

calext
Internet-Draft
Intended status: Informational
Expires: 29 September 2026

R. Kashyap
iCalDAV Project
28 March 2026

Machine-Readable Property Dependencies for iCalendar (RFC 5545)
draft-kashyap-calext-ical-property-deps-00

Abstract

RFC 5545 defines inter-property interaction rules for iCalendar components in prose scattered across multiple RFC sections. No published, standalone representation of these rules exists in a form suitable for machine consumption. This document specifies a formal dependency graph for VEVENT components with 27 properties and 20 edges spanning six edge types. It classifies properties into five merge safety categories that determine whether concurrent edits can be combined without user intervention. The specification is published as YAML and JSON artifacts for cross-language consumption.

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 29 September 2026.

Copyright Notice

Copyright (c) 2026 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Scope	3
1.2. Merge Unit	3
2. Terminology	4
3. Edge Type Definitions	4
4. Edge Strength	5
5. Property Dependencies	5
5.1. Cross-VEVENT Edges	7
5.2. RRULE Internal Constraints	7
6. Merge Safety Classification	7
6.1. Safe	7
6.2. Dependent	8
6.3. Scheduling	8
6.4. Immutable	8
6.5. Always-Update	8
6.6. Set-Valued Properties	9
7. Machine-Readable Artifact	9
7.1. Schema Fields	9
7.2. Versioning	10
8. Implementation Evidence	10
9. Security Considerations	11
10. IANA Considerations	11
11. References	11
11.1. Normative References	11
11.2. Informative References	12
Acknowledgements	12
Author's Address	12

1. Introduction

CalDAV [RFC4791] treats calendar resources as opaque blobs. Clients PUT entire iCalendar objects; servers use ETags for conflict detection. When two clients modify the same event, the only options are whole-event replacement or manual resolution. Property-level merge is not part of the protocol.

The barrier to property-level merge is not algorithmic. Three-way merge for structured data is well understood. The barrier is that [RFC5545] property dependencies are implicit: changing DTSTART can invalidate RRULE expansion, EXDATE membership, and VALARM trigger offsets, but this is expressed only in prose across separate RFC sections. Without a formal model of which properties depend on which, a merge engine cannot determine whether two concurrent edits are safe to combine.

This document extracts these implicit dependencies into a formal graph for VEVENT components and classifies each property by merge safety.

1.1. Scope

This specification covers VEVENT properties only. VTODO and VJOURNAL have different dependency structures (e.g., the Tasks draft removes the VTODO DURATION-to-DTSTART requirement). Extension to other component types is future work.

The graph includes all VEVENT properties with non-trivial inter-property dependencies, plus commonly-edited safe properties. [RFC5545] Section 3.6.1 defines additional VEVENT properties with no inter-property dependencies that are omitted for brevity: ATTACH, COMMENT, CONTACT, RELATED-TO, and RESOURCES. All are safe with set cardinality and union merge semantics.

Single-property cardinality constraints (e.g., DTSTART is REQUIRED) and value-level constraints (e.g., DTEND datetime MUST be greater than or equal to DTSTART datetime) are not modeled. This graph captures structural and type-level dependencies only.

1.2. Merge Unit

CalDAV resources can contain multiple VEVENTs sharing a UID: one master and zero or more exception VEVENTs (each identified by RECURRENCE-ID). This graph models properties within a single VEVENT, but three edges cross VEVENT boundaries. A merge engine consuming this graph MUST operate on the full UID group (all VEVENTs sharing a UID), not individual VEVENTs in isolation. Edges marked "cross_vevent" in the machine-readable artifact require this UID-group scope.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are used throughout this document:

Edge: A directed relationship between two properties in the dependency graph. Each edge has a type and a strength.

Merge safety: A classification of whether a property can be independently merged when two clients concurrently edit the same event.

UID group: The set of all VEVENTs sharing the same UID value within a calendar resource. Consists of one master VEVENT and zero or more exception VEVENTs identified by RECURRENCE-ID.

Significant change: A change to a property classified as "dependent" or "scheduling" in the merge safety classification. Significant changes require SEQUENCE to be incremented per [RFC5546] Section 2.1.4.

