

Calendaring extensions
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2022

N. Jenkins
R. Stepanek
FastMail
M. Douglass
BCS
July 9, 2021

**JSCalendar: Converting from and to iCalendar
draft-ietf-calext-jscalendar-icalendar-05**

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 10, 2022.

Copyright Notice

Copyright (c) 2021 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	4
1.1.	Note (to be deleted later)	4
1.2.	Motivation	4
1.3.	Scope and caveats	4
1.4.	Notational Conventions	5
2.	iCalendar pre-processing	5
3.	Translating iCalendar components to JSCalendar	6
3.1.	VALARM	7
3.2.	VCALENDAR	9
3.3.	VEVENT	10
3.4.	VLOCATION	10
3.5.	VTIMEZONE, STANDARD, DAYLIGHT	11
3.6.	VTODO	12
4.	Translating iCalendar properties to JSCalendar	13
4.1.	ATTACH	13
4.2.	ATTENDEE	13
4.3.	CALSCALE	16
4.4.	CATEGORIES	16
4.5.	CLASS	16
4.6.	COLOR	17
4.7.	COMMENT	17
4.8.	COMPLETED	17
4.9.	CONCEPT	17
4.10.	CONFERENCE	18
4.11.	CONTACT	18
4.12.	CREATED	19
4.13.	DESCRIPTION	19
4.14.	DTEND, DTSTART, DUE, DURATION	20
4.15.	ESTIMATED-DURATION	23
4.16.	EXDATE	23
4.17.	EXRULE	23
4.18.	DTSTAMP and LAST-MODIFIED	23
4.19.	GEO	24
4.20.	IMAGE	24
4.21.	LOCATION	24
4.22.	METHOD	24
4.23.	ORGANIZER	25
4.24.	PERCENT-COMPLETE	25
4.25.	PRIORITY	27
4.26.	PRODID	28
4.27.	RECURRENCE-ID	28
4.28.	RELATED-TO	28
4.29.	REQUEST-STATUS	29
4.30.	RESOURCES	29
4.31.	RDATE	30
4.32.	RRULE	30

4.33.	SEQUENCE	34
4.34.	STATUS	34
4.35.	STRUCTURED-DATA	34
4.36.	SUMMARY	35
4.37.	TRANSP	35
4.38.	UID	36
4.39.	URL	36
5.	Translating iCalendar Recurrences	36
5.1.	Translating iCalendar Recurrences: Simple objects with overrides	36
5.2.	Translating iCalendar Recurrences: Overrides with no master	36
6.	Translating iCalendar: Further examples	36
6.1.	Recurring event with ATTACH	37
6.2.	Simple event with CONTACT	39
6.3.	Simple event with RESOURCES	40
6.4.	Recurring event. Attendees only in overrides	40
7.	Translating JSCalendar properties to iCalendar components	43
7.1.	jsevent	43
7.2.	jsgroup	44
7.3.	jstask	45
7.4.	timezones	46
7.5.	alerts	47
7.5.1.	action	47
7.5.2.	trigger	48
7.5.3.	todo	49
8.	Translating JSCalendar properties to iCalendar properties	49
8.1.	categories	49
8.2.	created	50
8.3.	duration	50
8.4.	estimatedDuration	51
8.5.	keywords	51
8.6.	locations	52
8.7.	participants	52
9.	Security Considerations	53
10.	IANA Considerations	53
11.	Acknowledgments	53
12.	References	53
12.1.	Normative References	53
12.2.	Informative References	54
Appendix A.	Outdated document sections	55
A.1.	Note	55
A.2.	JSTask	55
A.3.	JSGroup	55
A.4.	Common properties	56
A.4.1.	Time properties and types	58
A.4.2.	Locations	59
A.4.3.	Participants	60

A.5. Custom properties	61
Authors' Addresses	62

[1.](#) Introduction

[1.1.](#) Note (to be deleted later)

This is still very much a work in progress. There are implementations of the mapping but there may be changes over the coming weeks.

[1.2.](#) 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.

[1.3.](#) Scope and caveats

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

- o 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.
- o iCalendar may contain arbitrary, non-standardised data with custom properties/attributes. These will be translated using the same approach as jCal.

