

Lacuna

A provenance record specification

VERSION	0.7
DATE	13 October 2026 (v0.6, 14 September 2026 · v0.5, 4 September 2026 · v0.4, 1 September 2026 · v0.3, 28 August 2026 · v0.2, 24 August 2026 · v0.1, 22 August 2026)
AUTHOR	Christopher Banks, Objects of Affection Collection
STATUS	Published to establish prior art. Not a standard until implemented by someone other than its author.
DOI	10.5281/zenodo.22063353 (all versions) · 10.5281/zenodo.22063354 (v0.1) · 10.5281/zenodo.22084159 (v0.2) · 10.5281/zenodo.22150280 (v0.3) · 10.5281/zenodo.22234788 (v0.4) · 10.5281/zenodo.22303877 (v0.5) · 10.5281/zenodo.22751179 (v0.6)
CITE AS	Banks, C. (2026). <i>Lacuna: A Specification for Provenance Records with Declared Absence</i> . Zenodo.
LICENSE	CC BY 4.0
CHANGES IN 0.7	Section 6 gains 6.0 and section 5 gains 5.7 , and BOTH ARE NORMATIVE CHANGES . A value that no source records and no party withholds is an absence and MUST be typed under section 5; section 6 carries only a value a NAMED party declined to disclose. <code>not_applicable</code> MUST rest on a positive ground and MUST NOT be declared for a fact whose former existence is uncertain; an undecidable case is <code>not_yet_obtained</code> naming the party who could settle it. Section 10 gains requirement 15 and requirement 17's no-variation range moves to 1 through 15. Non-normative in the same version: section 5.1 records that <code>never_required</code> has one instance across a complete sweep, correcting a figure of zero that rested on a glob which did not reach a record filed under a second naming convention, and

that the CIDOC CRM SIG heard issue 723 on 8 October 2026 and deferred it without answering either the vocabulary question or the CRM mapping.

CHANGES IN 0.6 Section 5 gains **5.6** and **THIS IS A NORMATIVE CHANGE**. A gap declared `not_yet_obtained` **MUST** carry `absence_review_by`, a date or NULL and never a condition, on the same rule section 6.1 applies to a withholding; section 10 gains **requirement 14**; **requirement 16's no-variation range moves to 1 through 14 in the same edit, and the SHOULD and the range renumber to 15 and 16**. A second, optional field, `absence_enquired_on`, records that the named holder was asked and when, so that a holder who does not answer is distinguishable from a holder who was never asked. The change closes a defect found by the author on 2026-09-14 while comparing this specification against the Getty Provenance Index's documentation act and the ICOM-CIDOC Statement of Principles of Museum Documentation (2012): section 5.5 required work to have an addressee and never required it to have a date, so a gap could be correctly typed as work and stay open forever with nothing to show that it had. The Principles' own requirement 14 already asks for a documentation upgrade "by an agreed deadline" at the level of a collection; this version asks it at the level of the gap. **Non-normative in the same version:** section 11 credits the Principles where prior art is discussed, and section 13 declares two further gaps rather than closing them. **No requirement changes as a result of either.**

CHANGES IN 0.5 Section 6 is amended and **THIS IS A NORMATIVE CHANGE**. `field_disclosure` gains an optional `withheld_until` date, and section 10 gains **requirement 13**, which requires a known end date on a withholding to be recorded where one exists; **requirement 14's no-variation range moves to 1 through 13 in the same edit, and the SHOULD and the range renumber to 14 and 15**. The change corrects a sentence in section 6 that was true of the case it was written from and false in general: withholding was described as something *"nobody can ever close, by design."* It answers a case put to the author on 2026-09-04 by an archivist at a major auction house, **who is not named here because he has not been asked**. The pattern is adopted from ISAD(G) element 4.1 and is not claimed as novel. **Non-normative in the same version:** section 11 credits ISAD(G)

where prior art is discussed, section 13 declares two further gaps rather than closing them, and three em-dashes inherited from earlier versions are replaced under the house style rule. **No requirement changes as a result of any of those.**

CHANGES IN 0.4 Section 5.4 is added: an absence entry describes the record rather than the world, and an external source that fills a declared gap supersedes the entry rather than conflicting with it. **This is a normative change**, adding two MUST NOT clauses. It answers an open world objection put to the author on 28 August 2026 by a standards practitioner in cultural heritage informatics, who is not named here because he has not been asked. Section 10 gains **requirement 12**, which requires a conforming record to declare the specification it conforms to, its version and its persistent identifier; requirement 14's no-variation range moves to 1 through 12 in the same edit. Section 14, correction semantics, and section 5.5 are carried forward from work completed after v0.3 was deposited.

CHANGES IN 0.3 Section 11 is expanded. CRMinf, Huggett (2020) and the archival science literature on silence are added alongside Linked Art, CIDOC CRM and Art Tracks, each quoted where it states its own position, and the pattern across them is named. **No normative change.**

CHANGES IN 0.2 Section 9 now requires a conforming implementation to publish the method by which its hashes are computed, and points at the normalization rules and reference implementation published as [doi:10.5281/zenodo.22083313](https://doi.org/10.5281/zenodo.22083313). Section 13 records that gap as resolved rather than removing it. No other normative change.

A **lacuna** is the term for a declared missing portion of a manuscript or text. It is precisely, and only, what this specification contributes: an absence that is stated rather than left blank.

It sits deliberately beside **Colophon**, the tool it is implemented in. A colophon is the printer's mark that states who made a thing and how; a lacuna is what the text declares to be missing. **One names what is known about the making, the other names what is absent.** Both are drawn from the vocabulary of the physical book.

Lacuna is an unregistered mark in use. Nothing here asserts a registration.

This document specifies a structure for recording the custody of an object over time.

Its distinguishing feature is that **absence is a first class, typed, reasoned entry rather than a missing value.** Where a conventional record leaves a field blank, a conforming record states which kind of absence this is and why, as a row with the same standing as any other. A record with declared gaps is more trustworthy than one that appears complete, because the second kind cannot be distinguished from one that was never checked.

A second feature follows from a real maker's refusal during the first week of field testing: **a withheld fact is not an absent one.**

Withholding is recorded separately from absence, and a withheld field must name who withheld it.

This specification does not authenticate anything. See section 1.2.

Abstract

This document specifies a structure for recording the custody of an object over time.

Its distinguishing feature is that **absence is a first class, typed, reasoned entry rather than a missing value**. Where a conventional record leaves a field blank, a conforming record states which kind of absence this is and why, as a row with the same standing as any other. A record with declared gaps is more trustworthy than one that appears complete, because the second kind cannot be distinguished from one that was never checked.