3. Edge Type Definitions

The graph uses six edge types with the following semantics:

depends_on: The source property's interpretation requires the target property's value. Changing the target makes the source's cached result stale, but the source's stored value remains syntactically valid. Example: RRULE depends_on DTSTART ([RFC5545] Section 3.8.5.3) because recurrence expansion uses DTSTART's day-of-week, month, etc. for BY* rules.

derived_from: The source property's value was computed from the target at creation time. The target can change afterward without requiring the source to update. Example: RECURRENCE-ID derived_from DTSTART.

requires: The source property requires the target to be present. The source without the target is a protocol violation. Example: ATTENDEE requires ORGANIZER ([RFC5545] Section 3.8.4.1).

mutually_exclusive_with: The source and target MUST NOT both be

present on the same VEVENT. This relationship is symmetric.
Example: DTEND mutually_exclusive_with DURATION ([RFC5545]
Section 3.6.1).

type_consistency: The source and target must use the same value type
(DATE or DATE-TIME). This relationship is symmetric. Example:
EXDATE type_consistency DTSTART ([RFC5545] Section 3.8.5.1).

computes_with: The source and target are used together in
computation. Neither needs updating when the other changes.
Example: DTSTART computes_with DURATION to produce the implied end
time.

4. Edge Strength

Each edge carries a strength indicating the RFC conformance level of
the constraint it represents:

must: An RFC MUST or MUST NOT requirement. Violation is a protocol
error.

should: An RFC SHOULD or SHOULD NOT recommendation. Violation is
suboptimal but conformant.

advisory: Not a conformance requirement. Useful for generating
warnings (e.g., an EXDATE for a date not in the recurrence set is
a no-op, not a violation).

informational: Annotation only. Documents a computational
relationship with no conformance or warning implication. Used for
computes_with and derived_from edges.

5. Property Dependencies

The following table lists the 20 inter-property edges. All edges
derive from [RFC5545]. Additional RFCs ([RFC7986], [RFC5546],
[RFC6047], [RFC6638]) contribute property definitions and scheduling
semantics used in the merge safety classification (Section 6), but no
edges in this table originate from those documents. Symmetric edges
(type_consistency, mutually_exclusive_with) are listed on both
participating properties, yielding 20 entries for 16 unique
relationships.

Source	Target	Type	Str.	Ref
RRULE	DTSTART	depends_on	must	S3.8.5.3
EXDATE	DTSTART	type_cons.	must	S3.8.5.1
EXDATE	RRULE	depends_on	adv.	S3.8.5.1
RDATE	DTSTART	type_cons.	must	S3.8.5.2
VALARM	DTSTART	depends_on	must	S3.8.6.3
VALARM	DTEND	depends_on	must	S3.8.6.3
VALARM	DURATION	depends_on	must	S3.8.6.3
REC-ID	DTSTART	derived_from	info.	S3.8.4.4
REC-ID	RRULE	depends_on	must	S3.8.4.4
REC-ID	RDATE	depends_on	must	S3.8.4.4
ATTENDEE	ORGANIZER	requires	must	S3.8.4.1
DTEND	DURATION	mut_excl.	must	S3.6.1
DTSTART	DURATION	computes_w.	info.	S3.8.2.5
DURATION	DTSTART	depends_on	must	S3.8.2.5
DTSTART	DTEND	type_cons.	must	S3.6.1
DTSTART	EXDATE	type_cons.	must	S3.8.5.1
DTSTART	RDATE	type_cons.	must	S3.8.5.2
DTSTART	UNTIL	type_cons.	must	S3.3.10
DTEND	DTSTART	type_cons.	must	S3.6.1
DURATION	DTEND	mut_excl.	must	S3.6.1

Table 1: Inter-Property Dependency Edges for VEVENT

Abbreviations: REC-ID = RECURRENCE-ID; type_cons. = type_consistency; mut_excl. = mutually_exclusive_with; computes_w. = computes_with; adv. = advisory; info. = informational. All references are to [RFC5545] sections.

VALARM is a sub-component nested inside VEVENT, not a VEVENT property. Its edges are modeled at the VEVENT level for merge safety purposes. A complete VALARM merge requires set-level comparison of sub-components, not scalar property diffing. The VALARM depends_on edges are conditional on each alarm's TRIGGER RELATED parameter: RELATED=START (the default) depends on DTSTART; RELATED=END depends on DTEND or, when DTEND is absent, on DURATION. A merge engine that unconditionally flags all VALARMS when DTEND changes will produce false positive conflicts for RELATED=START alarms.

The DURATION depends_on DTSTART edge ([RFC5545] Section 3.8.2.5) captures a non-obvious format constraint: if DTSTART uses the DATE value type (all-day event), DURATION MUST use dur-day or dur-week format (e.g., P1D, P2W), not time granularity (e.g., PT1H). Changing DTSTART from DATE-TIME to DATE requires validating the DURATION format.

