enosema

GUIDE
Guide

EFG 2519:2026

Terminology management — Concept registers — Operation guide

First edition

Reference number
EFG 2519:2026

© EFS 2026

(c) 2026 Enosema Foundation.

This document is made available under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Unless otherwise specified, or required in the context of its implementation, you are free to use, share, and adapt this document, provided that appropriate credit is given to the Enosema Foundation and that any changes are indicated. The full license text is available at https://creativecommons.org/licenses/by/4.0/.

Enosema Foundation
2885 Sanford Ave Sw Suite 32361
Grandville MI 49418-1342
USA

E-mail: info@enosema.org

Published in the USA

© EFS 2026 – All rights reserved

Contents

Page

Foreword ..... **v**

Introduction ..... **vi**

1 Scope **1**

2 Normative references **1**

3 Terms, definitions and abbreviated terms **2**
3.1 Terms and definitions 2
3.2 Abbreviated terms 3

4 Operating context and register scope **3**
4.1 Choosing what the register is for 3
4.2 Audience analysis worksheet 3
4.3 Boundary decisions 4
4.4 Context records 4
4.5 Stakeholder interview question bank 4
4.6 Register scope — worked boundary statement 5

5 Governance in practice **5**
5.1 Establishing the control body 5
5.2 Roles in daily operation 6
5.3 Meeting and ballot practice 6
5.4 Decision records 6
5.5 Appeals practice 6
5.6 Role assignment in practice (RACI) 7
5.7 Meeting minutes template 7
5.8 Asynchronous-first governance 7

6 Planning the register **7**
6.1 Identifier scheme design 7
6.2 Edition and versioning policy 8
6.3 Planning criteria of subdivision 8
6.4 Planning for breadth and depth 8
6.5 Planning publication 9
6.6 URN resolution design 9
6.7 Edition policy — a worked example 9

7 Concept operations **9**
7.1 Proposal intake 9
7.2 Review practice 10
7.2.1 Definition review (per ISO 704, ISO 10241-1) 10
7.2.2 Placement review 10
7.2.3 Relation review 10
7.2.4 Record review 10
7.3 Decision records and acceptance 11
7.4 Change operations 11
7.4.1 Extending 11
7.4.2 Splitting 11
7.4.3 Merging 11
7.4.4 Instantiating 11
7.4.5 Retiring 11
7.5 Publication runbook 11
7.6 A worked review 12
7.7 Ballot announcement template 12
7.8 Publication log 13
7.9 Intake service levels 13
7.10 Triage table 13

© EFS 2026 – All rights reserved

8 Localization operations **14**
8.1 Declaring the localization model 14
8.2 The definition-bearing workflow 14
8.3 The term-only workflow 14
8.4 Assignment of work 15
8.5 Record practice 15
8.6 Equivalence assessment worksheet 15
8.7 Cadence 15

9 Relation management practice **16**
9.1 Choosing a relation type 16
9.2 Implementing the constraints 16
9.3 Review practice for relations 16
9.4 Cross-vocabulary relations 17
9.5 Relation families in records — worked patterns 17
9.5.1 Hierarchical (generic) 17
9.5.2 Hierarchical (partitive, binary) 17
9.5.3 Partitive as an n-ary relation file 17
9.5.4 Sequential as an n-ary relation file 18
9.5.5 Lifecycle pair 18
9.5.6 Mapping across vocabularies 18
9.6 Mapping vocabulary table 19

10 Tooling, automation and repository practice **19**
10.1 The reference toolchain 19
10.2 Repository layout 19
10.3 Validation automation 20
10.4 Branching and review practice 20
10.5 Support records 20
10.6 CI workflow — a complete example 20
10.7 Validator invocation patterns 21
10.8 Choosing what to automate first 21

11 Evaluation in practice **21**
11.1 Monitoring 22
11.2 The periodic review 22
11.3 Internal audit 22
11.4 Self-assessment against EFS 2513 22
11.5 Sample review record 23
11.6 Audit trace walkthrough 23

12 Improvement practice **24**
12.1 Nonconformity handling 24
12.2 Improvement decisions 24
12.3 The improvement log 24
12.4 Continual improvement, realistically 24
12.5 Improvement log — worked entries 24

Annex A (normative) **Concept proposal form** **26**

Annex B (normative) **Decision record template** **29**

Annex C (normative) **Self-assessment checklist against EFS 2513** **31**

Annex D (normative) **Worked example — the ES Relation Vocabulary** **35**

Annex E (normative) **Requirements class adoption worksheet** **39**

Bibliography **41**

© EFS 2026 – All rights reserved

Foreword

The Enosema Foundation (“ES”) is the premier non-profit standardization organization for terminology standardization and terminology related processes such as management. It facilitates the education, standardization, research, promotion, definition, and usage of terminology resources and management practices globally.

ES works with international partners and experts across the globe, reflecting the international nature of its mission. More information about ES is available on the official website (https://www.enosema.org).

The procedures used to develop this document and those intended for its further maintenance are described in the ES Directives.

In particular, the different approval criteria needed for the different types of ES documents should be noted. This document was drafted in accordance with the editorial rules of the ES Directives.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ES shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be provided in the Introduction.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

This document was prepared by Technical Committee **Terminology**.

This document is a Guide (EFG 2519:2026) within the ES specifications and directives series (EFD 102, clause 4). It is informative: it describes practice that has proven useful, not requirements that conformity is assessed against. Wherever this document gives guidance, EFS 2513 states the corresponding requirement; the two documents are structured to be read together, clause by clause.

© EFS 2026 – All rights reserved

Introduction

0.1 Why an operation guide

EFS 2513 specifies what a conforming concept management system does. It says that a register shall have a control body, that changes shall go through proposal and review, that records shall be machine-readable, that localizations shall carry equivalence degrees. It does not state how an organization establishes and operates those arrangements in daily practice.

This guide fills that gap. It is to EFS 2513 what ISO 9004 is to ISO 9001: guidance for organizations that intend to implement, operate, and steadily improve a concept register, written from practice rather than from principle. Every clause of this guide corresponds to a clause of EFS 2513, and its annexes contain the working artifacts an organization needs — proposal forms, decision records, checklists, and a complete worked example from a register that ES itself operates.

0.2 How to use this guide

Reading paths by situation:

— **Starting a register**: read clauses 4 to 7 in order, then Annex D (the worked example), and adopt Annex A (proposal form) and Annex B (decision record) as the first working documents of the register.

— **Achieving conformity with EFS 2513**: apply Annex C (the self-assessment checklist) to current practice; each line cites the EFS 2513 requirement that it examines. Clause 11 explains how to run the assessment and what to record.

— **Improving an existing register**: run the assessment first, then apply clause 12 (improvement practice) to the findings.

— **Extending an existing system to localization**: clause 8.

— **Extending an existing system to a public, machine-readable publication**: clauses 7 (publication runbook), 9 (relation practice), and 10 (tooling), with EFS 2516 alongside.

0.3 Conventions

— The word **should** marks recommended practice; an organization may deviate, and records the deviation and its reason.

— Templates in annexes are starting points: adapt names, fields, and wording to the organization, and keep the structure.

— Examples marked as coming from the ES Relation Vocabulary are real: the dataset is published by ES at the Enosema concept register, and its files are cited verbatim.

— Guidance that corresponds to a specific EFS 2513 requirement is cross-referenced inline in the form (EFS 2513, «clause»), so the two documents can be read side by side.

