

JSCalendar: Converting from and to iCalendar

Working Draft Standard

Warning for drafts

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

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

© 2022 The Calendaring and Scheduling Consortium, Inc.

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

The Calendaring and Scheduling Consortium, Inc.

4390 Chaffin Lane
McKinleyville
California 95519
United States of America

copyright@calconnect.org
www.calconnect.org

Contents

Abstract.....	v
Introduction.....	vi
Motivation.....	vi
Scope and caveats.....	vi
Notational Conventions.....	vii
1. Scope.....	1
2. Normative references.....	1
3. Terms and definitions.....	2
4. iCalendar pre-processing.....	2
5. Translating iCalendar components to JSCalendar.....	2
5.1. VALARM.....	2
5.2. VCALENDAR.....	5
5.3. VEVENT.....	5
5.4. VLOCATION.....	6
5.5. VTIMEZONE, STANDARD, DAYLIGHT.....	6
5.6. VTODO.....	7
6. Translating iCalendar properties to JSCalendar.....	7
6.1. ATTACH.....	7
6.2. ATTENDEE.....	7
6.3. CALSCALE.....	9
6.4. CATEGORIES.....	9
6.5. CLASS.....	10
6.6. COLOR.....	10
6.7. COMMENT.....	10
6.8. COMPLETED.....	10
6.9. CONCEPT.....	11
6.10.CONFERENCE.....	11
6.11.CONTACT.....	11
6.12.CREATED.....	12
6.13.DESCRPTION.....	12
6.14.DTEND, DTSTART, DUE, DURATION.....	13
6.15.ESTIMATED-DURATION.....	14
6.16.EXDATE.....	14
6.17.EXRULE.....	15
6.18.DTSTAMP and LAST-MODIFIED.....	15
6.19.GEO.....	15
6.20.IMAGE.....	15
6.21.LOCATION.....	15
6.22.METHOD.....	16
6.23.ORGANIZER.....	16
6.24.PERCENT-COMPLETE.....	16
6.25.PRIORITY.....	18
6.26.PROID.....	18
6.27.RECURRENCE-ID.....	18
6.28.RELATED-TO.....	18
6.29.REQUEST-STATUS.....	19
6.30.RESOURCES.....	19
6.31.RDATE.....	19
6.32.RRULE.....	19
6.33.SEQUENCE.....	22
6.34.STATUS.....	22
6.35.STRUCTURED-DATA.....	22
6.36.SUMMARY.....	23
6.37.TRANSP.....	23
6.38.UID.....	23
6.39.URL.....	23
7. Translating non-standard iCalendar to JSCalendar.....	24

7.1. urn:ietf:rfcXXXX#components..... 24

7.2. urn:ietf:rfcXXXX#properties..... 24

7.3. urn:ietf:rfcXXXX#parameters..... 25

7.4. urn:ietf:rfcXXXX#prop..... 25

8. Translating iCalendar Recurrences..... 26

8.1. Translating iCalendar Recurrences: Simple objects with overrides..... 26

8.2. Translating iCalendar Recurrences: Overrides with no master..... 26

9. Translating iCalendar: Further examples..... 26

9.1. Recurring event with ATTACH..... 26

9.2. Simple event with CONTACT..... 28

9.3. Simple event with RESOURCES..... 29

9.4. Recurring event. Attendees only in overrides..... 29

10. Translating JSCalendar objects to iCalendar..... 32

10.1.Event..... 32

10.2.Group..... 32

10.3.Task..... 33

11. Translating JSCalendar properties to iCalendar..... 33

11.1.alerts..... 33

11.2.categories..... 36

11.3.color..... 36

11.4.created..... 37

11.5.duration..... 37

11.6.estimatedDuration..... 37

11.7.keywords..... 37

11.8.Jocations..... 38

11.9.participants..... 39

11.10.imezones..... 39

12. Translating non-standard JSCalendar to iCalendar..... 40

12.1.New iCalendar properties..... 40

12.2.New iCalendar Parameters..... 43

13. JSCalendar base spec updates..... 44

14. Security Considerations..... 44

15. IANA Considerations..... 44

16. Acknowledgments..... 44

Bibliography..... 45

Abstract

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

Introduction

EDITORIAL NOTE —

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

Motivation

The JSCalendar [IETF RFC 8984](#) data format is used to represent calendar data, and is meant as an alternative to the widely deployed iCalendar [IETF RFC 5545](#) 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.

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.
- 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.

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

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

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 [IETF RFC 2119](#).

JSCalendar: Converting from and to iCalendar

1. Scope

2. Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IETF RFC 2119, S. BRADNER. *Key words for use in RFCs to Indicate Requirement Levels*. 1997. RFC Publisher. <https://www.rfc-editor.org/info/rfc2119>.

IETF RFC 2397, L. MASINTER. *The "data" URL scheme*. 1998. RFC Publisher. <https://www.rfc-editor.org/info/rfc2397>.

