR component.
source	SOURCE property.

Mapping JSGroup properties

12. Common properties

This section contains recommendations how to map JSCalendar from and to iCalendar. It lists all common

JSCalendar object properties in alphabetical order.

Property	iCalendar counterpart
@type	Determined by the iCalendar component type: jsevent for VEVENT, jstask for VTODO, jsgroup for VCALENDAR.
alerts	Each entry maps to a VALARM component. The action property maps to iCalendar ACTION, where both iCalendar DISPLAY and AUDIO values map to the display action. An EMAIL value maps to a JSCalendar email action. <i>relativeTo</i> and <i>offset</i> map to the TRIGGER property.
categories	CONCEPT property, defined in [draft-ietf-calext-ical-relations] .
color	COLOR property, as specified in [RFC7986] .
created	CREATED property. Fractional seconds SHOULD be preserved with the SUBSECOND parameter.
description	DESCRIPTION property.
descriptionContentType	Implementation-specific.
excluded	EXDATE property. Fractional seconds SHOULD be preserved with the SUBSECOND parameter.
freeBusyStatus	TRANSP property.
invitedBy	Implementation-specific.
keywords	CATEGORIES property, as specified in [RFC7986] .
links	ATTACH ([RFC5545]), URL or IMAGE ([RFC7986]) properties with URI value types map to the the Link <i>href</i> . The FMTTYPE parameter maps to <i>type</i> , the SIZE parameter to <i>size</i> . Mapping other properties is implementation-specific.
locale	LANGUAGE parameter of the SUMMARY or DESCRIPTION property.
localizations	Implementation-specific.
locations	See Section 12.2 .
method	METHOD property of the embedding VCALENDAR.
participants	See Section 12.3 .
priority	PRIORITY property.
privacy	CLASS property.
prodId	PRODID property.
recurrenceOverrides	RDATE and EXDATE properties, and any VEVENT or VTODO instances with a recurrence-id and same UID as the mapped main object. If the DTSTART property defines a SUBSECOND parameter, but the RECURRENCE-ID of a recurrence instance does not, then use the SUBSECOND parameter value of DTSTART to determine the recurrence override time stamp.
recurrenceRule	RRULE property. For all-day calendar objects, map the <i>until</i> property value to an iCalendar DATE (effectively removing the time component). To convert a DATE-typed UNTIL from iCalendar, set the time components of the LocalDateTime value to 23:59:59. If the iCalendar UNTIL value is a UTC date time, convert it to the local time in the JSCalendar calendar object time zone. To convert to iCalendar where the DTSTART or DUE property is of type DATE, omit the time component of the LocalDateTime value.
relatedTo	RELATED-TO property.

Property	iCalendar counterpart
replyTo	An iCalendar ORGANIZER with a mailto: URI mapped to the imip method, or any other URI mapped to the other method. Mapping multiple methods is implementation-specific.
sequence	SEQUENCE property.
showWithoutTime	Implementation-specific.
start	Maps to the DTSTART property. See Section 12.1 .
status	STATUS property.
timeZone	Maps to the TZID parameter. See Section 12.1 .
timeZones	Each entry in the property maps to a VTIMEZONE in the embedding VCALENDAR component.
title	SUMMARY property.
uid	UID property.
updated	DTSTAMP and LAST-MODIFIED properties. Fractional seconds SHOULD be preserved with the SUBSECOND parameter.
useDefaultAlerts	Implementation-specific.
virtualLocations	See Section 12.2 .

Translation between JSCalendar and iCalendar

12.1. Time properties and types

iCalendar defines two different time types, DATE and DATE-TIME, where the latter may occur in three forms (with local time, with UTC time, with local time and time zone reference). In contrast, JSCalendar does not define a distinct type for dates, and date times are defined with the LocalDateTime type only.

A JSCalendar time maps to the iCalendar DATE type if all of the following criteria apply:

For all other cases, the time maps to an iCalendar DATE-TIME:

- The start (due) property value has zero time, or is not set.
- The duration (estimatedDuration) property value has zero time, or is a multiple of days or weeks, or is not set.
- The timeZone property value is null, or is not set.
- With local time and time zone reference, if the timeZone property value is set and does not equal Etc/UTC.
- With UTC time, if the timeZone property value equals Etc/UTC.
- With local time, if the timeZone property value is null or not set.

Fractional seconds SHOULD be preserved with the SUBSECOND parameter.

12.2. Locations

The iCalendar counterpart for JSCalendar Location objects is the iCalendar [\[RFC5545\]](#) LOCATION property, or implementation-specific.