A second feature follows from a real maker's refusal during the first week of field testing: **a withheld fact is not an absent one**. Withholding is recorded separately from absence, and a withheld field must name who withheld it.

This specification does not authenticate anything. See section 1.2.

Section 01

Scope

1.1 What a record is

A provenance record is **a sequence of dated assertions about who held an object, when, and on what basis**. It is about custody across time. Each assertion carries its own source.

1.2 What a record is not, stated first because it is the most common error

A provenance record makes no claim that an object is genuine. Authentication is a judgment about an object at a moment. Provenance is a record of custody over time. They are different acts, they fail differently, and conflating them is the defect this

specification exists to avoid.

A conforming implementation **MUST NOT** describe its output as proof of authenticity, and **MUST NOT** present an integrity guarantee (section 9) as one.

A certificate of authenticity issued by a third party **MAY** be recorded as an input to a record, qualified as attested (section 8). It is evidence that someone made a claim. It is not the claim of this record.

1.3 Domain independence

The structure is fixed. **The vocabulary is not.**

A conforming record uses the terms the maker actually uses. A label that produces "waves" records waves, not editions. A grower records a harvest. **Translating a maker's own vocabulary into a generic one is a small act of erasure and a conforming implementation SHOULD avoid it.**

What may never vary: the required fields, what counts as a source, what an absence looks like, and what a subject may assert about their own object. Those are fixed before any party adopts the specification, including its author.

Section 02

Conformance language

MUST, MUST NOT, SHOULD, MAY carry their ordinary specification meanings. A conforming implementation satisfies every **MUST**.

Section 03

Parties

A **party** is a person, organization, group, or an explicitly unidentified actor.

```
parties(id, kind, display_name, created_at) kind ∈ { person, organization, group, unidentified } display_name is non-empty
```

unidentified is a party, not a null. "Sold to a private collector whose name was not recorded" is a real event with a real actor, and modeling it as a missing party loses the event. A record **MUST** represent an unnamed holder as an `unidentified` party rather than omitting the custody event.

Relations between parties (succession, control, corporate identity over time) are recorded separately, each with its own effective date and source, because **institutions rename, merge, and dissolve, and a record spanning decades must survive that.**

Section 04

Custody events

The spine of a record.

```
custody_events(id, object_id, kind, began_on, ended_on, quantity, unit, parent_event_id, method, absence_state, absence_reason, absence_holder, absence_asserted_at, absence_review_by, absence_enquired_on, created_at) kind ∈ { origin, held, transfer, gap }
```

absence_holder and **absence_asserted_at** are populated only on `gap` rows, and section 5.5 states when each is mandatory. **absence_asserted_at is not created_at**. `created_at` records when the row appeared and never changes. `absence_asserted_at` records **when the absence was last affirmed**, and moves when a custodian re-checks an open gap and finds it still open. **A record that cannot**

distinguish those two is unable to say whether a ten-year-old gap was ever looked at twice. `absence_review_by` and `absence_enquired_on` are populated only on `not_yet_obtained` rows; section 5.6 states when each is mandatory.

KIND	MEANING
<code>origin</code>	the object came into existence
<code>held</code>	a party had custody across a span
<code>transfer</code>	custody moved
<code>gap</code>	a declared absence in the chain. See section 5.

origin exists to protect gap. Without an explicit origin, "nothing came before this" would have to be expressed as a gap, and the record would carry a permanent false absence at the start of every chain. A conforming implementation **MUST** enforce that an `origin` event has no parent.

`quantity` and `unit` are required because **not every object is indivisible**. A harvest divides. A bolt of cloth is cut. Quantity **MUST** be positive; a zero quantity is not a custody event.

Section 05

Absence is a typed, reasoned entry

The central requirement of this specification.

A gap in a chain of custody is **not** a null, a blank field, or a flag. It is a row, of the same kind as every other row, and it **MUST** state which kind of absence it is and why.

5.1 The absence vocabulary

`absence_state` ∈ { `unknowable`, `not_applicable`, `not_yet_obtained`, `never_required` }

STATE	MEANING	WHAT A READER SHOULD DO
<code>unknowable</code>	the fact cannot be recovered by anyone	stop looking
<code>not_applicable</code>	the question does not arise for this object	stop asking
<code>not_yet_obtained</code>	the fact exists and has not been collected	this is work, and it can be closed
<code>never_required</code>	the question is not proper to ask here	do not ask ⚠️ (<i>no instances; see the note below</i>)

The completeness test for this vocabulary is that each state licenses a different action. A vocabulary whose members produce identical behavior is one state wearing four names.

⚠️ **`never_required` IS USED ONCE IN THIS AUTHOR'S RECORDS, AND THE QUESTION IS OPEN (v0.7).**

Measured 2026-09-22 against the naming convention this specification prescribes, `never_required` returned zero. Re-measured against a complete sweep of the directory, `never_required` returns one. The difference is a single record filed under a second naming convention, which the prescribed glob did not reach. **One counting defect produced two wrong figures, a conformance count and an entry tally, and only the conformance count was repaired at the time.**

⚠️ **No tally of the other states is published here, deliberately.** Any such figure is a dated measurement over a corpus that is still being written, it was measured three different ways while preparing this version, and a count published in a citable document outlives the week in which it was true. **The claim below needs no denominator to be checked: it is about a member used once, and the entry it was used on is named.**

The one instance is the strongest evidence in this section, and it argues against the member rather than for it. It is an entry recording what an automated work has printed: the work produces a unique object for each visitor, the object leaves the building, and nothing records what was made or for whom. **The author's own paper describes that same entry, in writing, as undecided between an absence nobody can recover and a question no documentation regime ever asked.** The record nonetheless commits it to `never_required`.

★ **So the honest claim is not that the member is unused. It is that the member was used once, in the one entry the author could not classify.** A state reached only where the recorder was uncertain is not evidence that the state is needed; it is evidence that it absorbs uncertainty, which is the failure mode this specification exists to prevent.

It is retained in v0.7 and it is not defended. The state passes the completeness test on reasoning (it licenses "do not ask", which no other member licenses) and fails it on evidence. The question of whether the vocabulary should have three members and a note was put to the CIDOC CRM SIG on 2026-09-22 as part of issue 723.

The SIG heard issue 723 on 2026-10-08 and did not settle it. It was deferred rather than closed, the group's view being that the pattern is neither of the two existing constructs offered to it and may require a construct of its own, and the group asked for further cases. **No answer was given on whether this vocabulary should have three members or four, and none was given on the CRM mapping.**

So the member is retained, undefended, and the question stays open in the document rather than being closed by the author while a standards body is still considering it. It is recorded here so that a reader meets the weakness in the

specification rather than discovering it in the data.