© EFS 2026 – All rights reserved

Terminology management — Concept registers — Operation guide

1 Scope

This guide describes practice for establishing, operating, evaluating, and improving a concept register — the register-facing part of a concept management system as specified in EFS 2513.

It gives guidance, templates, checklists, and worked examples for:

— scoping a register and analysing its audiences (clause 4);

— establishing and operating governance: control bodies, decision records, appeals (clause 5);

— planning identifiers, editions, and concept system structure (clause 6);

— day-to-day concept operations: intake, review, acceptance, change, retirement, and publication (clause 7);

— localization practice (clause 8);

— relation management practice (clause 9);

— tooling, automation, and repository practice (clause 10);

— evaluation and improvement practice (clauses 11 and 12); and

— working artifacts: proposal form, decision record, self-assessment checklist, a complete worked example, and an adoption worksheet (annexes A to E).

This guide is informative. It places no requirements; EFS 2513 places the requirements, and this guide suggests ways to meet them. An organization is free to follow different practice and still conform to EFS 2513, provided the requirements themselves are met.

This guide does not supersede EFD 101 (standardization and publication processes) or EFD 102 (standard document requirements); ES's own documents follow those directives, and this guide's examples are drawn from practice conducted under them.

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.

EFS 2504:2025, *Terminology management — Multilingual terminology*

EFS 2510:2026, *Terminology management — Concept systems and concept relations*

EFS 2513:2026, *Terminology management — Concept management systems — Requirements*

EFS 2516:2026, *Terminology management — Machine-readable terminology interchange*

EFD 101:2026, *Enosema Directive: Standardization and publication*

EFD 102:2026, *Enosema Directive: Standard document requirements*

ISO 704, *Terminology work — Principles and methods*.

© EFS 2026 – All rights reserved

ISO 1087-1[1)] , *Terminology work — Vocabulary — Part 1: Theory and application*.

ISO 19135-1[2)] , *Geographic information — Procedures for item registration — Part 1: Fundamentals*.

ISO 10241-1, *Terminological entries in standards — Part 1: General requirements and examples of presentation*.

ISO 639-3[3)] , *Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages*.

ISO 15924, *Information and documentation — Codes for the representation of names of scripts*.

ISO 24229, *Information and documentation — Codes for written language conversion systems*.

W3C SKOS, *SKOS Simple Knowledge Organization System Reference*, W3C Recommendation, https://www.w3.org/TR/skos-reference/

3 Terms, definitions and abbreviated terms

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in EFS 2510, EFS 2513 and EFS 2516 apply.

This guide additionally uses the following working terms.

3.1.1
intake
first handling of a proposal: completeness check, triage, and assignment to a reviewer, before any technical judgement is made

3.1.2
decision record
controlled record of one decision of the control body: what was proposed, what was decided, why, and on whose vote

3.1.3
publication run
one execution of the register publication procedure: validation, build, checks, and release of the register's published artifacts

3.1.4
register edition
a published, immutable state of the register, identified by year or version, whose records do not change after publication

1) Withdrawn.

2) Withdrawn.

3) Withdrawn.

© EFS 2026 – All rights reserved

3.1.5
submission channel
the mechanism by which proposals reach the register, typically the issue tracker of the register's source repository

3.2 Abbreviated terms

CI	continuous integration
CMS	concept management system
ES	Enosema Foundation
GCR	glossarist concept register
RA	registration authority
SKOS	Simple Knowledge Organization System
TC	technical committee
URN	uniform resource name

4 Operating context and register scope

This clause gives guidance for the requirements of EFS 2513, clause 4.

4.1 Choosing what the register is for

Before any record is created, write down the register's purpose in one sentence that names the audience and the decision the register supports. Registers that skip this sentence accumulate concepts nobody uses and cannot answer the question "why does this item exist?" at review time.

Useful forms of the sentence:

— "The X register provides the authoritative set of Y used by Z when they do W."

— "This register is the single point of reference for concepts defined in the documents of organization Q."

From the purpose sentence, derive the register type. The common patterns:

Table 1

Pattern	What it holds	Main consumers
Vocabulary register	Controlled vocabulary: relation types, statuses, categories	Systems that validate or exchange data
Terminology register	Domain concepts with designations and definitions	Authors, translators, content systems
Reference-data register	Code lists and item sets with lifecycle	Interoperating systems, data pipelines
Standards-derivative register	Concepts extracted from published documents	Readers of the documents, tool vendors

A register may be several of these at once; record the mix, because it changes the review emphasis (terminology registers review definitions hardest; reference-data registers review lifecycle hardest).

4.2 Audience analysis worksheet

For each audience, record the answers to five questions. The answers drive the register's scope, publication formats, and stability commitments.

© EFS 2026 – All rights reserved

a) **Who** consumes the concepts — roles, not individuals.

b) **What decisions do they make** with the concepts?

c) **What breaks for them** if a concept changes: translations, schemas, published citations, training material?

d) **What format do they need** — browsable pages, CSV, TTL, JSON-LD, TBX, a GCR package?

e) **How do they cite** a concept — by identifier, by term, by document clause?

EXAMPLE The ES Relation Vocabulary register, from the purpose sentence outward:

— Purpose: "provides the authoritative set of concept relation types used in ES concept systems and exchanged between terminology management systems."

— Audiences: terminology management systems (validate relation type identifiers), standards authors (cite relation types when drafting), SKOS consumers (project vocabularies).

— Breakage: a renamed relation type identifier breaks validating systems — hence the stability commitment (clause 6) and the never-rename rule (clause 7).

— Formats: browsable site, TTL, JSON-LD, CSV (EFS 2516 bulk exports).

— Citation: by URN `urn:enosema:relation-type:<identifier>` and by the identifier itself in EFS 2510, clause 6.

4.3 Boundary decisions

Two boundaries are worth deciding explicitly and writing down, because they are the most common sources of scope disputes later:

— **What is not in the register.** Name the nearest topics that are excluded. A boundary by exclusion is checkable; a boundary by impression is not.

— **Where concepts come from.** A register that only standardizes its own concepts reviews differently from one that adopts concepts from other bodies. For adopted concepts, record the source and the mapping relation type (`exact_match`, `close_match`, …) at intake, not later.

4.4 Context records

Keep the following as controlled records (a directory in the register repository is enough):

— the purpose sentence and register type;

— the audience worksheet;

— the two boundary decisions; and

— the date of the last review of these records.

EFS 2513 (4.3) requires the scope to be reviewed when the context changes. Practical trigger list: a new consumer class appears; a consuming system is retired; the organization's charter or product scope changes; a review interval elapses (clause 11).

4.5 Stakeholder interview question bank

When preparing the audience worksheet (Clause 4), interview real consumers rather than assuming. The questions that produce usable answers:

a) "Show me the last time you used one of our concepts. What did you look at, and what were you looking for?" — surfaces real entry points (pages, exports, citations) instead of imagined ones.

© EFS 2026 – All rights reserved

b) "What would you do if this concept's definition changed tomorrow?" — surfaces breakage: translations, schemas, training material.

c) "How do you currently check that you are using the right term?" — surfaces validation needs (machine-checkable identifiers) versus lookup needs (search, browse).

d) "Which concepts do you use most? Least?" — surfaces the depth planning inputs of clause 6.

e) "What do we publish that you cannot use?" — surfaces format gaps and the publication profile requirements.

f) "Who else in your organization consumes these?" — surfaces the second-order audience that scope planning usually misses.