- o 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.
- o 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.
- o JSCalendar supports fractional seconds in time values whereas iCalendar does not. A subsequent specification will define how fractional seconds can be represented in iCalendar.

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.

This document defines mappings for the following specifications.

- o Internet Calendaring and Scheduling Core Object Specification (iCalendar)
- o iCalendar Transport-Independent Interoperability Protocol (iTIP)
- o New Properties for iCalendar
- o Event Publishing Extensions to iCalendar
- o Support for iCalendar Relationships
- o VALARM Extensions for iCalendar

Therefore all of these specifications **MUST** be implemented to follow this specification.

1.4. 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. 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":

- o If the property value type is "BINARY", the base64 encoding MUST be preserved.
- o 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.

3. Translating iCalendar components to JSCalendar

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

At present VFREEBUSY and VJOURNAL are not mapped in jscalendar.

3.1. VALARM

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

```
BEGIN: VEVENT
...
BEGIN: VALARM
...
END: VALARM
BEGIN: VALARM
...
END: VALARM
END: VEVENT
```

maps to

```
{
  "@type": "Event",
  ...
  "alerts": {
    "1": {
      "@type": "Alert",
      ...
    },
    "2": {
      "@type": "Alert",
      ...
    }
  }
}
```

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.

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

```
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\n
  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": "Event",
  ...
  "alerts": {
    "1": {
      "@type": "Alert",
      "action": "display",
      "trigger": {
        "@type": "AbsoluteTrigger",
        "when": "19970317T133000Z"
      }
    },
    "2": {
      "@type": "Alert",
      "action": "display",
      "trigger": {
        "@type": "OffsetTrigger",
        "offset": "-PT30M"
      }
    }
  },
  "3": {
    "@type": "Alert",
    "action": "email",
    "trigger": {
      "@type": "OffsetTrigger",
      "offset": "-P2D",
      "relativeTo": "end"
    }
  }
}
```

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

3.2. VCALENDAR

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


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

maps to

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

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

[3.3.](#) VEVENT

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

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

maps to

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

[3.4.](#) VLOCATION

A [\[draft-ietf-calext-eventpub-extensions\]](#) VLOCATION component is mapped to a JSCalendar object with a type of "Location". Any properties within the VLOCATION must be mapped as described below.


```
BEGIN: VEVENT
...
BEGIN: VLOCATION
...
END: VLOCATION
END: VEVENT
```

maps to

```
{
  "@type": "jsevent",
  ...
  "locations": {
    "1": {
      "@type": "Location",
      ...
    }
  }
}
```

3.5. 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

Note that

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


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

3.6. VTOD0

A [\[RFC5545\]](#) VTOD0 component is mapped to a JSCalendar object with a type of "jstask".

```
BEGIN: VTOD0
...
END: VTOD0
```

maps to

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


4. Translating iCalendar properties to JSCalendar

This section is an alphabetic list of [[RFC5545](#)] and [[RFC7986](#)] properties and how they are mapped to JSCalendar.

4.1. ATTACH

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

- o A uri value
- o 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](#)

4.2. ATTENDEE

An [[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.

For example:


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

maps to

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

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 participant "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"

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.

[4.3.](#) CALSCALE

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

[4.4.](#) CATEGORIES

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

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

maps to

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

[4.5.](#) CLASS

Maps to the "privacy" property. The iCalendar property value maps to the JSCalendar value as follows:

CONFIDENTIAL "secret"

PRIVATE "private"

PUBLIC "public"

iana-token and x-name verbatim copy

[4.6.](#) COLOR

Maps to the "color" property. Copy the verbatim value.

[4.7.](#) 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.

[4.8.](#) COMPLETED

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

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

maps to

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

[4.9.](#) CONCEPT

This [\[draft-ietf-calext-ical-relations\]](#) property may appear multiple times in components.

Each instance of the property is mapped on to a member of the JSCalendar "categories" property.


```
...  
CONCEPT:http://example.com/event-types/arts/music  
CONCEPT:http://example.com/performance-types/arts/live  
...
```

maps to

```
...  
"categories": {  
  "http://example.com/event-types/arts/music": true,  
  "http://example.com/performance-types/arts/live": true  
}  
...
```

4.10. CONFERENCE

Maps to a "VirtualLocation" object. The property value maps to the "uri" property of the virtual location.