IETF RFC 4648, S. JOSEFSSON. *The Base16, Base32, and Base64 Data Encodings*. 2006. RFC Publisher. <https://www.rfc-editor.org/info/rfc4648>.

IETF RFC 5545, B. DESRUISSEAU (ed.). *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. 2009. RFC Publisher. <https://www.rfc-editor.org/info/rfc5545>.

IETF RFC 5546, C. DABOO (ed.). *iCalendar Transport-Independent Interoperability Protocol (iTIP)*. 2009. RFC Publisher. <https://www.rfc-editor.org/info/rfc5546>.

IETF RFC 6638, C. DABOO and B. DESRUISSEAU. *Scheduling Extensions to CalDAV*. 2012. RFC Publisher. <https://www.rfc-editor.org/info/rfc6638>.

IETF RFC 6868, C. DABOO. *Parameter Value Encoding in iCalendar and vCard*. 2013. RFC Publisher. <https://www.rfc-editor.org/info/rfc6868>.

IETF RFC 7265, P. KEWISCH, C. DABOO and M. DOUGLASS. *jCal: The JSON Format for iCalendar*. 2014. RFC Publisher. <https://www.rfc-editor.org/info/rfc7265>.

IETF RFC 7493, T. BRAY (ed.). *The I-JSON Message Format*. 2015. RFC Publisher. <https://www.rfc-editor.org/info/rfc7493>.

IETF RFC 7986, C. DABOO. *New Properties for iCalendar*. 2016. RFC Publisher. <https://www.rfc-editor.org/info/rfc7986>.

IETF RFC 8259, T. BRAY (ed.). *The JavaScript Object Notation (JSON) Data Interchange Format*. 2017. RFC Publisher. <https://www.rfc-editor.org/info/rfc8259>.

IETF RFC 8984, N. JENKINS and R. STEPANEK. *JSCalendar: A JSON Representation of Calendar Data*. 2021. RFC Publisher. <https://www.rfc-editor.org/info/rfc8984>.

IETF RFC 9073, M. DOUGLASS. *Event Publishing Extensions to iCalendar*. 2021. RFC Publisher. <https://www.rfc-editor.org/info/rfc9073>.

IETF RFC 9074, C. DABOO. *"VALARM" Extensions for iCalendar*. 2021. RFC Publisher. <https://www.rfc-editor.org/info/rfc9074>.

Internet-Draft draft-ietf-calext-ical-relations-00, MICHAEL DOUGLASS. *Support for Icalendar Relationships*. 2016. <https://datatracker.ietf.org/doc/html/draft-ietf-calext-ical-relations-00>.

Internet-Draft draft-ietf-calext-jscalendarbis-00, NEIL JENKINS and ROBERT STEPANEK. *JSCalendar: A JSON Representation of Calendar Data*. 2022. <https://datatracker.ietf.org/doc/html/draft-ietf-calext-jscalendarbis-00>.

3. Terms and definitions

No terms and definitions are listed in this document.

4. 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 [IETF RFC 5545, Section 3.1](#).

iCalendar data uses an “escape” character sequence for text values and property parameter values. See [IETF RFC 5545, Section 3.1](#) and [IETF RFC 5545, Section 3.3](#) as well as [IETF RFC 6868](#).

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 [IETF RFC 8259, Section 7](#), 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 [IETF RFC 4648, Section 4](#), which is the base64 method used in [IETF RFC 5545](#).

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.

5. Translating iCalendar components to JSCalendar

This section is an alphabetic list of [IETF RFC 5545](#) components and how they are mapped to JSCalendar.

At present VFREEBUSY and VJOURNAL are not mapped in JSCalendar.

5.1. VALARM

An [IETF RFC 5545](#) 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 [IETF RFC 5545](#) 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.

Any ATTENDEE properties in the VALARM MUST be ignored.

An [IETF RFC 5545](#) 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 iCalendar DESCRIPTION property is required for both ACTION:EMAIL and ACTION:DISPLAY. Additionally, the iCalendar SUMMARY property is required for ACTION:EMAIL.

Store the SUMMARY in the optional alert title property and the DESCRIPTION is the alert description property.

The [IETF RFC 5545](#) example VALARMS will be mapped as follows, assuming they are all in the same event:

```

BEGIN:VEVENT
DTSTAMP:20200522T142047Z
DTSTART;TZID=America/New_York:20220622T120000