A count is a dated measurement. Re-measure before citing; the figures above are of 2026-09-22.

5.2 The load bearing constraints

A conforming implementation **MUST** enforce all of the following:

```
(kind = 'gap') = (absence_state IS NOT NULL) -- gaps and only gaps carry a
state kind <> 'gap' OR length(btrim(absence_reason)) > 0 -- a gap MUST say
why absence_state IS NULL OR absence_state IN (the four) -- closed
vocabulary absence_state <> 'not_yet_obtained' OR
length(btrim(absence_holder)) > 0 -- see 5.5: work must have an addressee
absence_state IS NULL OR absence_asserted_at IS NOT NULL -- 5.5: an absence
is a dated claim absence_state <> 'not_yet_obtained' OR absence_review_by
IS NOT NULL -- see 5.6: work has a date or it is not work absence_review_by
IS NULL OR absence_state = 'not_yet_obtained' -- 5.6: only work is reviewed
absence_enquired_on IS NULL OR absence_state = 'not_yet_obtained' -- 5.6:
only a named holder can have been asked
```

The biconditional is deliberate and is the whole mechanism. If `absence_state` were merely nullable, it would decay into a binary "known / unknown" flag within a month, because **NULL is where unresolved things go**. Requiring that a gap declare its kind, and that nothing else may declare one, is what prevents that decay.

And a gap MUST carry prose. A typed absence with no reason is a category with no content.

5.3 Why this matters

Prior systems represent gaps by omission or by relative sequencing. Omission is indistinguishable from never having looked. **A record whose absences are declared can be audited; one whose absences are silent cannot.**

5.4 An absence entry describes the record, not the world

An absence entry is a statement about this record and the party keeping it. It is not a claim about what happened.

`not_yet_obtained` asserts that this record does not hold the value and that nobody has yet been to look. It does not assert that no transfer occurred. `unknowable` asserts that the fact cannot be recovered, which is a claim about recoverability rather than about the event. **In every case the subject of the sentence is the record, not the object's history.**

This matters most where records meet. A second record that supplies a value this one declares absent does not contradict this record; it supersedes the entry. A conforming implementation **MUST NOT** treat a declared absence as evidence that nothing occurred, and **MUST NOT** treat an external source that fills a declared gap as a conflict to be resolved.

Why this is stated rather than assumed. In an open world model, anything not stated is unknown rather than false, and a positively asserted gap would otherwise appear to conflict with knowledge held elsewhere that fills it. **It does not conflict, because the two sentences have different subjects.** One says the world contains no such event. The other says this record does not hold it. **Only the second is ever asserted here.**

5.5 `not_yet_obtained` MUST name who can close it, and MUST carry a date

`not_yet_obtained` is the cheapest of the four states to assert, and left unconstrained it becomes the default for every field nobody has worked. A record in which most gaps carry it is not a record with many open questions. It is a record with a blank field under a longer name.

This is the defect 5.1 already forbids one level up. There, four states that licensed the same action would be one state wearing four names. Here, one state applied to everything collapses the distinction from the other side, and the result is the same: **an entry that carries no information because it never varies.**

The vocabulary distinguishes the four states by what a reader should *do*.

`not_yet_obtained` says **this is work, and it can be closed**. Work that can be closed has an addressee. So:

A gap whose `absence_state` is `not_yet_obtained` MUST name, in `absence_holder` , the party believed to hold the fact: a person, an institution, an archive, a registry, or a named record series. It need not be reachable, and it need not be right. **It must be nameable.**

If no party can be named, the entry is mistyped and the correct state is `unknowable` . That is the operational difference between the two, and until now the specification asserted the difference without giving anyone a way to apply it.

`unknowable` means the search space is empty. `not_yet_obtained` means it is not, and here is where it starts.

And every absence entry MUST carry `absence_asserted_at` , the date the claim was made. An absence is a statement about a record at a moment, not a standing fact about the world (5.4). A `not_yet_obtained` written three years ago and never revisited is a different claim from one written this morning, and a record that cannot tell them apart is asserting something it has not checked.

A conforming implementation MUST NOT treat these as advisory. A gap that names no holder and a gap that carries no date are both non-conforming, and an implementation **MUST** reject them rather than store them and warn.

The diagnostic this makes possible, and the reason it is worth the friction. Because holders and dates are required, a record can now be measured against itself: **a record in which every gap is `not_yet_obtained` , every holder is the same party, or every `absence_asserted_at` is the date the record was created, has been generated rather than worked.** Each of those is machine-detectable and none of them requires judgment.

Where this clause came from, stated because it changes how much weight it can carry. It was not designed. **The exemplar record was already doing it.** The Court of Tenacity record, written 2026-08-25 before any of this was specified, closes with a table whose fourth column names, for every open gap, the party who can close it: *"OAC production records", "the Custodian, one line", "the Custodian, enquiry sent 2026-08-25", "the issuer"*. One entry carries the date the enquiry went out. **A practitioner writing carefully produced both fields without being asked, and the specification had not noticed.** This clause ratifies an observed practice rather than proposing a new one, which is the same provenance as section 6 and the same reason to trust it.

No conformance threshold is specified, deliberately. A rule such as *"no more than N% of fields may be not_yet_obtained"* would be satisfied by retyping honest gaps as `unknowable`, which converts a visible unworked record into a false claim that the facts are unrecoverable. **A cap would make the record worse while making the metric better.** The requirement is that each entry carry its holder and its date; what the distribution then reveals is for a reader to judge.

5.6 not_yet_obtained MUST carry a review date, and an elapsed date is visible as one

Until version 0.6, section 5.5 required work to have an addressee and never required it to have a date. A gap could be correctly typed as work, correctly name the party who holds the fact, and stay open for ten years with nothing in the record to show that the work had not been done. **That is the defect section 6.1 corrected for withholding one version ago, in mirror image:** a withholding with no end date read as correctly held forever, and nothing prompted a review. An open gap with no review date reads as correctly open forever, and nothing prompts one either.

A gap whose absence_state is not_yet_obtained MUST carry absence_review_by, the date by which the party keeping the record undertakes to have asked the holder or restated the entry.

`absence_review_by` is a **date or NULL, and never a condition**, on the rule section 6.1 already states and for the reason it gives there: "when the archive reopens", "when funds allow", "when the estate is settled" are conditions with no listener. **For `not_yet_obtained` NULL is not conforming.** Work with no date is not work; it is a blank field under a longer name, which is the thing 5.5 forbids from the other side. A condition may still be described in `absence_reason`, which is prose and always was.