Mapping parameters:

FEATURE: Maps to the "features" property of the virtual location.

LABEL: Maps to the "name" property of the virtual location.

LANGUAGE: No mapping.

4.11. 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 link to the participants "links" property.

LANG Set the participants "language" property.

For an example see [Section 6.2](#)

[4.12.](#) **CREATED**

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

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

maps to

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

[4.13.](#) **DESCRIPTION**

Copy the value, preprocessed according to [Section 2](#) into the "description" property.

Mapping parameters:

ALTREP No mapping.

LANG Use the "locale" property.

Example:


```
BEGIN:VEVENT
```

```
...
```

```
DESCRIPTION: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.
```

```
...
```

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

4.14. 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.

Example - DTSTART and DTEND in same timezone:

```
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 different 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": {
    "1": {
      "@type": "location",
      "relatedTo": "end",
      "timeZone": "America/Los_Angeles"
    }
  }
}
```

Example - 3 day event:

```
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,
  ...
}
```


[4.15.](#) ESTIMATED-DURATION

Copy the ESTIMATED-DURATION value into the JSCalendar "estimatedDuration" property.

For example:

```
...  
ESTIMATED-DURATION:PT18H  
...
```

maps to

```
...  
"estimatedDuration": "PT18H"  
...
```

[4.16.](#) 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".

[4.17.](#) EXRULE

Maps to the "excludedRecurrenceRules" property. Also see [Section 4.32](#).

[4.18.](#) 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.

[4.19.](#) GEO

Maps to a Location object, with only the "coordinates" property set. Note that the JSCalendar coordinates property value MUST be a valid "geo" URI, so replace the ";" character in the iCalendar value with "," and prepend the resulting string with "geo:".

[4.20.](#) IMAGE

Maps to a Link object with the iCalendar property value mapped to the location "href" property, and the "rel" property set to "icon".

For a binary value use a base64 data uri in the "href" property.

Mapping parameters:

ALTREP No mapping.

FMTTYPE Maps to the "contentType" property of the Link object.

DISPLAY Maps to the "display" property of the Link object. The property values "BADGE", "GRAPHIC", "FULLSIZE" and "THUMBNAIL" map to their lower-case equivalent in JSCalendar.

[4.21.](#) LOCATION

If any [[draft-ietf-calext-eventpub-extensions](#)] "VLOCATION" components are present, then the [[RFC5545](#)] "LOCATION" property should be ignored.

To map the property create a "locations" property with a single "location" and set the "description" property to the value of the [[RFC5545](#)] "LOCATION" property.

Mapping parameters:

ALTREP Maps to a Link object in the Location "links" property, with the "href" property set to the parameter value.

[4.22.](#) METHOD

Maps to the "method" property of the JSCalendar object. The JSCalendar property value is the lowercase equivalent of the iCalendar property value.

[4.23.](#) ORGANIZER

Maps to the "replyTo" property of the JSCalendar object. An iCalendar property value in the "mailto:" URI scheme, maps to the "imip" method, any other value maps to the "other" method.

If the iCalendar component also contains an ATTENDEE with the same calendar user address then map that ATTENDEE as defined in [Section 4.2](#) and add the "owner" role to the Participant "roles" property. Otherwise, use the ORGANIZER property to map to a Participant object. The "roles" property of the Participant MUST only contain the "owner" role and the "expectReply" property value MUST be "false". Any iCalendar parameters map as defined for ATTENDEE.

TBD: SENT-BY parameter. Example.

[4.24.](#) PERCENT-COMPLETE

For all methods other than REPLY (or no method), the PERCENT-COMPLETE applies to the VTOD0 as a whole. In this case it the value is used to set the JSCalendar "percentComplete" property in the task object.

```
BEGIN: VCALENDAR
PRODID:-//ABC Corporation//NONSGML My Product//EN
METHOD:PUBLISH
BEGIN:VTOD0
...
PERCENT-COMPLETE:39
END:VTOD0
END: VCALENDAR
```

maps to

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

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 VTOD0 object.