```

CC/WD 51026:2022

```
DURATION:PT1H
SUMMARY:event with alarms
UID:6252D6C40A8308BFE25BBEFsimple-alarms-1
BEGIN:VALARM
TRIGGER;VALUE=DATE-TIME:20220508T120000Z
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": "Group",
  "prodId": "//example.org//Example V3.14//EN",
  "entries": [
    {
      "@type": "Event",
      "alerts": {
        "1": {
          "@type": "Alert",
          "action": "display",
          "trigger": {
            "@type": "AbsoluteTrigger",
            "when": "2022-05-08T12:00:00Z"
          }
        },
        "2": {
          "@type": "Alert",
          "action": "display",
          "description": "Breakfast meeting with executive\n
team at 8:30 AM EST.",
          "trigger": {
            "@type": "OffsetTrigger",
            "offset": "-PT30M"
          }
        },
        "3": {
```

```

    "@type": "Alert",
    "action": "email",
    "title": "*** 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.",
    "trigger": {
      "@type": "OffsetTrigger",
      "offset": "-P-2D",
      "relativeTo": "end"
    }
  }
},
"created": "2022-05-02T20:04:12Z",
"start": "2022-06-22T12:00:00",
"timeZone": "America/New_York",
"duration": "PT1H",
"title": "event with alarms",
"uid": "6252D6C40A8308BFE25BBEFsimple-alarms-1"
}
]
}

```

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

5.2. VCALENDAR

A [IETF RFC 5545](#) VCALENDAR component may be mapped to a JSCalendar object with a type of "Group".

```

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

```

maps to

```

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

```

Note that a single Event or Task MAY be converted without a surrounding Group 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.

5.3. VEVENT

A [IETF RFC 5545](#) VEVENT component is mapped to a JSCalendar object with a type of "Event".

```

BEGIN: VEVENT
...
END: VEVENT

```

maps to

```

{

```

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

5.4. VLOCATION

A [IETF RFC 9073](#) 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": "Event",
  ...
  "locations": {
    "1": {
      "@type": "Location",
      ...
    }
  }
}
```

5.5. VTIMEZONE, STANDARD, DAYLIGHT

A [IETF RFC 5545](#) 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

- 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.

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

maps to

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

```

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

```

5.6. VTODO

A [IETF RFC 5545](#) VTODO component is mapped to a JSCalendar object with a type of “Task”.

```
BEGIN: VTODO
```

```

...
END: VTODO

```

maps to

```

{
  "@type": "Task",
  ...
}

```

6. Translating iCalendar properties to JSCalendar

This section is an alphabetic list of [IETF RFC 5545](#) and [IETF RFC 7986](#) properties and how they are mapped to JSCalendar.

6.1. ATTACH

A [IETF RFC 5545](#) 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 [Clause 9.1](#).

6.2. ATTENDEE

An [IETF RFC 5545](#) 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": "mailto: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	<p>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.</p> <p>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.</p> <p>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.</p>
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 IETF RFC 6638 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.

6.3. CALSCALE

A [IETF RFC 5545](#) CALSCALE has no equivalence in JSCalendar. It is ignored.

6.4. CATEGORIES

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

...
CATEGORIES: APPOINTMENT, EDUCATION

CC/WD 51026:2022

CATEGORIES:MEETING

...

maps to

...

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

6.5. 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

6.6. COLOR

The COLOR property is defined in [IETF RFC 7986](#) and maps to the JSCalendar "color" property. Copy the value verbatim.

6.7. COMMENT

There is no direct mapping for this property which may appear multiple times in [IETF RFC 5545](#).

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.

6.8. 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",
```

...

6.9. CONCEPT

This [Internet-Draft draft-ietf-calext-ical-relations-00](#) 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
}
...
```

6.10. 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.

6.11. CONTACT

A CONTACT property is mapped on to a participant object with a “roles” property of “contact”. The value of the property is used to set the jscalendar participant “name” property.

QUESTIONS — what to use for the id? I have a uid but that’s my implementation. There may be multiple CONTACT properties. If we create PARTICIPANT objects going the other way — how to link them — “name” == CONTACT value?

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 [Clause 9.2](#).

6.12. 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": "Event",
  ...
  "created": "1996-03-29T13:30"00Z",
  ...
}
```

6.13. DESCRIPTION

Copy the value, preprocessed according to [Clause 4](#) 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 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": "Event",
  ...
  "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.",
  ...
}
```

}

6.14. 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 1 — 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": "Event",
  ...
  "start": "2017-03-15T15:00:00",
  "timeZone": "America/New_York",
  "duration": "PT1H"
  ...
}
```

EXAMPLE 2 — DTSTART and DTEND in different timezone

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

maps to

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

```

...
"start": "2017-03-15T15:00:00",
"timeZone": "America/New_York",
"duration": "PT7H"
...
"locations": {
  "1": {
    "@type": "location",
    "relatedTo": "end",
    "timeZone": "America/Los_Angeles"
  }
}
}

```

EXAMPLE 3 — 3 day event

```

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

maps to

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

```

6.15. ESTIMATED-DURATION

Copy the ESTIMATED-DURATION value into the JSCalendar “estimatedDuration” property.

For example:

```

...
ESTIMATED-DURATION:PT18H
...

maps to

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

```

6.16. 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”.

6.17. EXRULE

Maps to the “excludedRecurrenceRules” property. Also see [Clause 6.17](#).

6.18. DTSTAMP and LAST-MODIFIED

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

Not a scheduling entity	The IETF RFC 5545 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.