A gap MAY also carry `absence_enquired_on`, the date the named holder was asked. It is optional because not every gap has been worked yet. It exists because a holder who was asked and did not answer, and a holder who was never asked, are different facts about the record, and a record that renders them identically is silent about its own effort. **An enquiry that receives no answer changes nothing else in the row.** The gap stays `not_yet_obtained`, the holder stays named, and the record now shows who was asked and when. Silence from a holder is a complete answer about the holder and no answer about the fact.

What happens when the date passes and nobody has acted. 🛑 A conforming implementation MUST NOT change a row because `absence_review_by` has elapsed. Section 5.4 requires an entry to be a dated claim by the party keeping the record, and section 6.1 applies the same rule to a withholding. **An implementation MAY surface that the date has elapsed. Only a person may restate the row**, and a restatement is a correction under section 14: the gap is closed by a supersession naming the source (14.5); or it is re-dated, with the reason the work was not done recorded in the correction; or it is retyped to `unknowable`, with the search that exhausted the space recorded as the reason. **An elapsed `absence_review_by` with no correction after it is therefore visible for exactly what it is: a commitment the record-keeper made and did not keep.** That is not a defect in the record. It is the record doing its job, on its own keeper.

The diagnostic this adds to the one in 5.5. A record in which every review date is the same date, or every review date has elapsed with no correction, has been generated or abandoned, and each of those is machine-detectable without judgment.

This pattern is adopted, not invented. The ICOM-CIDOC Statement of Principles of Museum Documentation (2012), requirement 14, already asks a museum whose collection is inadequately documented to implement a programme to upgrade the documentation "by an agreed deadline." That requirement operates at the level of a collection. **This section applies the same deadline at the level of the gap, where the work is actually done, and nothing more.** The two are consistent: a collection-level programme is the sum of its gap-level review dates.

5.7 not_applicable MUST rest on positive grounds, and uncertainty is not grounds unknowable and not_applicable both tell a reader to stop, and they stop for opposite reasons. One says a fact existed and cannot be recovered. The other says there was never a fact to recover. **A reader who acts on the wrong one abandons a recoverable history or hunts a fact that never was.**

Section 5.5 gave not_yet_obtained and unknowable an operational test: can a party be named. Until now there was no equivalent test for not_applicable, and in practice it is the harder pair.

The defect was named by a practitioner rather than found here. A paper conservator put it plainly: for conservation documentation it is often impossible to establish whether information was recorded and destroyed, or never recorded at all, and that distinction belongs to a historian rather than to the person examining the object. **The specification was asking a recorder to settle a question outside their competence, and offering no route when they could not.**

So: a gap whose absence_state is not_applicable MUST rest on a positive ground, stated in absence_reason : something about this object, or the class it belongs to, that makes the question not arise. *"Built by the family that lived in it, so there is no architect"* is a ground. *"No record of an architect was found"* is not; that is an absence of evidence and belongs in another state.

🚫 **Uncertainty between "never existed" and "existed and was lost" is NOT a ground for either state.** It means the search space is not exhausted, because a party could still establish which of the two it is. **The correct state is `not_yet_obtained`, naming that party:** the historian, the archive, the record series that could settle it. The recorder is not asked to answer a question their discipline cannot answer. **They are asked to name who could.**

This narrows `not_applicable` deliberately. It is the rarest state in the author's own records (three entries, measured 2026-09-22) and it should stay rare: it is the only state that tells a reader a question is improper rather than merely unanswered.

Section 06

Withholding is not absence

Discovered in field testing, first week, by the first maker asked to supply the fields for a record.

Asked for manufacturing details, the maker replied that he would not share his factory, and would give country of production. That is the correct answer from a small maker: a manufacturer is a genuine competitive advantage, and a provenance system that requires exposing a supply chain does not understand what it is selling.

No state in section 5.1 expresses it. The fact is known to the subject; the question plainly arises; and asking is legitimate. **Withholding is a fact about disclosure, not about knowledge**, and it belongs on a different axis.

```
field_disclosure(id, table_name, row_id, field_name, disclosure,
withheld_by, withheld_until, reason, asserted_at) disclosure ∈ { public,
withheld } (disclosure = 'public') = (withheld_by IS NULL) -- a withheld
field MUST name its withholder withheld_until IS NULL OR disclosure =
'withheld' -- only a withholding can have an end
```

A withheld field MUST name who withheld it. "Withheld" alone is the same silence the specification exists to eliminate. **"Withheld by the maker" is information:** a reader in ten years learns two true things, where the object was made and that the maker chose not to name the factory.

⚠ A KNOWN AND DELIBERATE WEAKNESS: the absence of a disclosure row means public. A field nobody has classified is therefore treated as disclosed. This is recorded here rather than hidden because **an undocumented default is an undeclared decision**, and a future version should consider requiring explicit classification for a named set of sensitive fields.

6.0 A value that no source records and no party withholds is an ABSENCE, not a withholding

Added in v0.7, from the author's own records, and it is a defect of this specification rather than of the records that met it.

The constraint above already forbids the error: `(disclosure = 'public') = (withheld_by IS NULL)` means a withheld field **MUST** name its withholder, so a withholding entry naming nobody was never conforming. **What the prose never said is where such a value goes instead**, and in the absence of that sentence, seven entries across the author's own records were filed here rather than in section 5, each one describing itself as "not withheld by any named party; not recorded in any source read." Seven of seven withholding entries written after 2026-08-30 were absences. **A constraint that forbids an error without routing it is a constraint recorders work around.**

The rule, normative:

- **Section 6 carries only a value that a NAMED party declined to disclose.** If no party can be named, the entry does not belong here.
- **A value that no source records and no party withholds is an absence and MUST be typed under section 5:** `not_yet_obtained` where a party could still supply it (for a sale price, the parties to the sale), or `unknowable` where the search space is exhausted.

- A conforming implementation **SHOULD** reject a `field_disclosure` row with `disclosure = 'withheld'` and no `withheld_by`, with an error naming section 5 as the destination.

Why the two look alike and are not. Both leave a reader without the value. They license opposite actions, which is the completeness test of section 5.1: a withholding says the fact exists, is held by a named party, and may open on a date; an absence says nobody has it, or nobody has looked, and names who could close it. **A price nobody recorded is not a secret. Filing it as one invents a withholder who does not exist**, and it is the more damaging direction, because it implies a party chose concealment where none did.

 **Records dated before v0.7 are not rewritten** (§14.1). The seven entries stand as written, against the version they were written to, and this section is how the next record avoids them.

6.1 A withholding **MAY** end, and where it does the date is recorded

Until version 0.5 this section asserted that of a withholding "nobody can ever close it, by design." That was true of the single case it was written from and false in general, and it is corrected here rather than quietly dropped.