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
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
      }
    },
    ...
  }
}
```

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


```

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
      },
      ...
    }
  }
}

```

4.25. PRIORITY

Simply copy value into the JSCalendar "priority" property.

[4.26.](#) **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.

When mapping to a single 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",
  ...
}
```

[4.27.](#) **RECURRENCE-ID**

Refer to [Section 5](#) for information on mapping recurrence ids.

[4.28.](#) **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.

The RELTYPE parameter values map to their lowercase equivalents in the "relation" property.

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"


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

[4.29.](#) REQUEST-STATUS

Copy the value into the JSCalendar "requestStatus" property.

[4.30.](#) 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. Then add the link to the participants "links" property.

LANG Set the participants "language" property.

For an example see [Section 6.3](#)

4.31. RDATE

If the RDATE has a RANGE=THISANDFUTURE parameter then the recurrence MUST be split at this RDATE.

Truncate the original object before this RDATE, create a new master representing the object and link them by setting the jscalendar "relatedTo" property in both.

Otherwise 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.

4.32. RRULE

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

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

maps to

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

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.

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

maps to

```
...
"recurrenceRules" : [{
  "@type" : "RecurrenceRule",
  ...
  "byday": [{
    "day": "mo",
    "nthOfPeriod": -1
  }]
  ...
}],
...
```

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.

Some examples:

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

[4.33.](#) SEQUENCE

Copy the value into the JSCalendar "sequence" property.

[4.34.](#) 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.

[4.35.](#) STRUCTURED-DATA

This property is mapped on to a JSCalendar "link" object with the value mapped on to the JSCalendar "href" property in a manner depending on the "STRUCTURED-DATA" "VALUE" parameter:

VALUE=TEXT Copy the value as a [[RFC2397](#)] data uri either as plain text or by encoding as a base64 value. If plain text the value may need escaping as per [[RFC2397](#)].

VALUE=BINARY Copy the value as a [[RFC2397](#)] data uri specifying base64 encoding.

VALUE=URI Copy the value as-is into the href.

The "STRUCTURED-DATA" "SCHEMA" parameter is mapped on to a JSCalendar "schema" property within the link object.

The "STRUCTURED-DATA" "FMTTYPE" parameter is mapped on to a JSCalendar "contentType" property within the link object.

For example:

```
...
STRUCTURED-DATA;FMTTYPE=application/ld+json;
SCHEMA="https://schema.org/SportsEvent";
VALUE=TEXT:{\n
  "@context": "http://schema.org",\n
  "@type": "SportsEvent",\n
  "homeTeam": "Pittsburgh Pirates",\n
  "awayTeam": "San Francisco Giants"\n
}\n
...
```

maps to (with data truncated)

```
...
"links": {
  "1": {
    "@type" : "Link",
    "contentType": "application/ld+json",
    "schema": "https://schema.org/SportsEvent",
    "href": "data:base64;ewogICAgICAgICJAY29udGV4dCI6IC..."
  }
}
...
```

[4.36.](#) SUMMARY

Copy the value into the JSCalendar "title" property.

Mapping parameters:

ALTREP No mapping.

LANG Use the "locale" property.

[4.37.](#) 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".

4.38. UID

Copy the value into the JSCalendar "uid" property.

4.39. URL

Maps to a Link object in the JSCalendar object's "links" property, with the URL property value mapped to the Link "href" property.

5. Translating iCalendar Recurrences

5.1. Translating iCalendar Recurrences: Simple objects with overrides

A simple object with overrides will be converted to a jsCalendar master event with the rules, recurrence dates and exclusion dates translated appropriately.

Overrides MUST be mapped on to a jsCalendar patch object and added to the "recurrenceOverrides" property of the master event with the key being the value of the iCalendar RECURRENCE-ID translated to a json format.

Any override property with the same value as the master SHOULD be omitted. Remaining properties MAY be added in full. Where appropriate, differences SHOULD be expressed as a patch.

This can result in a significant reduction in size for objects with small changes to overrides, for example changing the participation status of an attendee.

5.2. Translating iCalendar Recurrences: Overrides with no master

When inviting an attendee to a single instance of a recurring event, only that override should be sent to the attendee. In this case the override should be a complete jsCalendar object with the type set to the type of the master.

Additionally, there MUST be a recurrenceId property set to the value of the recurrence id for that override. If the timezone of the start of the instance is different from the master value, then there must also be a "recurrenceIdTimeZone" property set to the start timezone of the master.