5.1. Cross-VEVENT Edges

RECURRENCE-ID lives on an exception VEVENT; its targets (RRULE, RDATE, DTSTART) live on the master VEVENT. These edges are marked "cross_vevent: true" in the machine-readable artifact. A merge engine MUST resolve these edges across the full set of VEVENTs sharing a UID.

5.2. RRULE Internal Constraints

COUNT and UNTIL are mutually exclusive within an RRULE value ([RFC5545] Section 3.3.10). This is captured as an internal constraint rather than an inter-property edge, because both are parameters of the RRULE value, not separate VEVENT properties.

6. Merge Safety Classification

Given two concurrent edits to the same event (local and remote), can a property be independently merged? This section classifies all 27 VEVENT properties into five categories.

6.1. Safe

Properties: SUMMARY, DESCRIPTION, LOCATION, URL, GEO, PRIORITY, CATEGORIES, COLOR ([RFC7986]), CLASS, TRANSP, STATUS.

No cross-property dependencies. Independent merge is always valid. If local changes SUMMARY and remote changes LOCATION, both edits can be kept without conflict.

STATUS is structurally safe but semantically hazardous. Auto-merging STATUS:CANCELLED with a concurrent SUMMARY edit produces a cancelled event with a new title. A merge engine SHOULD surface STATUS:CANCELLED as a manual-resolution candidate.

6.2. Dependent

Properties: DTSTART, DTEND, DURATION, RRULE, EXDATE, RDATE, VALARM.

Can be merged if graph validation passes. For example, if local changes DTSTART and remote adds an EXDATE, the merge engine MUST verify that the EXDATE value type still matches the new DTSTART type.

6.3. Scheduling

Properties: ATTENDEE, ORGANIZER, REQUEST-STATUS.

When CalDAV Scheduling [RFC6638] is active, modifying these properties triggers scheduling messages per [RFC6047]. They MUST NOT be auto-merged.

On servers without [RFC6638] (e.g., Radicale, Baikal), ATTENDEE is dependent (requires ORGANIZER) and ORGANIZER is also dependent (removing ORGANIZER while ATTENDEEs are present violates the "requires" edge regardless of scheduling support). REQUEST-STATUS falls back to safe. The machine-readable artifact captures this via a "scheduling_fallback" field.

6.4. Immutable

Properties: UID, CREATED, RECURRENCE-ID.

Set once within a VEVENT instance. MUST NOT be changed after initial creation. UID identifies the event ([RFC5545] Section 3.8.4.7). CREATED records creation time ([RFC5545] Section 3.8.7.1). RECURRENCE-ID identifies which occurrence of a recurrence series was modified ([RFC5545] Section 3.8.4.4).

6.5. Always-Update

Properties: SEQUENCE, DTSTAMP, LAST-MODIFIED.

Automatically set on any edit. DTSTAMP and LAST-MODIFIED are set to the current UTC time.

SEQUENCE has a non-trivial merge rule. If only one side changed significant properties, the merge engine SHOULD preserve that side's SEQUENCE. If both sides changed significant properties, the merge engine SHOULD use $\max(\text{local}, \text{remote}) + 1$. If neither changed significant properties, the merge engine SHOULD use $\max(\text{local}, \text{remote})$. A blind $\max + 1$ on every merge causes the escalation loop documented in existing CalDAV client implementations.

6.6. Set-Valued Properties

Properties with cardinality "set" (EXDATE, RDATE, CATEGORIES, ATTENDEE, VALARM) hold multiple values. The merge safety classification determines whether a property can be independently merged with edits to other properties but does not specify how to merge concurrent edits to the same multi-valued property.

The machine-readable artifact includes a "merge_op" field for set properties: "union" (EXDATE, RDATE, CATEGORIES) for additive merge, or "conflict" (ATTENDEE, VALARM) for set-level comparison requiring manual resolution.

EXDATE union requires checking for corresponding exception VEVENTs on the same date before applying. Adding an EXDATE for a date that has an exception VEVENT produces conflicting signals (the occurrence is both excluded and replaced).

7. Machine-Readable Artifact

The dependency graph is published as structurally equivalent YAML and JSON files at <https://github.com/iCalDAV/ical-dependency-graph>. The YAML includes inline commentary; the JSON uses fields prefixed with underscore ("_note", "_schema") for annotations. Underscore-prefixed fields are non-normative.