The case that corrected it: an auction house records that client information not printed in its catalogues is held under **a hundred years of confidentiality**, while anything printed in a catalogue at the time of sale is public. **The buyer of a lot sold in 1985 is therefore not unknown and not lost. The fact is held, and it opens in 2085.**

A permanent commercial confidence and a century-dated legal seal are different facts, and a record that renders them identically is the same silence this section exists to eliminate. That is the defect section 5.5 forbids one axis over: a value that never varies carries no information.

`withheld_until` is **a date or NULL**, and never a condition.

WITHHELD_UNTIL	MEANING	WHAT A READER SHOULD DO
NULL	no known end	stop. Nothing will change.
a date	it opens then	wait, and here is until when

Why a date and not a condition. "Withheld until the owner dies", "until proceedings conclude", "until circumstances change" are conditions with no listener. **They read as correctly held forever and nothing ever prompts a review.** Requiring a date or NULL forces the recorder to choose between a commitment and an honest permanent withholding. **A condition may still be described in reason , which is prose and always was,** exactly as section 5 pairs a typed state with mandatory prose.

 **A CONFORMING IMPLEMENTATION MUST NOT CHANGE A ROW BECAUSE A DATE HAS PASSED.** Section 5.4 requires an entry to be a dated claim by the party keeping the record. **An implementation MAY surface that a withheld_until date has elapsed. Only a person may restate the row.**

This pattern is adopted, not invented. ISAD(G) element 4.1, *Conditions Governing Access*, already requires an archival description to state the period of closure, the date the material opens, and the authority that imposed the condition; DACS 4.1 makes the element required. **This section previously held two of those three.** The contribution here is applying the archival closure date to object custody records alongside typed absence, and nothing more.

Section 07

Consideration, components, and contested claims

Consideration (what was exchanged) is recorded separately from the transfer event, because a transfer may have no consideration, and a price is a different kind of fact from a custody change.

Components record that an object has parts with their own histories. Each component **SHOULD** record the name as it appears on the artifact rather than a canonical title, so that **a disagreement between an object, its documentation, and its certificate is detectable by query rather than by memory.**

Contested claims record that two sources disagree, as a first class entry. **A contradiction may be preserved.** A record that flattens two disagreeing sources into one destroys the fact that they disagree, which is often the finding.

Section 08

Every field carries its source

```
field_sources(id, table_name, row_id, field_name, source_ref, qualifier,
asserted_at) source_ref is non-empty qualifier ∈ open set, e.g. { attested,
inferred, reported }
```

A conforming record **MUST** be able to answer, for any field, **where that value came from.**

`qualifier` distinguishes a fact attested by a document from one inferred by the recorder from one reported by an interested party. **These are different epistemic objects and a record that renders them identically is lying by formatting.**

 **Corroboration is not sourcing.** Two sources agreeing is not a source. A generator that fills five fields consistently produces exactly the pattern that fabrication produces, and beats an honest record with four declared gaps on every surface measure. **This is why declared absence is the center of this specification and internal consistency is not a conformance criterion.**

Section 09

Integrity, and precisely what it does not buy

A record's serialized form is hashed (SHA-256) and the hash entered in an append only chain in which each entry links to the previous. The chain root is periodically timestamped to a public blockchain. **Hashes only. No record content, no token, no coin.**

What this proves: the record has not been altered since it was issued.

What it does not prove: that the object is genuine, that the assertions in the record are true, or that the person who made them was entitled to.

A conforming implementation **MUST** state both halves wherever it states either. **An integrity guarantee presented without its limits is an authentication claim in disguise**, which section 1.2 forbids.

The serialization and normalization rules required to **reproduce** these hashes are specified in the OAC Provenance Verification Kit, **doi:10.5281/zenodo.22083313**, which carries a reference implementation in the public domain and test vectors. A conforming implementation **MUST** publish the method by which its hashes are computed. **A hash whose method is unstated is not reproducible by a third party, and therefore proves nothing to one.**

Section 10

Conformance

A conforming record:

1. **MUST NOT** claim to authenticate.
2. **MUST** represent every gap as a typed, reasoned entry (section 5).
3. **MUST** enforce the closed absence vocabulary and the gap biconditional.

4. **MUST** record withholding separately from absence, naming the withholder (section 6).
5. **MUST** be able to state the source of any field (section 8).
6. **MUST** represent an unnamed holder as an unidentified party, not as an omission.
7. **MUST** state the limits of its integrity guarantee wherever it states the guarantee.
8. **MUST** name, on any gap declared `not_yet_obtained` , the party who can close it (section 5.5).
9. **MUST** date every absence entry, so that an unrevisited claim is visible as one (section 5.5).
10. **MUST** append corrections and **MUST NOT** apply them in place (section 14.1).
11. **MUST** state, on any correction, what it leaves unchanged and the status of any issued copy (sections 14.3 and 14.4).
12. **MUST** state, in the record itself, the specification it conforms to, that specification's version, and its persistent identifier. **In the record, not in the interface that displays it.**
13. **MUST** record, on any withheld field whose end is known, the date it opens (section 6.1). **A withholding with a known end that is not recorded is indistinguishable from a permanent one.**
14. **MUST** record, on any gap declared `not_yet_obtained` , the date by which it will be reviewed, and **MUST NOT** alter the gap because that date has passed (section 5.6). **An open gap with no review date is indistinguishable from an abandoned one.**
15. **MUST** rest any gap declared `not_applicable` on a positive ground stated in the record, and **MUST NOT** declare it for a fact whose former existence is uncertain (section 5.7). **Uncertainty between "never existed" and "existed and was lost" is work, not a closed question.**
16. **SHOULD** use the subject's own vocabulary for domain terms.
17. **MUST NOT** vary requirements 1 through 15 for any individual adopter. **A standard is defined by what it refuses to change.**

The range in the last requirement is load-bearing and is the easiest thing in this document to break. It is written as a range so that a new requirement is covered by it automatically; **a requirement added above it without extending the range would be negotiable per adopter, which is the one thing this specification says it will not do.** Requirements 8 through 11 were added after the range first read *1 through 7*, and extending it was the same edit. **Requirement 12 was added 2026-08-30 and the range moved to *1 through 12* in the same edit, deliberately and in one change naming both halves.** Requirement 13 was added 2026-09-04 and the range moved to *1 through 13* in that same edit, which also renumbered the SHOULD and the range itself. Requirement 14 was added 2026-09-14 and the range moved to *1 through 14* in that same edit, with the SHOULD and the range renumbering to 15 and 16. Requirement 15 was added 2026-09-24 and the range moved to *1 through 15* in that same edit, with the SHOULD and the range renumbering to 16 and 17. Requirements 1 through 14 keep their numbers, deliberately: each new MUST was inserted after them rather than among them, so that nothing already cited elsewhere moved.