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

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.

```
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": {
      "1": {
        "@type": "Link",
        "rel": "enclosure",
        "contentType": "text/plain",
        "href": "http://example.org/doc1.txt"
      },
      "2": {
        "@type": "Link",
        "rel": "enclosure",
        "contentType": "text/plain",
        "href": "http://example.org/doc2.txt"
      },
      "3": {
        "@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
    }
}
}
]
}

```

6.2. Simple event with CONTACT

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

```

BEGIN:VCALENDAR
CALSCALE:GREGORIAN
PRODID:-//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",
  "prodId": "-//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,\n  +1-919-555-1234",
          "links": {

```



```

        "1": {
            "@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"
}
]
}

```

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.

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

```

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,
        "links": {
          "1": {
            "@type": "Link",
            "href": "http://example.org/vcards/vbede.vcf",
            "rel": "alternate"
          }
        },
        "roles": {
          "attendee": true
        },
        "scheduleStatus": "1.2"
      }
    },
    "replyTo": {
      "imip": "mailto:douglm@example.org"
    }
  },
  "2020-05-24T12:00:00": {
    "participants": {
      "daae4cf-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"
}
}
}
]
}

```

7. Translating JSCalendar properties to iCalendar components

This section lists 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.

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.


```
{
  "@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
```

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

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


```
{
  "@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
```

[7.3.](#) jstask

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

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


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

maps to

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

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

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.


```
{
  "@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
```

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.

7.5. alerts

Each member of a JSCalendar alerts property maps to a [RFC5545] VALARM component. Only display and email alarms are allowed in JSCalendar.

7.5.1. action

The JSCalendar "alert" property maps to the [RFC5545] ACTION property. The value SHOULD be the uppercased version if the JSCalendar "alert" property.

For example:


```
...  
  "action": "display",  
  ...
```

maps to

```
ACTION:DISPLAY
```

and

```
...  
  "action": "email",  
  ...
```

maps to

```
ACTION:EMAIL
```

7.5.2. trigger

A JSCalendar trigger with a type of "AbsoluteTrigger" maps on to a [\[RFC5545\]](#) TRIGGER property with a "VALUE" parameter of "DATE-TIME" and a value taken from the JSCalendar "when" property.

For example:

```
"trigger": {  
  "@type": "AbsoluteTrigger",  
  "when": "20210315T133000Z"  
}
```

maps to

```
TRIGGER;VALUE=DATE-TIME:20210315T133000Z
```

A JSCalendar trigger with a type of "OffsetTrigger" maps on to a [\[RFC5545\]](#) TRIGGER property with a duration value taken from the JSCalendar "offset" property.

If the JSCalendar trigger has a "relativeTo" property with the value "end" then the [\[RFC5545\]](#) TRIGGER property will have a RELATED=END parameter.

For example:


```
"trigger": {  
  "@type": "OffsetTrigger",  
  "offset": "-P2D",  
  "relativeTo": "end"  
}
```

maps to

```
TRIGGER;RELATED=END:-P2D
```

and

```
"trigger": {  
  "@type": "OffsetTrigger",  
  "offset": "-PT30M"  
}
```

maps to

```
TRIGGER:-PT30M
```

7.5.3. todo

Need to deal with "acknowledged" and "relatedTo". Also in the icalendar to jscalendar.

8. Translating JSCalendar properties to iCalendar properties

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

8.1. categories

Each member of the JSCalendar "categories" property maps on to a [\[draft-ietf-calext-eventpub-extensions\]](#) CONCEPT property with the value being the key of each member.

For example:


```
...
"categories": {
  "http://example.com/event-types/arts/music": true,
  "http://example.com/performance-types/arts/live": true
}
...
```

maps to

```
...
CONCEPT:http://example.com/event-types/arts/music
CONCEPT:http://example.com/performance-types/arts/live
...
```

8.2. created

The JSCalendar "created" property maps on to a [\[RFC5545\]](#) CREATED property with the value being the [\[RFC5545\]](#) UTC date-time derived from the value of the property.