Record the interviews with dates; the worksheet cites them. A scope statement grounded in six interviews survives review far better than one grounded in plausibility.

4.6 Register scope — worked boundary statement

The two boundary decisions of Clause 4, stated for a concrete register:

— **Excluded by name**: designation-level relations (those are recorded in the datasets that use this vocabulary, not in the vocabulary itself); per-deployment deployment metadata; and any relation type not yet accepted by the control body (proposals live in the issue tracker until decided).

— **Origin**: all items originate in the register's own decisions; items adopted from other bodies enter through a proposal that cites the source and states the mapping relation (`exact_match`, `close_match`, …) at intake.

5 Governance in practice

This clause gives guidance for the requirements of EFS 2513, clause 5.

5.1 Establishing the control body

The control body needs a written charter. Half a page is enough; the essential contents are:

a) **Composition**: how many members, appointed by whom, term length.

b) **Quorum and voting**: what counts as a decision; whether the chair votes; handling of abstentions (EFD 101, balloting practice is a good default: Yes / No / Abstain, two-thirds Yes for acceptance, "No" votes carry constructive comments).

c) **Decision channels**: synchronous (meetings) and asynchronous (written ballot on the issue tracker); asynchronous ballots should have explicit opening and closing dates.

d) **Records**: where decision records (Annex B) are kept; who writes them (the register manager is a good default).

e) **Appeals**: to whom an appeal is addressed, in what time it is acknowledged, and who hears it other than the original decision-makers (EFS 2513, 5.3).

EXAMPLE A minimal charter (working text, adapt freely):

> The X Register Control Body consists of three to five members appointed by the Board of Y. The chair is appointed by the Board and does not vote except to break ties. Decisions are taken by written ballot on the register issue tracker, opened for at least 10 working days, and pass by two-thirds of votes cast excluding abstentions. Every decision is recorded by the register manager as a decision record within five working days. Appeals are addressed to the Board of Y, which hears them without participation of the members of the original decision.

© EFS 2026 – All rights reserved

5.2 Roles in daily operation

Map the roles of EFS 2513 (5.5) onto named practices:

— **register manager**: runs the publication runs (clause 7), writes decision records, maintains register.yaml, and is the contact point named on the register site;

— **concept owners**: one per concept domain or subdivision; they draft proposals and respond to review comments on their concepts;

— **reviewers**: check proposals against the review checklists (clause 7); at least one reviewer should not be an author of the proposal under review;

— **submitters**: everyone entitled to open proposals through the submission channel.

A single small team may hold several roles; the separations that matter are: the register manager should not be the sole reviewer of their own proposals (EFS 2513, 5.5), and appeals should not be decided by the authors of the contested decision.

5.3 Meeting and ballot practice

Most registers run on asynchronous ballots; meetings are for deadlock and design discussions. Practice that works:

— every ballot issue carries the proposal link, the review summary, and the exact question to be voted on;

— ballots state their window (`opens X`, `closes Y`, time zone included);

— results are posted in the same issue as the ballot, with the decision record linked;

— meetings, when held, produce minutes whose actions reference ballot issues — never the reverse.

5.4 Decision records

The control body records one decision record for each decision (Annex B). The rationale field of the decision record is its most valuable content: a decision log that states grounds allows a new contributor to understand the register without reading its meeting history.

Minimum fields: identifier, date, proposal link, decision (accepted / returned / rejected / withdrawn), rationale, votes, effects (records created, identifiers issued, editions affected).

Keep decision records numbered and immutable once closed; corrections are new records that supersede, never edits.

5.5 Appeals practice

An appeal is processed in four steps, with dates at each:

a) **Acknowledgement** (within the charter's stated time): appeal received, forwarded to the appeal body.

b) **Standing check**: is the appellant affected, is the appeal within time, does it state grounds?

c) **Hearing**: the appeal body reviews the decision record, the proposal, and the grounds; it may consult the control body but its members who decided the original matter do not vote.

d) **Outcome**: appeal accepted (decision reversed or redone) or rejected; the outcome is recorded as a decision record referencing the contested one.

Publish the appeal outcomes summary in the register's periodic review (clause 11); a register with repeated appeals on the same ground has a process defect, not a participant problem.

© EFS 2026 – All rights reserved

5.6 Role assignment in practice (RACI)

Map operations to roles once, so proposals do not stall on “who does this?”. R = responsible, A = accountable, C = consulted, I = informed.

Table 2

Operation	Register manager	Reviewer	Control body	Submitter
Proposal intake	R	I	I	A (submits complete form)
Technical review	C	R/A	I	C
Control-body decision	R (records)	C	A	I
Publication run	R/A	I	I	I
Appeal handling	C	I	C	A (raises appeal)
Periodic review	R (runs)	C	A (approves findings)	I

5.7 Meeting minutes template

Meetings produce minutes whose actions always reference ballot or proposal issues:

```
Register Control Body — meeting 2026-11-05
Present: <names>
Agenda:
1. Review outcome, proposal #87 — consensus unclear -> ballot #87 opened
   (closes 2026-11-09). Action: manager (DR after close).
2. Criteria register: proposal to add criterion "publication-channel"
   -> owner to draft wording against ISO 704 subdivision principles.
   Action: owner, by 2026-11-19.
3. Intake times trending high -> intake checklist to move to CI template.
   Action: manager, next sprint.
```

Figure 1

5.8 Asynchronous-first governance

Registers whose participants span time zones run almost entirely on written ballots; meetings are the exception for deadlock and design discussions. Two consequences worth planning for:

— the charter’s decision channel default should be the written ballot, with meetings convened only when a ballot fails or a question is design-level; and

— every ballot thread should be self-contained — proposal link, review summary, exact question, voting mechanics — because for many members the thread **is** the meeting.

6 Planning the register

This clause gives guidance for the requirements of EFS 2513, clause 6.

6.1 Identifier scheme design

The identifier scheme is decided once, written down as a rule, and never broken. Practice has shown the following rules to be durable:

a) **Separate the planes.** Item identifiers are permanent; editions and statuses are not part of the identifier. `relation-type/supersedes` is an identifier; `superseded` is a status; the year is an edition. Do not bake dates or states into identifiers.

b) **One URN namespace per register**, of the form `urn:<org>:<register>:<collection>:<local-id>`, registered once and documented on the register site.

EXAMPLE The ES Relation Vocabulary:

© EFS 2026 – All rights reserved

Table 3

Item	URN
Dataset	`urn:enosema:relation-vocabulary:2026`
Relation type item	`urn:enosema:relation-type:<identifier>`, for example `urn:enosema:relation-type:broader_generic`

a) **Identifiers are for machines and citations, names are for humans.** Never rename an identifier to improve its readability; improve the preferred designation instead.

b) **Reserve the right to sub-collections.** If the register may grow collections later (relation types today, designation types tomorrow), give each collection its own namespace segment from day one, so that URNs never need to change shape.

6.2 Edition and versioning policy

The edition and versioning policy answers three questions in writing.

— **What is an edition?** An edition is a published, immutable snapshot (a register edition, 3.1.4). Item editions may differ from dataset editions; both policies are recorded.

— **What is a new version of an item?** A new version changes the intension — the definition, the relations or the designation status — of the item; editorial corrections within one publication run are not a new version.