WHY REQUIREMENT 12 EXISTS, because it looks like housekeeping and is not

Section 12 of this document says it becomes a standard when somebody else implements it. Until now that event was UNOBSERVABLE. A third party could produce a perfectly conforming record and nothing in it would say so; it could not be found, counted, cited or checked. **A specification whose own definition of success cannot be detected by it has no way to know whether it has succeeded.**

It also settles attribution structurally rather than commercially. Without it, every adopter negotiates credit separately and the outcome depends on leverage. **Requirement 16 forbids exactly that for requirements 1 through 14**, so self-identification is now inside the thing this specification refuses to change.

THE LICENSE DOES NOT DO THIS WORK AND MUST NOT BE RELIED ON FOR IT. CC BY 4.0's conditions trigger on Share or Adapt of the licensed material, which is this text. **A record that CONFORMS to a specification is not the specification.** Read at source 2026-08-30. No legal opinion is offered and none is needed: the requirement makes the license question irrelevant.

In the RECORD, not the interface. A displayed line is a design decision a partner may revise. A field in the record survives redesign, screenshot, re-hosting and export.

 **IT BUYS DETECTION, NOT COMPLIANCE.** A party can declare conformance and breach every other requirement. This makes a claim of conformance visible and checkable; it does not make it true.

AND IT IS WHAT MAKES SECTION 5.4 OPERATIVE, WHICH IS THE TECHNICAL REASON RATHER THAN THE COMMERCIAL ONE

Section 5.4 states that an absence entry is a claim about the record rather than about the world. That is the answer to the open world objection: a second source that fills a declared gap supersedes the entry instead of contradicting it, because the two sentences have different subjects.

BUT A CONSUMING SYSTEM CAN ONLY APPLY THAT READING IF IT KNOWS WHICH SPECIFICATION THE RECORD FOLLOWS. Encountered without a declaration, a typed absence is just a field. It could equally be a closed world assertion that nothing happened, which is the reading section 5.4 exists to forbid. **Section 5.4 is unenforceable in practice unless the record says what it is.**

So requirement 12 is not an attribution device that happens to help commercially. It is the mechanism that makes the open world answer machine-actionable, and the two changes in this version belong together rather than merely arriving together.

IT IDENTIFIES THE SPECIFICATION, NOT ITS AUTHOR

A conforming record names the specification, its version and its identifier. It does not name Objects of Affection Collection, and it must never be amended to. This keeps the requirement inside ordinary standards practice, where a conforming document declares its schema, and **it means a record stays correctly identified if this specification ever outlives the institution that wrote it.**

Section 11

Relationship to existing work

This is not open ground and this section exists so that no reader is misled into thinking it is. The problem this document addresses has been described repeatedly, in more than one discipline, for longer than this document has existed.

Linked Art (version 1.0, released 19 February 2025) is the nearest prior art. It is mature, implemented by multiple institutions, and it models custody transfer. It represents uncertainty largely through relative sequencing and by omission.

CIDOC CRM provides the underlying event model much of this vocabulary descends from, and acknowledges the intrinsic incompleteness of information about the past.

CRMinf (version 1.2.1), the CIDOC CRM extension for argumentation, models belief: premises, inference making, and an I6 Belief Value carrying at minimum TRUE, FALSE and UNKNOWN. It states: *"it is recommended to use a richer vocabulary of belief values, at least including 'UNKNOWN'. CRMinf does not further prescribe such a vocabulary."* Its axis is confidence in a stated proposition. The axis here is the reason a field has no value. **These are compatible questions and they are not the same question.**

Art Tracks (Carnegie Museum of Art), CMOA Digital Provenance Standard v0.2, 14 October 2016 attempted structured provenance for museum collections and documented its own difficulties, including the tendency of an unresolved value to settle into a binary known/unknown flag. Section 5.2 exists because of that published experience. Its standard states: *"We have decided not to include a vocabulary or 'levels' of uncertainty, because we don't have a good system for determining how these levels would be specified."*

Huggett (2020), *Capturing the Silences in Digital Archaeological Knowledge*, Information 11(5), 278, doi:10.3390/info11050278 examines what he terms ignorance in archaeological data and proposes that the silences be addressed within a digital environment. It supplies a diagnosis this document depends on: that metadata is typically *"focused on the needs of discovery"*, and is therefore built to find things rather than to record what is not there. It proposes no schema.

Archival science holds an established literature on absence, including the term of art *archival silence*, the distinction between absence of evidence and evidence of absence, and formal gaps analyses conducted by institutions against their own holdings. **That literature theorizes absence. It does not, so far as the author has found, specify a machine checkable vocabulary for it.**

★ **Its descriptive standards, however, do specify one thing this document had missing until version 0.5. ISAD(G) element 4.1, *Conditions Governing Access*, requires a description to state the period of closure, the date the material opens, and the authority that imposed the restriction; DACS 4.1 makes the element required. Section 6.1 adopts the closure date from that practice. It is named here so that no reader takes it for an invention of this document.**

The museum sector's own principles do the same for the other axis. The ICOM-CIDOC *Statement of Principles of Museum Documentation* (version 6.2, 2012) requires, at requirement 9, that provenance be recorded "from its discovery or creation to the present" and that "the documentation should cite the source of this information"; at requirement 14, that an inadequately documented collection be upgraded "by an agreed deadline"; and at requirement 7, that the system restrict confidential information, which is section 6's axis. **It says nothing about what a**

record owes when a link in that chain was sought and not found. Section 5.6 adopts requirement 14's deadline at the level of the gap. Its silence on the unfound link is the sentence this document exists to supply, and it is named here so that the supply is seen as an extension of the Principles rather than a departure.

Certificate issuers authenticate at a moment. **They are an input to a record, not a competitor to one,** and a certificate recorded here is qualified as attested.

The pattern across these is the reason this document exists. Several independent bodies of work identify the need for a vocabulary of absence or uncertainty. Two decline in writing to supply one and give their reasons. One theorizes the problem at length and stops short of a schema. One addresses it as critique rather than as instrument. **The gap is not that the problem is unnoticed. It is that the problem is well described and not specified.**

The contribution claimed by this document is narrow and specific: absence as a typed, reasoned, first class entry with a closed vocabulary and an enforced biconditional, plus the separation of withholding from absence. **Everything else here is assembled from existing practice and is described as such.**

Section 12

The provenance of this document

Applying the specification to itself.