For example:

```
...
"created": "2021-03-15T13:30"00Z"
...
```

maps to

```
...
CREATED:20210315T133000Z
...
```

8.3. duration

The JSCalendar "duration" property is only valid for event objects. Copy the JSCalendar "duration" property in to the [\[RFC5545\]](#) DURATION property.

For example:


```
...  
"duration": "PT1H"  
...
```

maps to

```
...  
DURATION:PT1H  
...
```

[8.4.](#) estimatedDuration

The JSCalendar "estimatedDuration" property is only valid for task objects. Copy the JSCalendar "estimatedDuration" property in to the [\[RFC5545\]](#) ESTIMATED-DURATION property.

For example:

```
...  
"estimatedDuration": "PT18H"  
...
```

maps to

```
...  
ESTIMATED-DURATION:PT18H  
...
```

[8.5.](#) keywords

Each member of the JSCalendar "categories" property maps on to a [\[RFC5545\]](#) CATEGORIES property with the value being the key of each member.

For example:


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

maps to

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

or alternatively

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

8.6. locations

JSCalendar locations should be mapped on to [\[draft-ietf-calext-eventpub-extensions\]](#) VLOCATION components. Additionally, for backwards compatibility, a location should be mapped on to a [\[RFC5545\]](#) LOCATION property. This property should be mapped from the only location or the one related to the start.

8.7. participants

JSCalendar participants will be mapped on to different iCalendar properties and components depending on their jsCalendar role values.

A participant with a role containing "contact" MUST be mapped on to an iCalendar CONTACT property and SHOULD also be mapped on to a [\[draft-ietf-calext-eventpub-extensions\]](#) PARTICIPANT component which provides a better mapping.

A participant with a role containing "owner" MUST be mapped on to an iCalendar ORGANIZER property and SHOULD also be mapped on to a [\[draft-ietf-calext-eventpub-extensions\]](#) PARTICIPANT component which provides a better mapping.

A participant with a role containing any of "attendee", "optional" or "informational" MUST be mapped on to an iCalendar ATTENDEE property and SHOULD also be mapped on to a

[[draft-ietf-calext-eventpub-extensions](#)]PARTICIPANT component which provides a better mapping.

A more complete mapping may be achieved by creating a [[draft-ietf-calext-eventpub-extensions](#)]PARTICIPANT component.

For all properties the participants jsCalendar "language" property, if present, is mapped on to the iCalendar "LANG" property parameter.

For all properties if the participant contains a jsCalendar "link" with a "rel" of "alternate" then the value of the link is used for the iCalendar "ALTREP" property parameter.

Where do we get the cua?

9. Security Considerations

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

10. IANA Considerations

None.

11. 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.

12. References

12.1. Normative References

[[draft-ietf-calext-eventpub-extensions](#)]
"Event Publishing Extensions to iCalendar",
<<https://tools.ietf.org/html/draft-ietf-calext-eventpub-extensions>>.

[[draft-ietf-calext-ical-relations](#)]
"Support for iCalendar Relationships",
<<https://tools.ietf.org/html/draft-ietf-calext-ical-relations>>.

[[draft-ietf-calext-valarm-extensions](#)]

"VALARM Extensions for iCalendar",
<<https://tools.ietf.org/html/draft-ietf-calext-valarm-extensions>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC2397] Masinter, L., "The "data" URL scheme", [RFC 2397](#), DOI 10.17487/RFC2397, August 1998, <<https://www.rfc-editor.org/info/rfc2397>>.

[RFC5545] Desruisseaux, B., Ed., "Internet Calendaring and Scheduling Core Object Specification (iCalendar)", [RFC 5545](#), DOI 10.17487/RFC5545, September 2009, <<https://www.rfc-editor.org/info/rfc5545>>.

[RFC5546] Daboo, C., Ed., "iCalendar Transport-Independent Interoperability Protocol (iTIP)", [RFC 5546](#), DOI 10.17487/RFC5546, December 2009, <<https://www.rfc-editor.org/info/rfc5546>>.

[RFC6638] Daboo, C. and B. Desruisseaux, "Scheduling Extensions to CalDAV", [RFC 6638](#), DOI 10.17487/RFC6638, June 2012, <<https://www.rfc-editor.org/info/rfc6638>>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", [RFC 7159](#), DOI 10.17487/RFC7159, March 2014, <<https://www.rfc-editor.org/info/rfc7159>>.