— **When does a reference break?** A reference does not break. The policy of EFS 2513 (7.6.4) applies: identifiers are issued once and resolve for the operational life of the published records, either to the current record or to the retired record together with its successor information. Editions give consumers the as-of view; the current view resolves to the latest edition.

6.3 Planning criteria of subdivision

EFS 2510 requires every subdivision to carry its criterion; in operation, the practical step is a **criteria register** — a page in the register repository listing every criterion in use, its wording, and the concepts it subdivides.

EXAMPLE A criteria register entry:

```
criterion: power-source
wording: "Subdivides transport equipment by its primary energy source."
applies-to: road-vehicle
subdivisions: motorized, human-powered
since: 2026-01
```

Figure 2

When a proposal introduces a new criterion, the reviewer checks the criteria register first: an existing criterion that produces the same subdivision means the proposal should join it, not fork it.

6.4 Planning for breadth and depth

Practical heuristics (EFS 2513, 6.2):

— A subdivision level earns its place when a consumer decision depends on it, not when a distinction merely exists.

— Two to seven coordinate concepts per subdivision is comfortable for review; larger sets usually hide a missing intermediate concept or a missing criterion.

— Plan depth by consumption: every level a consumer must traverse to use a concept should correspond to a decision they make.

© EFS 2026 – All rights reserved

6.5 Planning publication

Decide at planning time, not at first-release time:

— the publication formats (EFS 2516: CSV, TTL, JSON-LD at minimum);

— the publication cadence (continuous on merge, or scheduled runs);

— the base URL and path of the published site; and

— whether editions are published as separate URLs or as one current URL with archives.

6.6 URN resolution design

A URN that cannot resolve is a string. Decide at planning time how each URN form resolves:

Table 4

URN form	Resolves to	Mechanism
Dataset URN	The dataset's published page	Register site routing
Concept URN	The concept's page on the published site	Stable URL derived from the identifier (EFS 2516, 8.3)
Collection URN	A listing page or SKOS collection	Generated pages

Two practices make this durable:

— derive public URLs from identifiers mechanically (one rule, no exceptions), so the resolution works before any page is hand-tuned; and

— keep old URLs working when editions change: an edition's pages remain reachable at their editioned paths even after a new edition publishes.

6.7 Edition policy — a worked example

The policy questions of clause 6 answered for a concrete register:

Table 5

Question	Answer (recorded)
What is an edition?	A publication run that changes the concept set or any intension. The dataset id is editioned (`<register>-<year>`); runs that only correct editorial defects refresh the same edition.
What is a new version of an item?	A change to the item's definition, relations, or designation status. Editorial corrections within a run are not a version change.
When do references break?	Never. Concept URNs resolve to the current edition; editioned paths resolve to the edition as published.

The answers are recorded once, cited at every review, and changed only by decision record — the policy is part of the register's contract with its consumers, and changing a contract has a decision trail.

7 Concept operations

This clause gives guidance for the requirements of EFS 2513, clause 7.

7.1 Proposal intake

When a proposal arrives through the submission channel (for this guide's assumed setup: an issue in the register repository), the register manager performs intake — a mechanical check, not a review:

a) **Completeness**: does the proposal carry all fields of the proposal form (Annex A)? Incomplete proposals are returned with the missing fields named.

© EFS 2026 – All rights reserved

b) **Triage**: is this a new concept, a change, or a question? Questions are answered or turned into proposals by the concept owner.

c) **Assignment**: one reviewer is assigned; the proposer is told who and the expected review window.

d) **Label**: the proposal issue is labelled with the affected concept system(s) and the operation type (create, revise, split, merge, retire, localize).

Intake should happen within a few working days; the clock on review starts after it.

7.2 Review practice

Reviews check the four dimensions of any proposal. Checklists keep reviews consistent between reviewers; the four checklists below are meant to be pasted into the review comment and ticked.

7.2.1 Definition review (per ISO 704, ISO 10241-1)

Table 6

Check	What to look for
Intension	The definition states the characteristics that delimit the concept, not an extension or a purpose
Genus	It begins with the superordinate concept, not with an article or the term itself
No requirements	No shall/should/may; definitions do not prescribe
Delimitation	Coordinate concepts within the subdivision remain distinguishable after the definition
Circularity	No definition by synonym within the same register

7.2.2 Placement review

Table 7

Check	What to look for
Criterion	The proposal names a criterion of subdivision, checked against the criteria register (clause 6)
Coordinate set	The proposed concept's coordinate concepts are listed; their union behaves sensibly (MECE check per EFS 2510, 4.3)
Dimensionality	Parallel criteria are kept as parallel subdivisions, not merged

7.2.3 Relation review

Table 8

Check	What to look for
Vocabulary	Relation type identifiers exist in the adopted vocabulary (EFS 2510, clause 6)
Direction	The direction matches the type's semantics; inverse pairs are consistent
N-ary rule	Joint decompositions use n-ary relations per the binary-versus-n-ary decision rule (EFS 2510, 7.2)
Lifecycle	Lifecycle relations link register items where both sides exist

7.2.4 Record review

Table 9

Check	What to look for
Record format	Concept file conforms to EFS 2516, clause 4; parses; fields present
Identifiers	Identifier scheme followed; URN correct; localized-concept map consistent
Sources	Claims are supported by cited sources

© EFS 2026 – All rights reserved

7.3 Decision records and acceptance

The control body decides on the review outcome. Three outcomes are normal:

— **Accepted**: the proposal is merged; a decision record (Annex B) is written; identifiers are issued at merge time;

— **Returned**: specific, actionable defects are listed; the proposal returns to the submitter without a decision;

— **Rejected**: the grounds are recorded in the decision record; rejection does not prevent a differently-framed future proposal.

For routine, uncontroversial proposals, the control body may delegate acceptance to review consensus; the charter should say so explicitly, and the decision record should record the delegation.

7.4 Change operations

Each change operation of EFS 2513 (7.5) has a concrete record pattern. The patterns below use the vocabulary of EFS 2510, clause 6.

7.4.1 Extending

Add the subordinate concept; record the criterion; check that the new concept's coordinate set updates the completeness picture of any n-ary relation over the parent (a previously `complete` rake may become `partial` — flag it in the proposal).

7.4.2 Splitting

a) Create the successor concepts.

b) On the predecessor record: set status to `superseded`; add the lifecycle relation `supersedes`/`superseded_by` to each successor.

c) Move or re-express relations of the predecessor: relations whose semantics transfer go to the appropriate successor, and the move is stated in the proposal.

d) Keep the predecessor record published; identifiers do not break.

7.4.3 Merging

a) Create the single successor.

b) Link it to each predecessor with `supersedes`/`superseded_by`.

c) Record in the successor's notes the mapping from predecessor designations; translation memories and consuming systems rely on this mapping.

7.4.4 Instantiating

Add the individual concept with the `instance_of`/`has_instance` pair; individual concepts rarely need new definitions, but they need a designation and a source.

7.4.5 Retiring

Set the record's status; add the retirement relation to the successor if one exists; leave everything else in place. Retirement is a status change, not a deletion — there is no deletion operation in a conforming register.

7.5 Publication runbook

A publication run (EFS 2516 formats) follows five steps; automate all of them (clause 10):

© EFS 2026 – All rights reserved

a) **Validate**: run the dataset validator — record format, identifier scheme, relation constraints (the ac/lc/na constraints of EFS 2510, clause 9), localization fields.

b) **Build**: run the publication toolchain to produce the site and the bulk exports (CSV, TTL, JSON-LD) per EFS 2516, clause 7.

c) **Check**: verify export row counts against concept counts; verify that no export dropped concepts or localizations; verify links resolve.

d) **Package**: produce the GCR package for the edition.

e) **Publish**: release site and exports; record the edition and the run in the publication log.