- **Sections 3 through 8** describe schema that is **implemented** and **not yet applied to a production database**. Status: built, not run.
- **Section 5's vocabulary** is a decision by the author dated 2026-08-20.
- **Section 6 was not designed.** It was produced by a real maker's refusal on 2026-08-22, the first week of field testing, which falsified the vocabulary in section 5.1. **A specification that survives contact with one maker unchanged has probably not met one.**

- **Section 9's chain** is specified and its production deployment is **not yet obtained**.
- **Adoption by any party other than the author.** State: `not_yet_obtained` , asserted 2026-09-01. **This line previously read "none", undated and with no method, a bare blank about the world, which is the exact thing section 5.4 of this document forbids.** A specification that abolishes the undifferentiated blank should not contain one.

What was searched, so the limit is legible: Zenodo's record API, on 2026-09-01, for the concept DOI and for the specification's title. **Three records returned, all deposited by the author.** No record by any other party.

AND THE POSITIVE CONTROL NOW PASSES, WHICH IS WHY THIS ASSERTION CAN BE RELIED ON. A title search returns this specification's own record first. **In v0.2 and v0.3 this line carried the opposite caveat: the control failed, the instrument could not find the thing known to exist, and the finding was therefore untrustworthy.** The cause was the missing keywords recorded at section 13, and it is now resolved. **An absence reported by an instrument that cannot say yes is not a finding, and this one now is.**

This document is therefore a specification, not a standard. It becomes a standard when somebody else implements it, and not before. **Requirement 12 of section 10 is what will make that event observable;** before it, a conforming record by a third party would have announced nothing at all.

Section 13

Not yet specified

Stated so that silence is not read as completeness.

- Alignment mapping to Linked Art and CIDOC CRM.
- Succession: what happens to a record when its custodian dies.
- Consent and publication rules where a record names a living person.

- Whether any field must be classified for disclosure by default (section 6).
- **Whether an upstream withholding is a distinct case from a subject's own.** Section 6 requires a withheld field to name its withholder, and satisfies that whether the withholder is the record's subject or a third party. **Two field cases in one week suggest the reader may need to tell them apart.** A provenance curator at a national museum described dealers bound by client confidentiality who *cannot reveal details about historic consignors or purchasers even when this information is known to them*, and added that she had no settled way to record it. A painter described the same structure from the other side: she keeps a complete record of every sale except where a gallery handles it, because the collector's name never reaches her. **In both, the record-keeper is in the reader's position rather than the withholder's.** 🚫 **No change is proposed here.** Section 5.1's own test has not been applied to it, and inventing the second case from the first is how a vocabulary bloats. **It is listed so the silence is not read as completeness.**
- **A search of a named source that returned nothing.** "Checked the Getty Provenance Index on this date; the object is not there" is a dated fact about a named source, and it is the evidence on which `unknowable` rests and the work on which `not_yet_obtained` is closed or re-dated. This specification carries it in `absence_reason`, which is prose, so a second researcher cannot find it, count it, or avoid repeating it. **No field is added here.** Whether a negative search is a row of its own kind, and what it must name (the source, the date, the searcher, the query), has not been worked, and inventing it from one week's cases is how a vocabulary bloats.
- **An absence with a direction in time.** Two cases in ten days: a unit number not yet assigned because the object was still being made, and a withholding whose end is bounded but not datable. Neither is any of the four states, because all four describe a fact already true somewhere in the world. Both were carried in prose. **No fifth state is added here**, for the reason 5.1 gives, and the cases are listed so the silence is not read as completeness.
- **Whether a declared value belongs in a record.** A commercial practice adopted 2026-09-02 records a client's own written valuation of an object, attributed and dated, as a sourced field. **The specification does not require a value field and this**

document does not add one. If it should, that is a specification change and a version, and it is not a pricing decision.

Resolved since first publication. *Discoverability of this specification's own deposit* was listed here through v0.3, on the measurement that the Zenodo record carried **no keywords at all**, so that a reader searching for provenance, absence, custody or records would not find it. **Measured again 2026-09-01: the record carries seven keywords:** provenance, declared absence, archival silence, chain of custody, records management, metadata standard, digital preservation. **And a title search now returns it first.** It is recorded as resolved rather than deleted, and it matters beyond itself: **it was the cause of the failed positive control that made section 12's adoption finding untrustworthy in v0.2 and v0.3.**

Resolved since first publication. *Correction semantics: an append only record needs a superseding entry, and its shape is not fixed here* was listed here through v0.3. **It is specified as of 2026-08-30 in section 14**, written after the first substantive correction to a real record occurred and found the specification silent. It is recorded as resolved rather than deleted, for the reason given below.

Resolved since first publication. *Serialization format and a normalization rule for hashing* was listed here in v0.1. It is specified as of 2026-08-24 in the OAC Provenance Verification Kit, **doi:10.5281/zenodo.22083313**, which carries two canonicalization profiles, a reference implementation released into the public domain, and test vectors sufficient for a second implementation to prove itself correct against the first. It is recorded as resolved rather than deleted, because a specification that quietly removes its own declared gaps is performing the failure this document was written against.

Correction semantics

A correction is not an edit. It is a new assertion about a previous assertion, and the previous one continues to exist in the world.

This follows from section 5.4. An entry describes the record at a moment. Correcting it does not make the earlier entry never have been asserted, and where the record has been issued, the earlier assertion is in somebody's hands and cannot be recalled.

14.1 Corrections are appended, never applied in place

A conforming implementation **MUST** represent a correction as a new entry that references the entry it corrects, and **MUST NOT** alter, overwrite or delete the corrected entry.

This is not a preference for audit trails. The record's integrity guarantee is a hash chain over its entries (section 9). **An implementation that edits an entry in place has not corrected the record; it has broken the only thing the chain buys**, and it has done so silently, because a recomputed chain over rewritten content is internally consistent and says nothing.

14.2 The correction vocabulary

```
correction_kind ∈ { transcription, vocabulary, addition, retraction,  
supersession }
```

KIND	WHAT HAPPENED	WHAT A READER SHOULD DO
transcription	the record misstated what a source says	re-read the entry; any prior citation of it is unsafe
vocabulary	a field used wording outside a closed vocabulary	nothing about the object changed
addition	new information; nothing prior was wrong	prior entries stand; there is now more
retraction	an asserted fact is withdrawn as unsupported	revisit anything that relied on it
supersession	a declared absence closed, or an entry replaced by better-sourced information	the old entry is history; the new one is current

The completeness test is section 5.1's and is applied here unchanged: each kind licenses a different action. **A vocabulary whose members produce identical behavior is one kind wearing five names**, and the distinction that matters most is between `addition` and `retraction`, because a reader who cited this record needs to know which one arrived.