[RFC7265] Kewisch, P., Daboo, C., and M. Douglass, "jCal: The JSON Format for iCalendar", [RFC 7265](#), DOI 10.17487/RFC7265, May 2014, <<https://www.rfc-editor.org/info/rfc7265>>.

[RFC7986] Daboo, C., "New Properties for iCalendar", [RFC 7986](#), DOI 10.17487/RFC7986, October 2016, <<https://www.rfc-editor.org/info/rfc7986>>.

12.2. Informative References

[[draft-apthorp-ical-tasks](#)]

"Task Extensions to iCalendar",
<<https://tools.ietf.org/html/draft-apthorp-ical-tasks>>.

[[draft-ietf-calext-jscalendar](#)]
 "Task Extensions to iCalendar",
 <<https://tools.ietf.org/html/draft-ietf-calext-jscalendar>>.

[Appendix A](#). Outdated document sections

[A.1](#). Note

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

[A.2](#). JSTask

A JSTask object maps to the iCalendar VTOD0 component type [[RFC5545](#)]. The following tables maps the JSTask-specific properties to iCalendar:

Property	iCalendar counterpart
due	Maps to the DUE property. See Appendix A.4.1 .
statusUpdatedAt	COMPLETED property. The JSTask status property MUST have value "completed".
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].

Table 1: Mapping JSTask properties

[A.3](#). JSGroup

A JSGroup maps to a iCalendar VCALENDAR containing VEVENT or VTOD0 components.

Property	iCalendar counterpart
entries	VEVENT and VTOD0 components embedded in a VCALENDAR component.
source	SOURCE property.

Table 2: Mapping JSGroup properties

A.4. 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 VTOD0, "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. _relativeTo_ and _offset_ map to the TRIGGER property.
color	COLOR property, as specified in [RFC7986] .
created	CREATED property.
description	DESCRIPTION property.
descriptionContentType	Implementation-specific.
excluded	EXDATE property.
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 _href_. The FMTTYPE parameter maps to _type_, the SIZE parameter to _size_. Mapping other properties is implementation-specific.
locale	LANGUAGE parameter of the SUMMARY or DESCRIPTION property.
localizations	Implementation-specific.
locations	See Appendix A.4.2 .
method	METHOD property of the embedding VCALENDAR.
participants	See Appendix A.4.3 .
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.
recurrenceRule	RRULE property. For all-day calendar objects, map the _until_ 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.

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 Appendix A.4.1 .
status	STATUS property.
timeZone	Maps to the TZID parameter. See Appendix A.4.1 .
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.
useDefaultAlerts	Implementation-specific.
virtualLocations	See Appendix A.4.2 .

Table 3: Translation between JSCalendar and iCalendar

[A.4.1](#). 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:

- o The "start" ("due") property value has zero time, or is not set.

- o The "duration" ("estimatedDuration") property value has zero time, or is a multiple of days or weeks, or is not set.
- o The "timeZone" property value is null, or is not set.
- o The "showWithoutTime" property value is not set or "true".
- o The recurrenceRule.frequency is at most daily.
- o All recurrenceOverrides recurrence ids have zero time

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

- o With local time and time zone reference, if the "timeZone" property value is set and does not equal "Etc/UTC".
- o With UTC time, if the "timeZone" property value equals "Etc/UTC".
- o With local time, if the "timeZone" property value is null or not set.

A.4.2. 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.
name	LOCATION property value.
rel	Implementation-specific.
timeZone	Implementation-specific.
uri	The LOCATION ALTREP parameter.

Table 4: 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.
+-----+-----+	

Table 5: Mapping virtualLocation properties

[A.4.3.](#) 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
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.

Table 6: Mapping Participant properties

[A.5.](#) 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.

Authors' Addresses

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

Email: neilj@fastmailteam.com
URI: <https://www.fastmail.com>

Robert Stepanek
FastMail
PO Box 234
Collins St West
Melbourne VIC 8007
Australia

Email: rsto@fastmailteam.com
URI: <https://www.fastmail.com>

Michael Douglass
Bedework Commercial Services
226 3rd Street
Troy, NY 12180
United States of America

Email: mdouglass@bedework.com
URI: <http://bedework.com>