Table 10

Check at every run	Expected result
Validation errors	Zero; any error fails the run
Export concept count	Equals dataset concept count, per format
TTL parses	As RDF, no parse errors
Identifiers resolve	Every `data.uri` resolves on the published site

A failed run publishes nothing; the register's published state only moves forward on a fully green run.

7.6 A worked review

What a review actually looks like when pasted into a proposal thread. The reviewer applies the four checklists of Clause 7 and answers in one comment:

> Reviewed per EFG 2519 clause 7.
>
> **Definition**: genus present ("controlled vocabulary" — good); no modal verbs; delimits against `term-status` and `system-status`, which share the family. One defect: the definition says "lifecycle status of a code" but the item class also covers systems; propose "lifecycle status of a register item".
>
> **Placement**: criterion `lifecycle-stage` exists in the criteria register and already generates `code-status`/`system-status` — this proposal joins it. Coordinate set fine.
>
> **Relations**: `supersedes`/`superseded_by` pair consistent; both referenced items exist (validated); no rakes affected.
>
> **Record**: file parses; URI matches the scheme; localized-concept map consistent; source cited to the SCHEMA.adoc of the source data.
>
> One required change (definition scope), otherwise conforming. Returned to the submitter for the definition fix; recommend acceptance after the change.

The decision record then references this review by link, and the publication run picks up the merged record.

7.7 Ballot announcement template

```
Ballot: Proposal #87 — accept ranks_above / ranks_below
Opens:  2026-11-02 (any time)
Closes: 2026-11-09 23:59 UTC
Votes:  yes / no / abstain, as reactions on this issue
Passes: two-thirds of yes+no, abstentions not counted
Question: Should the register accept "ranks_above"/"ranks_below"
          as relation types in the adopted vocabulary?
Review summary: <link to the review comment>
```

© EFS 2026 – All rights reserved

```
Decision record to follow: DR-2026-014
```

Figure 3

7.8 Publication log

One line per publication run, kept in the register repository:

```
- run: 2026-11-11T10:15Z
  trigger: merge of PR #12 (decision DR-2026-014)
  validation: 0 errors
  concepts: 109
  exports: [csv 110 lines, ttl, jsonld]
  edition: es-relation-vocabulary-2026
  result: published
```

Figure 4

7.9 Intake service levels

State the intake service levels in the charter or repository README, so submitters know what to expect and the register manager knows what is owed:

Table 11

Step	Expected window	Clock starts
Intake (completeness, triage, assignment)	3 working days	Proposal opened
Technical review	15 working days	Intake complete
Control-body decision (ballot)	10 working days ballot + 5 for the record	Review complete
Publication run after acceptance	Next scheduled run (or immediate for urgent consumer-reported fixes)	Decision record written

The windows are commitments to consumers as much as to submitters: intake and decision times are monitoring metrics of clause 11, and service levels that are stated but repeatedly missed are exactly the nonconformity the internal audit looks for.

7.10 Triage table

The intake triage decision, as a table:

Table 12

If the submission is…	It is…	Handled as
A complete proposal form	A proposal	Review path (clause 7)
A question about a concept	A query	Answered by owner; may become a proposal
A defect report on a published record	A consumer report	Urgent track (clause 12); recorded correction if confirmed
A request to rename an identifier	Declined by rule	Explained with the never-rename decision; a new preferred designation may be proposed instead
A proposal with missing fields	Incomplete	Returned with the missing fields named (clause 7, intake)

© EFS 2026 – All rights reserved

8 Localization operations

This clause gives guidance for the requirements of EFS 2513, clause 8, and applies the two localization models of EFS 2522 (clause 5) to register operation, with EFS 2504.

8.1 Declaring the localization model

A register operates one of two localization models, and the model is declared in the register charter:

— **the definition-bearing model**: each declared localization carries its own definition, and equivalence degrees are recorded per EFS 2504 and EFS 2522; and

— **the term-only model**: the authoritative-language definition serves all localizations, and each localization carries its designations, grammar, normative status and — for converted forms — its ISO 24229 writing system conversion system, without a localized definition.

The declared model decides what completeness means for the register (EFS 2522, requirement C-5): under the definition-bearing model, a localization without a definition is incomplete; under the term-only model, it is the declared form. The model is changed only by a decision of the control body, because the change alters the review workload and the consumer contract of every localization.

8.2 The definition-bearing workflow

Under the definition-bearing model, a localization moves through five states, and the state is visible in the proposal or issue that carries it:

a) **Requested**: a localization is required, by policy or by a recorded consumer need.

b) **Drafted**: a translator or domain expert produces the draft designations and definition.

c) **Assessed**: the equivalence degree is assigned (8.6).

d) **Reviewed**: a reviewer checks the terminology method as well as the language — designation type, normative status, and definition intension.

e) **Accepted**: the control body, or its delegate, accepts the localization. Acceptance of any degree other than `full` is a recorded decision with reasons (EFS 2513, 8.3).

8.3 The term-only workflow

Under the term-only model, the workflow concentrates on designations and on the authoritative definition:

a) **Designation production**: a language expert produces the localized designation set, with term types, normative status and grammar information per the language model of the register.

b) **Conversion control**: where a localized form is a romanization or transliteration, the record carries the target script (ISO 15924) and the ISO 24229 conversion system that governs the conversion (EFS 2504; EFS 2522, Group C).

c) **Review**: a reviewer checks the designations against the authoritative definition of the concept, in the authoritative language; the reviewer confirms that each localized designation can carry the intension of the concept in its language.

d) **Acceptance**: the localization is accepted as an equivalent designation set, and the entry records the authoritative definition as the definition of record for all languages.

The term-only model does not remove the equivalence question. A language expert who finds that no designation can carry the authoritative intension raises the gap as a recorded finding, per the resolution steps of the register charter.

© EFS 2026 – All rights reserved

8.4 Assignment of work

Localization work splits into two competences, and proposals state which competence a task requires:

— **translation drafting** (language competence): producing candidate designations and, under the definition-bearing model, candidate definitions; and

— **equivalence assessment** (terminology competence): deciding the degree, writing the equivalence note, and checking the position of the concept in the target-language concept system.

Assigning both competences to one person is acceptable when one person holds both. A translation produced by a tool without an equivalence assessment step is not accepted under either model.

8.5 Record practice

Each localization record carries the fields of EFS 2504 and EFS 2516: the ISO 639-3 language code; the ISO 15924 script, wherever the language is multiscript or the form is a romanization or transliteration; the ISO 24229 writing system conversion system identifier for converted forms; and, under the definition-bearing model, the equivalence degree and note.

8.6 Equivalence assessment worksheet

Under the definition-bearing model, the assessment is worked in the following order, and the outcome is recorded at each step:

a) **Intension comparison**: state the characteristics of the source concept and of the candidate target concept. Shared characteristics indicate a candidate `full` degree; overlapping but differing intensions indicate `partial`; no real correspondence indicates `none`.

b) **Concept-system check**: the candidate target concept occupies the same position in the target-language concept system as the source concept occupies in its system — the same superordinate concept and the same criterion of subdivision. A different position reduces the degree to at least `partial`, and the note states which position differs.

c) **Directionality check**: the correspondence holds in reverse only if the target term denotes the same extension when read back into the source language. A target term that means more, or less, than the source term in reverse gives the degree `directional`, and the note states the working direction.

d) **Conversion check**: a romanized or transliterated form names the ISO 24229 conversion system that produced or governs it; two romanizations of one term are two distinct localizations.

e) **Note drafting**: the equivalence note is one to three sentences that state what differs and for whom. A note that would not change the behaviour of a translator is too vague to record.