Property	iCalendar counterpart
coordinates	GEO property.
description	Implementation-specific.

Property	iCalendar counterpart
linkIds	Implementation-specific.
name	LOCATION property value.
rel	Implementation-specific.
timeZone	Implementation-specific.
uri	The LOCATION ALTREP parameter.

Mapping Location properties

The iCalendar counterpart for JSCalendar VirtualLocation objects is the iCalendar [\[RFC7986\]](#) CONFERENCE property.

Property	iCalendar counterpart
description	Implementation-specific.
name	LABEL parameter.
uri	CONFERENCE property value.

Mapping virtualLocation properties

12.3. Participants

The following table outlines translation of JSCalendar participants. An iCalendar ORGANIZER maps to both the replyTo property and a participant with role owner. If an ATTENDEE with the same CAL-ADDRESS value exists, then it maps to the same participant as the ORGANIZER participant. Other participants map to ATTENDEEs.

Property	iCalendar counterpart
attendance	ROLE parameter values REQ-PARTICIPANT, OPT-PARTICIPANT and NON-PARTICIPANT.
delegatedFrom	DELEGATED-FROM parameter
delegatedTo	DELEGATED-TO parameter
email	EMAIL parameter, if defined. Otherwise the CAL-ADDRESS property value, if it is a mailto: URI.
expectReply	RSVP parameter
kind	CUTYPE parameter
linkIds	Implementation-specific.
locationId	Implementation-specific.
memberOf	MEMBER parameter
name	CN parameter
participationStatus	PARTSTAT parameter
roles	ROLE parameter.
scheduleSequence	SEQUENCE property of the participant's latest iMIP message
scheduleUpdated	DTSTAMP property of the participant's latest iMIP message
sendTo	A CAL-ADDRESS with a mailto: URI maps to the JSCalendar imip method, any other URI to the other method. Mapping multiple methods is implementation-specific.

Mapping Participant properties

Property	iCalendar counterpart
13. Custom properties	

Mapping custom or unknown properties between JSCalendar and iCalendar is implementation-specific. Implementations might use vendor-extension properties, which could also serve as basis for discussion for a JSCalendar standard extension. Alternatively, an implementation could preserve iCalendar properties and components in JSCalendar by use of a vendor-extension property formatted as jCal [\[RFC7265\]](#) data.

14. Security Considerations

The same security considerations as for [\[draft-ietf-calext-jscalendar\]](#) apply.

15. IANA Considerations

None.

16. Acknowledgments

The authors would like to thank the members of CalConnect for their valuable contributions. This specification originated from the work of the API technical committee of CalConnect, the Calendaring and Scheduling Consortium.

17. References

17.1. Normative References

- [RFC2119]** Bradner, S., "[Key words for use in RFCs to Indicate Requirement Levels](#)", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997.
- [RFC5545]** Desruisseaux, B., "[Internet Calendaring and Scheduling Core Object Specification \(iCalendar\)](#)", RFC 5545, DOI 10.17487/RFC5545, September 2009.
- [RFC6638]** Daboo, C. and B. Desruisseaux, "[Scheduling Extensions to CalDAV](#)", RFC 6638, DOI 10.17487/RFC6638, June 2012.
- [RFC7159]** Bray, T., "[The JavaScript Object Notation \(JSON\) Data Interchange Format](#)", RFC 7159, DOI 10.17487/RFC7159, March 2014.
- [RFC7265]** Kewisch, P., Daboo, C. and M. Douglass, "[jCal: The JSON Format for iCalendar](#)", RFC 7265, DOI 10.17487/RFC7265, May 2014.
- [RFC7986]** Daboo, C., "[New Properties for iCalendar](#)", RFC 7986, DOI 10.17487/RFC7986, October 2016.

17.2. Informative References

- [draft-apthorp-ical-tasks]** "[Task Extensions to iCalendar](#)"
- [draft-ietf-calext-ical-relations]** "[Support for iCalendar Relationships](#)"
- [draft-ietf-calext-jscalendar]** "[Task Extensions to iCalendar](#)"

Authors' Addresses

Neil Jenkins
FastMail
PO Box 234
Collins St West
Melbourne, VIC 8007
Australia

E-Mail: neilj@fastmailteam.com

URI: <https://www.fastmail.com>

Robert Stepanek

FastMail

PO Box 234

Collins St West

Melbourne, VIC 8007

Australia

E-Mail: rsto@fastmailteam.com

URI: <https://www.fastmail.com>

Michael Douglass

Bedework

226 3rd Street

Troy, NY 12180

United States of America

E-Mail: mdouglass@bedework.com

URI: <http://bedework.com>