6.19. 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:”.

6.20. 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.

6.21. LOCATION

If any [IETF RFC 9073](#) “VLOCATION” components are present, then the [IETF RFC 5545](#) “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 [IETF RFC 5545](#) “LOCATION” property.

Mapping parameters:

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

6.22. METHOD

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

6.23. 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 [Clause 6.2](#) 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.

6.24. PERCENT-COMPLETE

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

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

maps to

```
{
  "@type": "Task",
  "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 VTODD 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
PROIDID:-//ABC Corporation//NONSGML My Product//EN
METHOD:REPLY
BEGIN:VTODD
...
ATTENDEE:mailto:douglm@example.org
PERCENT-COMPLETE:39
END:VTODD
END: VCALENDAR
```


maps to

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

}

6.25. PRIORITY

Simply copy value into the JSCalendar “priority” property.

6.26. PRODIG

For a vcalendar Group object with multiple Event and/or Task object the [IETF RFC 5545](#) VCALENDAR PRODIG is mapped to a JSCalendar “prodid” property in the group.

When mapping to a single Event and/or Task object the [IETF RFC 5545](#) VCALENDAR PRODIG is mapped to a JSCalendar “prodid” property in the group

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

maps to

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

6.27. RECURRENCE-ID

Refer to [Clause 8](#) for information on mapping recurrence ids.

6.28. 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": "Event",
  "prodid": "-//ABC Corporation//NONSGML My Product//EN",
  ...
}

```

6.29. REQUEST-STATUS

Copy the value into the JSCalendar “requestStatus” property.

6.30. 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 [Clause 9.3](#).

6.31. 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.

6.32. 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
}],
...

```

6.33. SEQUENCE

Copy the value into the JSCalendar “sequence” property.

6.34. 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.

6.35. 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 [IETF RFC 2397](#) data uri either as plain text or by encoding as a base64 value. If plain text the value may need escaping as per [IETF RFC 2397](#).

VALUE= BINARY Copy the value as a [IETF RFC 2397](#) 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” “FMPTYPE” parameter is mapped on to a JSCalendar “contentType” property within the link object.

For example:

```
...
STRUCTURED-DATA;FMPTYPE=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..."
  }
}
...
```

6.36. SUMMARY

Copy the value into the JSCalendar “title” property.

Mapping parameters:

ALTREP	No mapping.
LANG	Use the “locale” property.

6.37. 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”.

6.38. UID

Copy the value into the JSCalendar “uid” property.

6.39. URL

Maps to a Link object in the JSCalendar object’s “links” property, with the URL property value mapped to the Link “href” property.

7. Translating non-standard iCalendar to JSCalendar

iCalendar data may contain entries for which no standard property or object type exists in JSCalendar. This mainly occurs for experimental properties, parameter and components in iCalendar (X-extensions, see [IETF RFC 5545, Section 3.8.8.2](#)), but may be the case for other iCalendar data, too. Typically, the semantics or value types of such entries are unknown.

This section defines new properties, by which implementations SHOULD map such data. Applications that have more insight into the semantics of such iCalendar entries MAY choose to map differently.

NOTE The verbatim XXXX values in this section will be replaced with the RFC number of this document before publication. Also, the following properties require a redefinition of vendor-extension properties for JSCalendar. Currently, only domain names are allowed as prefix for vendor extensions. This document assumes that also URIs with the urn scheme are allowed.

7.1. urn:ietf:rfcXXXX#components

Type: JCalComp[] where JCalComp denotes a jCal-encoded component as defined in [IETF RFC 7265, Section 3.3](#)

This property is set on a JSCalendar object for which a standard mapping from an iCalendar component is defined. If set, it contains one or more iCalendar components that are direct children of the iCalendar component represented by the JSCalendar object.

```
BEGIN:VEVENT
UID:c2236fe3-4dc9-4b3b-8a18-cd8f29eca594
DTSTART:20220711T104800
BEGIN:X-COMP
UID:6dcff59c-d251-44c9-9010-a62cab390df0
END:X-COMP
END:VEVENT
```

maps to

```
{
  "@type": "Event",
  "uid": "c2236fe3-4dc9-4b3b-8a18-cd8f29eca594",
  "start": "2022-07-11T10:48:00",
  "urn:ietf:rfcXXXX#components": [
    ["x-comp", [
      ["uid", {}], "text", "6dcff59c-d251-44c9-9010-a62cab390df0"
    ]
  ]
}
```

7.2. urn:ietf:rfcXXXX#properties

Type: JCalProp[] where JCalProp denotes a jCal-encoded property as defined in [IETF RFC 7265, Section 3.4](#)

This property is set on a JSCalendar object for which a standard mapping from an iCalendar component is defined. If set, it contains one or more iCalendar properties that are set in the iCalendar component represented by the JSCalendar object.

```
BEGIN:VEVENT
UID:c2236fe3-4dc9-4b3b-8a18-cd8f29eca594
DTSTART:20220711T104800
X-PROP;X-PARAM=Bar:Foo
```