7.1. Schema Fields

Each property object contains:

cardinality: "scalar" (single value) or "set" (multiple values).

merge_safety: One of: "safe", "dependent", "scheduling", "immutable", "always_update".

Edge lists: Zero or more of: "depends_on", "derived_from", "requires", "mutually_exclusive_with", "type_consistency", "computes_with". Each entry is an object with "target" (property name) and "strength" (must/should/advisory/informational).

`merge_op:` (Set properties only.) "union" or "conflict".

`scheduling_fallback:` (Scheduling properties only.) The merge safety category to use on servers without [RFC6638].

`merge_rule:` (SEQUENCE only.) Three conditions: "one_side_changed" (PRESERVE), "both_sides_changed" (MAX_PLUS_ONE), "neither_changed" (MAX).

`cross_vevent:` (Optional, boolean.) True when the edge crosses VEVENT boundaries within a UID group.

7.2. Versioning

The "version" field uses semantic versioning:

- * Patch (1.0.x): Corrects an incorrect edge. No new properties or edge types.
- * Minor (1.x.0): Adds new properties, edge types, or schema fields. Existing edges unchanged.
- * Major (x.0.0): Changes existing edge semantics or restructures the schema.

8. Implementation Evidence

25 bugs were identified across 17 distinct open-source projects spanning 8 programming languages (JavaScript, Python, C, Java, PHP, Kotlin, C#, Vala) where the root cause is a violated property dependency. The bugs fall into three categories:

- * Expansion/interpretation (13 bugs): The dependency rule was not correctly implemented in RRULE expansion, EXDATE filtering, or property interpretation.
- * Validation/serialization (10 bugs): Invalid property combinations were accepted or produced.
- * Merge/sync (2 bugs): Concurrent edits or sync operations produced invalid state due to missing property-level awareness.

The EXDATE-type-must-match-DTSTART rule ([RFC5545] Section 3.8.5.1) alone accounts for 7 separate bugs across independent implementations. The RRULE-DTSTART interaction has bugs dating back to 2013. 13 of 25 bugs involve these same two rules across independent implementations.

Additionally, 7 major calendar libraries across 6 languages have documented cross-property validation gaps: ical4j (Java), ical.js (JavaScript), libical (C), Python icalendar (Python), sabre/vobject (PHP), rrule.js (JavaScript), and ical4android (Kotlin).

The full bug catalog with links to individual issues is available in the companion preprint [ICAL-DEP-GRAPH] and at the repository listed in Section 7.

9. Security Considerations

This document formalizes existing constraints from [RFC5545] and related specifications. It does not introduce new protocol elements or modify existing ones.

Implementations that consume the dependency graph for automated merge operations SHOULD validate the graph artifact's integrity before use (e.g., by verifying a checksum or fetching from a trusted source). A modified graph could cause a merge engine to incorrectly classify unsafe merges as safe, producing invalid iCalendar objects.

The scheduling properties (ATTENDEE, ORGANIZER, REQUEST-STATUS) have side effects beyond the local event when CalDAV Scheduling [RFC6638] is active. A merge engine MUST NOT auto-merge scheduling properties without the user's explicit consent, as this could result in unintended scheduling messages being sent to other participants.

10. IANA Considerations

This document has no IANA actions.

11. References

11.1. Normative References

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

- [RFC6047] Melnikov, A., Ed., "iCalendar Message-Based Interoperability Protocol (iMIP)", RFC 6047, DOI 10.17487/RFC6047, December 2010, <<https://www.rfc-editor.org/info/rfc6047>>.
- [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>>.
- [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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

11.2. Informative References

- [ICAL-DEP-GRAPH] Kashyap, R., "Formalizing Implicit Property Dependencies in RFC 5545: Toward Safe Property-Level Calendar Synchronization", DOI 10.5281/zenodo.19299690, March 2026, <<https://doi.org/10.5281/zenodo.19299690>>.
- [RFC4791] Daboo, C., Desruisseaux, B., and L. Dusseault, "Calendaring Extensions to WebDAV (CalDAV)", RFC 4791, DOI 10.17487/RFC4791, March 2007, <<https://www.rfc-editor.org/info/rfc4791>>.

Acknowledgements

The dependency rules formalized in this document were validated against 295+ tests across three codebases and six CalDAV servers (iCloud, Nextcloud, Radicale, Baikal, Stalwart, Zoho). The bug catalog was compiled from public GitHub issues across 17 open-source calendar projects.

Author's Address

Ravi Kashyap
iCalDAV Project
URI: <https://github.com/iCalDAV/ical-dependency-graph>