EXAMPLE The following assessment record shows the outcome fields:

```
concept: privacy (eng)
candidate: vie privée (fra)
degree: partial
note: >-
  French "vie privée" covers the legal-institutional aspects of
  privacy more strongly than the English concept, which also spans
  personal boundaries; drafting in French legal texts should prefer
  the narrower legal term where the context is regulatory.
```

Figure 5

8.7 Cadence

Two standing triggers keep localizations current:

© EFS 2026 – All rights reserved

— a change to the authoritative definition triggers review of the localizations whose degree is `partial`, `none` or `directional` under the definition-bearing model, and of the designations of all localizations under the term-only model, because a changed intension can leave a designation unable to carry the concept; and

— the periodic register review re-verifies a sample of `full`-degree localizations, because degrees drift as languages drift.

9 Relation management practice

Guidance for EFS 2513, clause 9, with EFS 2510.

9.1 Choosing a relation type

A decision procedure that settles most cases in four questions:

a) **Is this a structural fact** (kind-of, part-of, instance-of)? → hierarchical family; state the coordination method (generic, partitive, instantial) — the unqualified `broader`/`narrower` pair is for interchange with SKOS only.

b) **Is this a jointly-significant group** (a rake)?→ n-ary relation per EFS 2510, clause 7; use the binary-versus-n-ary decision rule.

c) **Is this a lifecycle or register fact** (superseded, replaced, retired, version)?→ lifecycle/versioning family; link register items, and keep concept-level and register-level relations distinct.

d) **Otherwise**: associative family; find the ISO 704 kind (contiguity, sequential, activity, instrumental, interactional, opposition) and use its directed pair. `related_concept` (untyped) is the last resort, not the default.

For cross-organizational links, use the mapping family (`exact_match` … `related_match`) and name the other vocabulary in the relation record; `equivalent` is never asserted across vocabularies that the register does not control.

9.2 Implementing the constraints

EFS 2510, clause 9 specifies mechanical constraints (ac1–ac6, lc1–lc3, na1–na5). In practice they become the register's validation suite, run on every publication run (clause 7) and ideally on every pull request. Implementation notes per family:

— **Acyclic hierarchical check (ac3)**: build the hierarchy graph from all hierarchical relation types and run a cycle detection from each node; report the full chain, not just "cycle found", so the author can see the loop.

— **Inverse consistency**: either materialize inverses on write and check them, or derive them at read; check that any materialized inverse uses exactly the inverse type from the vocabulary — a wrong inverse type (for example `related_concept` standing in for `deprecates`/`deprecated_by`) is a semantic bug, not a style issue.

— **Criterion presence (ac4, na)**: every n-ary relation file carries its criterion; a missing criterion fails validation rather than being inferred from prose.

— **Completeness sanity (na4)**: a `complete` rake whose members are all `optional` is rejected, with a pointer to the rule.

— **Register duality (lc3)**: where a register item and its concept record both exist, `replaces`/`replaced_by` links items and `supersedes`/`superseded_by` links concepts; the validator checks both legs exist.

9.3 Review practice for relations

Two habits catch most defects early:

© EFS 2026 – All rights reserved

— **Read the relation aloud** as a sentence: “the current concept <type> the related concept”. If the sentence is false or awkward, the direction or the type is wrong.

— **Check the rake before the edge**: when a proposal touches a concept that participates in an n-ary relation, re-validate the rake (members, completeness, criterion) — the most common breakage is an edit that made a `complete` rake incomplete without saying so.

9.4 Cross-vocabulary relations

Keep a small table of the vocabularies the register maps to, with their identifiers and the mapping types used. Every mapping relation record names the external vocabulary; a mapping without the vocabulary name is unverifiable and fails review.

9.5 Relation families in records — worked patterns

How each family looks when it is actually recorded. These are the patterns reviewers should see; anything that does not match a pattern below earns a question in review.

9.5.1 Hierarchical (generic)

```
related:
- type: broader_generic
  content: { eng: "kind of signal" }
  ref: { source: ES, id: signal }
```

Figure 6

The coordination method is stated in the type. The unqualified `broader`/`narrower` appears only in SKOS projections, never in source records (EFS 2510, 6.2 NOTE).

9.5.2 Hierarchical (partitive, binary)

```
related:
- type: has_part
  content: { eng: "has a locking device" }
  ref: { source: ES, id: locking_device }
```

Figure 7

Binary here because the assertion is a single part-of fact with no completeness claim.

9.5.3 Partitive as an n-ary relation file

When the parts jointly constitute the whole, the parts leave the concept record entirely and become a relation file:

```
# relations/railway-switch/construction.yaml
$id: "railway-switch/construction"
type: partitive_relation
comprehensive:
  ref: { source: ES, id: railway_switch }
members:
- ref: { source: ES, id: switch_blade }
  presence: required
  count: exactly_one
  delimiting: true
- ref: { source: ES, id: switch_motor }
  presence: optional
  count: exactly_one
- ref: { source: ES, id: locking_device }
  presence: required
```

© EFS 2026 – All rights reserved

```
  count: at_least_one
completeness: complete
criterion: "construction elements of the switch"
```

Figure 8

Reviewing this file: two or more members; all coordinate under the stated criterion; the `requiredexactly_oneoptional` combination is valid; exactly one delimiting member; the `complete` claim is consistent with an `optional` member being present (the rake is about the construction set, and the member presence describes instances, not the rake).

9.5.4 Sequential as an n-ary relation file

```
# relations/release-lifecycle/stages.yaml
$id: "release-lifecycle/stages"
type: sequential_relation
comprehensive:
  ref: { source: ES, id: release_lifecycle }
members:
- ref: { source: ES, id: proposed }
- ref: { source: ES, id: drafted }
- ref: { source: ES, id: reviewed }
- ref: { source: ES, id: published }
completeness: complete
criterion: "process step order"
```

Figure 9

Member order is significant here; validators check the count (two or more) and reviewers check that the order stated is the order that holds.

9.5.5 Lifecycle pair

```
related:
- type: supersedes
  content: { eng: "effective from edition 2" }
  ref: { source: ES, id: old_concept }
```

Figure 10

And on `old_concept`, the materialized inverse:

```
related:
- type: superseded_by
  ref: { source: ES, id: new_concept }
```

Figure 11

Both legs recorded; `replaces`/`replaced_by` is reserved for the register items and does not appear on the concept records here.

9.5.6 Mapping across vocabularies

```
related:
- type: exact_match
  content: { eng: "same item in the ISO 24229 register" }
  ref: { source: ISO, id: "urn:iso:24229:system-code:WebKit" }
```

Figure 12

© EFS 2026 – All rights reserved

The external vocabulary is named in the reference source field — a mapping without it cannot be audited (EFS 2513, 9.3).