END:VEVENT

maps to

```
{
  "@type": "Event",
  "uid": "c2236fe3-4dc9-4b3b-8a18-cd8f29eca594",
  "start": "2022-07-11T10:48:00",
  "urn:ietf:rfcXXXX#properties": [
    ["x-prop", {
      "x-param": "Bar"
    }], "unknown", "Foo"]
  ]
}
```

7.3. urn:ietf:rfcXXXX#parameters

Type: `String[String]` denotes a JSON object encoding iCalendar parameters as defined in [IETF RFC 7265, Section 3.5](#)

This property is set on a JSCalendar object for which a standard mapping from an iCalendar property is defined. If set, it contains one or more iCalendar parameters that are set on the iCalendar property represented by the JSCalendar object.

```
...
ATTACH;X-PARAM=Bar;VALUE=URI:https://example.com/foo.zip
...
```

maps to

```
...
{
  "@type": "Link",
  "href": "https://example.com/foo.zip",
  "urn:ietf:rfcXXXX#parameters": {
    "x-param": "Bar"
  }
}
...
```

7.4. urn:ietf:rfcXXXX#prop

A short-form variant of the properties property defined in [Clause 7.2](#) MAY be used to map a single property from iCalendar. This aims to provide a compact representation for mapping an iCalendar property without parameters and for which no value type is known. It is defined as follows:

Name: `urn:ietf:rfcXXXX#prop/propname` where *propname* denotes the ASCII lower-case name ([IETF RFC 5545, Section 3.1](#)) of the mapped property.

Type: `String` where the string must adhere to the definitions of [IETF RFC 7265, Section 5](#).

The following example contains both the long-form and short-form mapping for illustration:

```
...
X-Foo:Bar
...
```

maps to short-form

```
...
"urn:ietf:rfcXXXX#prop/x-foo": "Bar"
...
```

or long-form

```
...
"urn:ietf:rfcXXXX#properties": [
  ["x-foo", { }, "unknown", "Bar"]
]
...
```

8. Translating iCalendar Recurrences

8.1. 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.

8.2. 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.

9. Translating iCalendar: Further examples

This section provides more complete examples of translating from [IETF RFC 5545](#) to JSCalendar.

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

9.1. 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
PROID:-//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
      }
    }
  }
}

```

9.2. Simple event with CONTACT

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

```

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-XXXX
END:VEVENT
END:VCALENDAR

```

translates to

```

{
  "@type": "Group",
  "prodId": "-//Example.org//Example V3.13.2//EN",
  "entries": [
    {
      "@type": "Event",

```

```

"participants": {
  "40288108-733187c1-0173-3188007b-00000001": {
    "@type": "Participant",
    "roles": {
      "contact": true
    },
    "description": "Jim Dolittle, ABC Industries,\
+1-919-555-XXXX",
    "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"
}
]
}

```

9.3. Simple event with RESOURCES

TBD

9.4. 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": "Group",
  "prodId": "//Example.org//Example V3.13.2//EN",
  "entries": [
    {
      "@type": "Event",
      "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": {
    "daaeae4cf-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"
  }
}

```

```

    }
  }
]
}

```

10. Translating JSCalendar objects to iCalendar

This section lists the JSCalendar objects that map to [IETF RFC 5545](#) components.

10.1. Event

A JSCalendar object with a type of “Event” is mapped on to a [IETF RFC 5545](#) VEVENT component.

If it is a single VEVENT then a [IETF RFC 5545](#) VCALENDAR component must surround it and the JSCalendar “prodid” property will be converted to a [IETF RFC 5545](#) PRODID.

```

{
  "@type": "Event",
  "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 Event or Task objects the surrounding [IETF RFC 5545](#) VCALENDAR object must have a [IETF RFC 5545](#) PRODID set from either the Group “prodid” or generated.

10.2. Group

A JSCalendar object with a type of “Group” is mapped on to a [IETF RFC 5545](#) VCALENDAR component.

```

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

```

maps to

```

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

```



```

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

```

10.3. Task

A JSCalendar object with a type of “Task” is mapped on to a [IETF RFC 5545](#) VTODO component.

If it is a single VTODO then a [IETF RFC 5545](#) VCALENDAR component must surround it and the JSCalendar “prodid” property will be converted to a [IETF RFC 5545](#) PRODID.

```

{
  "@type": "Task",
  "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 Event or Task objects the surrounding [IETF RFC 5545](#) VCALENDAR object must have a [IETF RFC 5545](#) PRODID set from either the Group “prodid” or generated.

11. Translating JSCalendar properties to iCalendar

This section is an alphabetic list of all JSCalendar top-level properties that map to [IETF RFC 5545](#) iCalendar.

11.1. alerts

Each member of a JSCalendar alerts property maps to a [IETF RFC 5545](#) VALARM component. Only display and email alarms are allowed in JSCalendar.