14.3 A correction **MUST** state what did not change

The clause with the least precedent and the most value, and it was invented in practice rather than designed.

A correction **MUST** carry an explicit statement of what it leaves untouched.

A reader encountering an amendment cannot otherwise establish its blast radius. *"No custody entry was altered"* is what allows a party who cited this record to know their citation still holds, **without re-reading the whole record and without asking**. Its absence forces every reader to assume the worst, which in practice means assuming nothing and checking everything.

14.4 A correction **MUST** state the status of any issued copy

`issued_copy_status` $\in$ { `none_issued`, `standing`, `superseded`, `withdrawn` }

This is the clause that distinguishes a record from a document. A record that has been sent exists outside the keeper's control. A correction that is silent about the copy in somebody's hands has corrected the keeper's file and left the reader holding a version they do not know is old.

STATUS	MEANING
<code>none_issued</code>	the record has not left the keeper
<code>standing</code>	a copy was issued and is not withdrawn. It carries the earlier wording, and this is stated rather than remedied.
<code>superseded</code>	a corrected copy has been issued to the same party
<code>withdrawn</code>	the issued copy has been recalled and the party told not to rely on it

`standing` is the honest and most common case and must not be treated as a failure. A record issued as a file cannot be reached; **saying so is the remedy available, and it is a real one.**

14.5 Closing a declared absence is a correction

When a gap declared `not_yet_obtained` is closed, the closure **MUST** be recorded as a `supersession` naming the source that closed it, and the absence entry **MUST NOT** be deleted.

Section 5.5 requires an absence to name who can close it, and section 5.6 requires a date by which it will be reviewed. This is the other end of both sentences. Re-dating an elapsed review, or retyping an elapsed `not_yet_obtained` to `unknowable`, is likewise a correction under this section, and **MUST** record why the work was not done or what search exhausted the space. A record whose gaps quietly disappear as they are filled cannot show that anyone did the work, and is indistinguishable from a record that never declared them.

14.6 A correction describes the record, not the world

Section 5.4 applies unchanged. **Correcting an entry asserts that this record was wrong, or is now more complete. It never asserts that the object's history changed.**

14.7 The constraint this section exists to impose

``` corrections(id, object\_id, corrects\_entry\_id, correction\_kind, reason,  
unchanged\_statement, issued\_copy\_status, corrected\_at)

correction\_kind IN (the five) -- closed vocabulary issued\_copy\_status IN (the four) --  
closed vocabulary length(btrim(reason)) > 0 -- a correction MUST say why  
length(btrim(unchanged\_statement)) > 0 -- 14.3 corrected\_at IS NOT NULL -- a  
correction is a dated act ```

**And the prohibition, stated plainly because it is the thesis of this specification applied to itself: a record that changes quietly after issue is the condition this document exists to remove.** A silent correction is not a smaller failure than a declared error. **It is the same failure as an undeclared gap, committed by the keeper rather than inherited from the sources.**

## 14.8 Not specified here

- **How a correction reaches a party holding an issued copy.** issued\_copy\_status records what is true; it does not deliver anything, and no mechanism here obliges or enables notification.
- Whether a correction may itself be corrected, and how a chain of them is read.
- Whether an implementation may refuse to issue a record with an open withdrawn correction.

**Where this section came from.** The first substantive correction to a record written against this specification occurred on 2026-08-30, five days after the record was issued to a third party. **The specification could not say how to make it**, and section 13 listed correction semantics as unspecified. The record improvised: it appended rather than edited, stated what had not changed, and stated that the issued copy was standing and unaltered. **Every requirement above is that improvisation written down**, together with the one before it on 2026-08-25, which corrected a field that had used wording outside a closed vocabulary and said in its own words that *a record that changes quietly after issue is the condition this specification exists to remove*. **The practice reached the rule first, as it did in section 6 and section 5.5.**

---

---

## License and citation

This document is published as an enabling disclosure to establish prior art. Text copyright Christopher Banks, Objects of Affection Collection, 2026. **The structure it describes may be implemented by anyone, and the licence requires that this specification be credited where it is.** Establishing priority is the purpose of publishing it; excluding others is not. Attribution is not a toll on use: it is the mechanism by which a record's reader can find the rules the record was written to.

---

---

## Why v0.7 waited, what the standards body actually said, and why it deposits now (2026-10-13)

**v0.7 was held on the author's ruling of 2026-09-22, "hold the deposit until after Nuremberg."** The reason was stated at the time and was sound: the `never_required` question had been put to the CIDOC CRM SIG as part of issue 723, the SIG met 6 to 9

October 2026, and the same meeting was to consider whether the pattern maps to a typed E13 or to subclasses. Depositing before that would have meant depositing again a fortnight later, and **a version published two weeks before the answer it is waiting for is a version that exists to look current.**

**The meeting has happened and it did not produce those answers.** Issue 723 was heard on 2026-10-08 and was deferred rather than closed. The group's position was that the pattern is neither of the two existing constructs put to it and may need a construct of its own, and it asked for further cases, more than once. **The never\_required question was not answered. The CRM mapping was not settled.**

**The hold was waiting for an answer, and the answer is "not yet, bring evidence."** Continuing to wait would bind this specification to a question the standards body has itself deferred, with no date on it. **A hold whose release condition has become indefinite is not caution; it is a document that never ships.**

**So v0.7 deposits carrying what is actually finished, and says plainly what is not.** The two normative changes in this version, section 6.0 and section 5.7 with requirement 15, are the author's own, were found in the author's own records, and depend on nothing the SIG has yet to decide. The open questions are recorded as open, in section 5.1 and in section 13, rather than resolved by default because a deposit was due.

**Issue 723 remains open at the CIDOC CRM SIG and this specification does not claim otherwise.** Nothing here is endorsed, adopted or ratified by that body, and a deferral is not an approval.

**What this means for a record already written.** v0.6 (doi:10.5281/zenodo.22751179) remains valid for every record written against it. **Section 14.1 governs: a record is never rewritten to a later version of this specification,** so a record that declared conformance to v0.6 continues to conform to v0.6 and is not made deficient by anything in v0.7. A record written from today declares conformance to v0.7.

**And one thing is recorded rather than tidied away.** Between 2026-09-22 and this deposit there was no published rule for the section 6.0 case, because the fix was written and deliberately held. A recorder who met that case in those three weeks had the choice of typing the value under section 5 against an unpublished draft or leaving it where the published rule put it. **That gap was real, it was the cost of holding the version, and it is stated here rather than discovered later.**