9.6 Mapping vocabulary table

The cross-vocabulary table of Clause 9, worked for a register that maps to two external bodies:

Table 13

External vocabulary	Identifier form	Types used
ISO 24229 register	`urn:iso:24229:<itemClass>:<localId>`	`exact_match` (same item), `close_match` (functional equivalent)
A thesaurus partner	their local identifiers, prefixed in the record's ref source field	`broad_match` / `narrow_match` (their hierarchy differs), `related_match` (thematic overlap)

Rules the table enforces:

— one row per external vocabulary — a mapping record's ref source must appear here, or the mapping fails review; and

— the types column records what was **agreed** with the partner, not what is technically expressible: an `exact_match` claimed where the partner only agreed to a `close_match` is an inter-organizational incident, not a modelling choice.

10 Tooling, automation and repository practice

This clause gives guidance for the requirements of EFS 2513, clause 10.

10.1 The reference toolchain

The toolchain this guide assumes, and the one ES registers actually run:

— **Source of truth**: a git repository holding the dataset (concept records per EFS 2516, clause 4; register.yaml; relation files per EFS 2510, 7.8);

— **Model and validation**: the glossarist concept model — its schemas are the reference for the record format, and its validators cover the structural constraints;

— **Publication**: the concept-browser toolchain — builds the browsable site and the bulk exports (CSV, TTL, JSON-LD) from the dataset;

— **Process**: the repository's issue tracker for proposals and ballots; pull requests for record changes; CI for validation.

Nothing in EFS 2513 requires these specific tools; EFS 2516 conformance is what matters. But a new register saves weeks by starting from this stack, and the worked example (Annex D) is expressed in it.

10.2 Repository layout

A layout that has proven workable:

```
register-repo/
├── datasets/
│   └── <dataset-id>/
│       ├── register.yaml
│       ├── about-eng.md
│       ├── concepts/*.yaml          # concept records (EFS 2516, 4)
│       └── relations/               # n-ary relation files (EFS 2510, 7.8)
│           └── <comprehensive-id>/<criterion-slug>.yaml
├── site-config.yml                  # publication configuration
├── public/data/<dataset-id>/.gitkeep
├── scripts/generate.py              # generators, kept in-repo
```

© EFS 2026 – All rights reserved

```
└── .github/workflows/build.yml      # validation + build on push/PR
```

Figure 13

Keep generators in the repository: a dataset that can only be produced by an unrecorded process is not reproducible.

10.3 Validation automation

Run validation at three levels:

— **On every pull request**: record format validation, identifier scheme check, relation constraint suite (clause 9), YAML quoting lint (EFS 2516, 4.4). A red CI marks the proposal as not reviewable.

— **On merge**: a full publication run (validate → build → check → package) into a staging output, so a regression is caught before the next scheduled release.

— **On release**: the publication run of record, producing the published site and the exports for the edition.

The CI workflow for a register repository is small; the worked example (Annex D) shows one end to end.

10.4 Branching and review practice

— Proposals are pull requests: one proposal, one pull request, containing exactly the records the proposal touches.

— Reviewers review the pull request diff; the checklists of clause 7 are applied to the diff, not to the whole register.

— The register manager merges on control-body decision; merge commits reference the decision record identifier, which closes the loop from proposal to decision to published record.

— Direct pushes to the default branch are disabled for everyone, including the register manager (EFS 2513, 7.4.5: even editorial corrections are recorded).

10.5 Support records

Keep these alongside the dataset (EFS 2513, 10.4):

— the governance charter and charter history (clause 5);

— the criteria register and identifier scheme rule (clause 6);

— the publication log (one line per publication run: date, run result, edition);

— the decision records (Annex B); and

— the self-assessment records (clause 11).

10.6 CI workflow — a complete example

The register repository's build workflow, end to end. It validates and builds on every push and pull request; the main site pipeline performs the publication run of record.

```yaml
name: build

on:
  push:
    branches: [main]
  pull_request:
  workflow_dispatch:

permissions:
  contents: read
```

© EFS 2026 – All rights reserved

```yaml
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: 22
      - name: Build concept-browser site
        run: npx -y @glossarist/concept-browser@0.7.139 build
```

Figure 14

Notes on the shape:

— the workflow is **build-only**: publication of the site happens in the organization's main site pipeline (which mounts the register build), so the register repo cannot publish independently of a reviewed change;

— the toolchain version is pinned, because an unrequested toolchain upgrade is itself a change to published output and deserves a decision record (clause 12); and

— when validation rules grow (clause 9), they slot in as a step before the build — failing validation fails the pull request, which is the intake gate of clause 7.

10.7 Validator invocation patterns

The constraint suite of EFS 2510, clause 9 is invoked with dataset paths; representative invocations for the three families:

```
# structural: acyclic hierarchical relations, dangling references
validate-register --dataset datasets/<id> --rules ac1,ac2,ac3,ac4

# n-ary: member counts, presence/count consistency, completeness sanity
validate-register --dataset datasets/<id> --rules na1,na2,na3,na4,na5 \
                  --relations datasets/<id>/relations/

# lifecycle: duality between register items and concept records
validate-register --dataset datasets/<id> --rules lc1,lc2,lc3
```

Figure 15

The invocation set is recorded in the repository's CI configuration and in the publication log entry of each run; a constraint suite that is not executed at a known point is not a suite.

10.8 Choosing what to automate first

Registers moving to automation report the same ordering of value:

a) identifier scheme check and YAML lint (catches the most failures per effort);

b) relation constraint suite (ac3, na rules); then

c) localization field completeness (script + ISO 15924/24229 where applicable); and

d) export cross-checks (row counts per format).

11 Evaluation in practice

This clause gives guidance for the requirements of EFS 2513, clause 11.

© EFS 2026 – All rights reserved

11.1 Monitoring

Measure a small number of things that drive action:

Table 14

Metric	How to collect	Action trigger
Intake time (proposal opened → review started)	Issue tracker timestamps	Median exceeds the charter's stated window
Decision time (review complete → control-body decision)	Decision record dates	Median grows across two quarters
Validation error rate at publication runs	CI logs	Any non-zero trend
Consumer-reported issues	Issue tracker label	Any issue touching identifiers or references

A dashboard is built only for numbers that have action triggers: a metric without a trigger produces no improvement and is omitted from monitoring.

11.2 The periodic review

Run the review per concept system at a planned interval (annually is a workable default; EFS 2513, 11.2). The review is a working session with a written output, not a ceremony:

a) **Run the full validation suite** against the live dataset; log findings with severity.

b) **Sample records for depth checks**: for each criterion of subdivision, confirm it is still stated on the records it generated (EFS 2513, 11.2); sample `full`-degree localizations and re-verify the degree (clause 8).

c) **Check lifecycle duality**: retired items resolve; their successor information is present; register-item and concept-level lifecycle relations agree.

d) **Check the SKOS projection** against the full dataset: no concepts lost, no semantics altered (EFS 2516, 6.2).

e) **Review the context records** of clause 4: purpose sentence, audience worksheet, boundaries — still true?

The output is a review record: findings, disposition of each, and the improvement actions raised into the process of clause 12.

11.3 Internal audit

The audit examines the **system**, not the concepts: are decision records being written? Were any changes published without decision? Do role assignments match practice? Is the appeals channel used and functioning?

A light, effective pattern: once per year, trace five published changes end-to-end — proposal → review → decision record → publication run → live record. Every step must be documented and dated. Any gap is a nonconformity to disposition (clause 12).