11.1.1. action

The JSCalendar “action” property maps to the [IETF RFC 5545](#) ACTION property. The value SHOULD be the uppercased version of the JSCalendar “alert” property.

For example:

```

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

```

maps to

```
ACTION:DISPLAY
```

and

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

maps to

```
ACTION:EMAIL
```

11.1.2. title and description

The iCalendar DESCRIPTION property is required for both ACTION:EMAIL and ACTION:DISPLAY. Additionally, the iCalendar SUMMARY property is required for ACTION:EMAIL.

Both of these are optional in a JSCalendar alert. When converting is there is a title property in the alert use that for the SUMMARY.

Similarly if a JSCalendar description property is present use that for the description. Otherwise use the alert title property and if that is absent use the event title.

11.1.3. trigger

A JSCalendar trigger with a type of "AbsoluteTrigger" maps on to a [IETF RFC 5545](#) 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 [IETF RFC 5545](#) 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 [IETF RFC 5545](#) 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

11.1.4. todo

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

11.1.5. Example event with alerts

```
{
  "@type": "Group",
  "prodId": "//Example.org//Example V3.14//EN",
  "entries": [
    {
      "@type": "Event",
      "alerts": {
        "1": {
          "@type": "Alert",
          "action": "display",
          "trigger": {
            "@type": "AbsoluteTrigger",
            "when": "2022-05-08T12:00:00Z"
          }
        },
        "2": {
          "@type": "Alert",
          "action": "display",
          "description": "Breakfast meeting with executive\nteam
                        at 8:30 AM EST.",
          "trigger": {
            "@type": "OffsetTrigger",
            "offset": "-PT30M"
          }
        },
        "3": {
          "@type": "Alert",
          "action": "email",
          "title": "**** 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.",
          "trigger": {
            "@type": "OffsetTrigger",
            "offset": "-P-2D",
            "relativeTo": "end"
          }
        }
      },
      "created": "2022-05-02T20:04:12Z",
      "start": "2022-06-22T12:00:00",
      "timeZone": "America/New_York",
      "duration": "PT1H",
      "title": "event with alarms",
      "uid": "6252D6C40A8308BFE25BBEFsimple-alarms-1"
    }
  ]
}
```

CC/WD 51026:2022

```
}  
maps to
```

```
BEGIN:VEVENT  
CREATED:20220502T200412Z  
DURATION:PT1H  
DTSTAMP:20220502T201921Z  
DTSTART;TZID=America/New_York:20220622T120000  
LAST-MODIFIED:20220502T201921Z  
SUMMARY:event with alarms  
UID:6252D6C40A8308BFE25BBEFsimple-alarms-1  
BEGIN:VALARM  
ACTION:DISPLAY  
TRIGGER;RELATED=START:-PT30M  
DESCRIPTION:event with alarms  
END:VALARM  
BEGIN:VALARM  
ACTION:DISPLAY  
TRIGGER;VALUE=DATE-TIME:20220508T120000Z  
DESCRIPTION:event with alarms  
END:VALARM  
BEGIN:VALARM  
ACTION:EMAIL  
TRIGGER;RELATED=END:-P2D  
DESCRIPTION:A draft agenda needs to be sent out to the attendees  
to the weekly managers meeting (MGR-LIST). Attached is a pointer  
to the document template for the agenda file.  
SUMMARY:*** REMINDER: SEND AGENDA FOR WEEKLY STAFF MEETING ***  
END:VALARM  
END:VEVENT
```

11.2. categories

Each member of the JSCalendar “categories” property maps on to a [IETF RFC 9073](#) 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  
...
```

11.3. color

The value is copied verbatim to the [IETF RFC 7986](#) “COLOR” property.

11.4. created

The JSCalendar “created” property maps on to a [IETF RFC 5545](#) CREATED property with the value being the [IETF RFC 5545](#) UTC date-time derived from the value of the property.

For example:

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

maps to

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

11.5. duration

The JSCalendar “duration” property is only valid for event objects. Copy the JSCalendar “duration” property in to the [IETF RFC 5545](#) DURATION property.

For example:

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

maps to

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

11.6. estimatedDuration

The JSCalendar “estimatedDuration” property is only valid for task objects. Copy the JSCalendar “estimatedDuration” property in to the [IETF RFC 5545](#) ESTIMATED-DURATION property.

For example:

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

maps to

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

11.7. keywords

Each member of the JSCalendar “keywords” property maps on to a [IETF RFC 5545](#) 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  
...
```

11.8. locations

JSCalendar locations map to [IETF RFC 9073](#) VLOCATION components.

Additionally, for backwards compatibility, one location should be mapped on to a [IETF RFC 5545](#) LOCATION property.

11.8.1. coordinates

TODO. Need to decide if to use GEO or define new GEO-URI.

11.8.2. description

The “description” property maps to the DESCRIPTION property value of the VLOCATION.

11.8.3. links

TODO. First need to define top-level links property mapping.

11.8.4. locationTypes

The keys of the “locationTypes” property map to the LOCATION-TYPE property value of the VLOCATION. The keys MUST be separated by the COMMA character (U+002c) and SHOULD sort in ascending alphabetical order.

11.8.5. name

The “name” property maps to the NAME property value of the VLOCATION.

11.8.6. relativeTo

The “relativeTo” property maps to the RELATED-TO property value of the VLOCATION. TODO need updated definition of the RELATED-TO draft.

11.8.7. timeZone

The “timeZone” property maps to the TZID property value of the VLOCATION.

If the TimeZoneId value matches a name from the IANA Time Zone Database [TZDB](#) then this value MUST be set in the TZID property.

If the TimeZoneId identifies a custom TimeZone in the JSCalendar object, then the TZID property value MUST be set to the “tzId” property value of the custom TimeZone object, and its related VTIMEZONE added to the VCALENDAR component that encloses the VLOCATION.

11.8.8. uid

The “uid” property maps to the UID property value of the VLOCATION. The identifier of the Location object in the enclosing “locations” property maps to the JMAP-ID property parameter. The parameter MAY be omitted if the identifier of the Location matches the “uid” value.

11.9. 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 [IETF RFC 9073](#) 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 [IETF RFC 9073](#) 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 [IETF RFC 9073](#) PARTICIPANT component which provides a better mapping.

A more complete mapping may be achieved by creating a [IETF RFC 9073](#) 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?

11.10. timezones

The JSCalendar TimeZone objects within a “timezones” property are mapped on to [IETF RFC 5545](#) VTIMEZONE components within the surrounding VCALENDAR component. Each mapped TimeZone MUST only appear once.

```
{
  "@type": "Event",
  ...
  "timezones": {
    "/Example/Somewhere": {
      "@type": "TimeZone",
      "tzId": "Example/Somewhere",
      ...
    }
  }
}
```

```

    },
    "/Example/Somewhere-else": {
      "@type": "TimeZone",
      "tzId": "Example/Somewhere-else",
      ...
    }
  }
}

```

maps to

```

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

```

When converting multiple Event or Task objects the surrounding [IETF RFC 5545](#) VCALENDAR object must have a [IETF RFC 5545](#) PRODID set from either the Group “prodid” or generated.

12. Translating non-standard JSCalendar to iCalendar

JSCalendar objects may contain properties for which no standard mapping to iCalendar is defined. This may occur for vendor-extension properties ([Internet-Draft draft-ietf-calext-jscalendarbis-00](#), Section 3.3) or any other property for which there no standard mapping is defined.

This section defines properties and parameters with which implementations SHOULD map such data. Applications that have more insight into the semantics of such iCalendar entries MAY choose to map differently.

NOTE The verbatim XXXX values in this section will be replaced with the RFC number of this document before publication. Also, the following properties require a redefinition of vendor-extension properties for JSCalendar. Currently, only domain names are allowed as prefix for vendor extensions. This document assumes that also URIs with the urn scheme are allowed.

12.1. New iCalendar properties

12.1.1. X-RFCXXXX-JSPROP Property

Property name	X-RFCXXXX-JSPROP
Purpose	This property embeds JSCalendar properties in iCalendar.
Value type	URI, also see Format Definition for value restrictions.
Conformance	The property can be specified multiple times in a calendar component.
Property parameters	The X-RFCXXXX-JSNAME parameter MUST be set exactly once for this property. The VALUE parameter MAY be set once, in which case its value MUST be URI. Other IANA and non-standard property parameters can be specified on this property.
Description	This property maps a JSCalendar property to iCalendar. The property name is defined using the X-RFCXXXX-JSNAME parameter (Clause

[12.2.1](#)). The property value is encoded as a URL using the data scheme [IETF RFC 2397](#). The media type MUST be specified as application/json and MUST NOT define parameters. The property value MAY be any [IETF RFC 8259](#) Javascript Object Notation (JSON) value restricted by the [IETF RFC 7493](#) I-JSON profile, be that a JSON object, array, string, number, true or false.

Format
definition

This property is defined by the following notation:

```
jsprop = "X-RFCXXXX-JSPROP" jsprop-param ":" jsprop-value
CRLF

jsprop-value = "data:application/json" [";base64"] "," data
; data is defined in RFC 2397, section 3

jsprop-param = *(
; The following is MANDATORY and MUST NOT
; occur more than once
jsname-param /
;
; The following is OPTIONAL,
; but MUST NOT occur more than once.
;
(";" "VALUE" "=" "URI" ) /
;
; The following is OPTIONAL,
; and MAY occur more than once.
;
(";" other-param)
;
)
```

Example(s)

```
;
; encodes the property
; ...
; "foo": {
;   "bar": 1234
; }
;
X-RFCXXXX-JSPROP;X-RFCXXXX-JSNAME="foo"
: data:application/json;base64,eyJiYXQiOiAxMjM0fQ==
```

12.1.2. X-RFCXXXX-PROP Property

Property name	X-RFCXXXX-PROP
Purpose	This property embeds simple-typed JSCalendar properties in iCalendar.
Value type	Default is none. It MAY be set to BOOLEAN, FLOAT or INTEGER.
Conformance	The property can be specified multiple times in a calendar component.
Property parameters	The X-RFCXXXX-JSNAME parameter MUST be set exactly once for this property. The VALUE parameter MAY be set once, in which case its value MUST be BOOLEAN, FLOAT or INTEGER. Other IANA and non-standard property parameters can be specified on this property.

- Description This property maps a simple-valued JSCalendar property to iCalendar. The property name is defined using the X-RFCXXXX-JSNAME parameter. ([Clause 12.2.1](#)). The property value is set by the following rules:
- If the JSON property value is a string and its contents solely consists of characters defined in the value ABNF of [IETF RFC 5545, Section 3.1](#), then the iCalendar property value is a verbatim copy of the JSON property value. The VALUE parameter MUST NOT be set.
 - If the JSON property value is a number without fractional and exponent, then the iCalendar property value is a verbatim copy of the JSON property value. The VALUE parameter SHOULD be INTEGER if the JSON property is known to only allow integer values. Otherwise the parameter value MUST be FLOAT.
 - If the JSON property value is a number with fractional or exponent, then the iCalendar property value is the FLOAT representation of of the JSON property value (note that iCalendar FLOAT does not support exponents). The VALUE parameter MUST be FLOAT.
 - If the JSON property value is verbatim true then the iCalendar property value is TRUE, or false and FALSE, respectively. The VALUE parameter MUST be BOOLEAN.
 - If none of the above rules match, then the JSCalendar property MUST NOT map to the X-RFCXXXX-PROP property. Instead, it MAY map to the [X-RFCXXXX-JSPROP](#) property.

Format definition This property is defined by the following notation:

```

jsprop-short = "X-RFCXXXX-PROP"
              (
                jsprop-string /
                jsprop-int /
                jsprop-float /
                jsprop-bool
              )

jsprop-string = *(
                  ; The following is MANDATORY and MUST NOT
                  ; occur more than once
                  jsname-param /
                  ;
                  ; The following MUST NOT occur
                  ;
                  (";" "VALUE" "=" valuetype ) /
                  ;
                  ; The following is OPTIONAL,
                  ; and MAY occur more than once.
                  ;
                  (";" other-param)
                  ;
                )
              ":" value ; defined in RFC5545, section 3.1

jsprop-int =  *(
                ; The following are MANDATORY and each
                ; occur more than once
                jsname-param /
                ( "VALUE" "=" "INTEGER" )
                ;
                ; The following is OPTIONAL,
                ; and MAY occur more than once.
              )

```

```

;
(";" other-param)
;
)
": " integer ; defined in RFC5545, section

```

3.3.8

```

jsprop-float = *(
; The following are MANDATORY and each
MUST NOT
; occur more than once
jsname-param /
( "VALUE" "=" "FLOAT" )
;
; The following is OPTIONAL,
; and MAY occur more than once.
;
(";" other-param)
;
)
": " float ; defined in RFC5545, section

```

3.3.7

```

jsprop-bool = *(
; The following are MANDATORY and each
MUST NOT
; occur more than once
jsname-param /
( "VALUE" "=" "BOOLEAN" )
;
; The following is OPTIONAL,
; and MAY occur more than once.
;
(";" other-param)
;
)
": " boolean ; defined in RFC5545, section

```

3.3.2

```

Example(s)
;
; encodes the property
; ...
; "foo": 12.3
;
X-RFCXXXX-PROP
;X-RFCXXXX-JSNAME="foo"
;VALUE=FLOAT
:12.3

```

12.2. New iCalendar Parameters

12.2.1. X-RFCXXXX-JSNAME Parameter

Parameter name	X-RFCXXXX-JSNAME
Purpose	This parameter assigns a JSCalendar property name to JSCalendar data embedded in an iCalendar property.

Format definition	<code>jsname-param = "X-RFCXXXX-JSNAME" "=" DQUOTE *QSAFE-CHAR DQUOTE</code>
Description	This parameter is set on a X-RFCXXXX-JSPROP or X-RFCXXXX-PROP property. It assigns the JSCalendar property name to the value embedded in the iCalendar property value. The parameter value contains the verbatim name of the JSCalendar property. It MUST be quoted to preserve case.
Example(s)	<code>X-RFCXXXX-PROP;X-RFCXXXX-JSNAME="fooBar":baz</code>

13. JSCalendar base spec updates

- Make title and description optional in an alert. Used to hold SUMMARY and DESCRIPTION if present.

14. Security Considerations

The same security considerations as for [IETF RFC 8984](#) 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.

Bibliography

- [1] TZDB, IANA, "Time Zone Database", <https://www.iana.org/time-zones>.