11.4 Self-assessment against EFS 2513

Annex C contains a clause-by-clause checklist: every line names an EFS 2513 requirement and asks the question an assessor would ask. Run it:

— before first claiming a requirements class (Annex E of this guide);

— at each internal audit; and

— before each renewal of a public conformance claim (EFS 2513, 11.4).

Record the completed checklist with its date and the assessor; the claim of conformity then cites it.

© EFS 2026 – All rights reserved

11.5 Sample review record

The periodic review's written output, in the shape this guide expects (clause 11, EFS 2513, 11.2):

```
review:
  register: <dataset-id>
  date: <YYYY-MM-DD>
  interval: annual
  validation:
    suite: full
    findings: 0 errors, 2 warnings (both pre-existing, dispositioned)
  depth_checks:
    criteria_register:
      sampled: 6 criteria
      result: all stated on records they generated
    localizations:
      sampled: 5 full-degree
      reverified: 4 full, 1 reduced to partial (note added, DR-2026-021)
  lifecycle_duality:
    retired_items_checked: 7
    all_resolve_with_successor_info: true
  skos_projection:
    concepts_in_full_dataset: 109
    concepts_in_projection: 109
    semantics_altered: none
  context_records:
    reviewed: true
    changes_needed: none
  findings:
    - finding: "'partial' localization reduced on re-verification"
      disposition: corrected (note added); no sweep needed
  improvements_raised: []
  next_review: <date + interval>
```

Figure 16

11.6 Audit trace walkthrough

The once-a-year trace of five published changes, spelled out for one change so the pattern is unambiguous. The auditor picks a published record and follows it backwards:

a) **Live record**: the record on the published site says definition version B, accepted 2026-11-11.

b) **Publication run**: the publication log has an entry on that date; its trigger cites `merge of PR #12 (decision DR-2026-014).`

c) **Pull request**: PR #12 touches exactly the record and the files the proposal described; no other records changed.

d) **Proposal**: issue #87 carries a complete proposal form (Annex A).

e) **Review**: the review comment applies the clause 7 checklists with one required change and a recommendation.

f) **Decision**: DR-2026-014 records the question, the vote (3 yes), and the rationale, within the charter's window.

g) **Localization trigger**: the record has localizations; the change touched the intension; review tasks were opened per clause 8 (or the audit records the gap).

Every arrow resolved with a dated record: the change passes. Any arrow that dead-ends — an undocumented merge, a missing decision record, a localization that should have been re-reviewed and was not — is a nonconformity with a clause number attached.

© EFS 2026 – All rights reserved

12 Improvement practice

This clause gives guidance for the requirements of EFS 2513, clause 12.

12.1 Nonconformity handling

When validation, review, audit, or a consumer surfaces a nonconformity, work it in five recorded steps:

a) **Record**: file it with the finding, its source (validation rule, audit line, consumer report), and severity.

b) **Correct**: fix the affected records; for published records, the fix is itself a recorded change through the procedures of clause 7 (consumer reports on identifiers and references are always treated as urgent).

c) **Trace the cause**: records, process, or governance? A recurring YAML quoting defect is a process defect (add the lint); a decision repeatedly reversed is a governance defect (clarify the charter).

d) **Sweep**: check whether the same defect exists in sibling records — validation sweeps are cheap; manual sweeps target the family of the defect.

e) **Verify**: at the next review, confirm the nonconformity has not returned.

12.2 Improvement decisions

Improvements are control-body decisions like any other, recorded with Annex B's form. The standing inputs:

— monitoring metrics with triggered actions (clause 11);

— review and audit findings;

— consumer requirement changes (clause 4);

— vocabulary and schema evolution — new versions of EFS 2510 and EFS 2516, and evolution of registered vocabularies the register uses (such as new ISO 24229 conversion systems entering the localization fields); and

— toolchain upgrades, which are recorded decisions too: pin versions, record why, and re-run the full validation before and after.

12.3 The improvement log

Keep a one-page improvement log: date, source finding, decision reference, effect, verification date. At the next review, the log is the checklist: each improvement is verified to have had its effect, and unverified improvements are carried forward explicitly rather than silently dropped.

12.4 Continual improvement, realistically

A register improves when three loops run at once: the **record loop** (defects found → corrected → swept), the **process loop** (causes → charter, checklists, automation), and the **content loop** (consumers → scope → criteria). This guide's structure follows those loops on purpose; a register that only ever exercises the record loop repeats the same class of corrections without reducing their cause.

12.5 Improvement log — worked entries

The improvement log of clause 12, filled with two representative entries:

```
log:
- date: 2026-11-11
  source: build failure in publication run (YAML colon in definition)
  decision: DR-2026-017
  action: >-
    Add the scalar quoting rule to EFS 2516 (clause 4.4); add quoting
    lint expectation to the review checklist of EFG 2519, clause 7.
  effect: >-
```

© EFS 2026 – All rights reserved

```
    Run fails at lint with file and line before any manual review time
    is spent.
  verified: 2026-11-12 (next run green, lint caught a recurrence)

- date: 2026-11-01
  source: registry could not categorize the register's documents
  decision: DR-2026-012
  action: >-
    Introduce the `guide` doctype end to end: vocabulary spec, record
    front matter, registry category mapping, and site publication.
  effect: >-
    Guides publish in their own registry category; directives and
    standards unaffected.
  verified: 2026-11-11 (deploy verified live)
```

Figure 17

The verified field is the discipline: an improvement without a verification date is a promise, and the review of clause 11 treats it as an open item.

© EFS 2026 – All rights reserved

Annex A
(normative)

Concept proposal form

This form is the intake record for every operation on the register (create, revise, split, merge, instantiate, retire, localize). It is designed to be pasted into an issue in the register repository; fields marked (opt) may be left empty with a reason.

A.1 Operation header

```
operation: create            # create | revise | split | merge | instantiate | retire |
localize
concept_id: (opt — for non-create operations, the affected identifier)
dataset: <dataset-id>
submitter: <name / organization>
date: <YYYY-MM-DD>
```

Figure A.1

A.2 New concept (operation: create)

```
proposed_identifier: <per the identifier scheme>
urn: <if issued at proposal time; otherwise left for the register manager>

designations:
- designation: <term>
  type: expression            # expression | abbreviation | symbol | formula | graphical
  normative_status: preferred # preferred | admitted | deprecated
- designation: (opt) <abbreviation or other form>
  type: abbreviation
  normative_status: admitted

superordinate_concept: <identifier of the genus concept>
criterion_of_subdivision: <wording, checked against the criteria register>

definition: >-
  <The definition. Genus first; no articles; no shall/should/may;
  states the delimiting characteristics.>

notes: (opt)

examples: (opt)

sources:
- ref: <document or publication>
  clause: (opt)
  link: (opt)

relations:
- type: <relation type per EFS 2510, clause 6>
  related_concept: <identifier>
  content: (opt — explanation of this specific assertion)
```

Figure A.2

© EFS 2026 – All rights reserved

A.3 Revision (operation: revise)

```
reason_for_change: >-
  <What is wrong with the current record and why the change is needed.>
fields_changed:
- field: definition            # designation | definition | relations | sources | status
  current: >-
    <verbatim current content>
  proposed: >-
    <verbatim proposed content>
impact_on_consumers: >-
  <What changes for systems citing this concept. Intension change?
  Localization review needed? SKOS projection affected?>
```

Figure A.3

A.4 Split (operation: split)

```
predecessor: <identifier>
successors:
- proposed_identifier: <id-1>
  designation: <term-1>
  definition_summary: >-
    <One sentence: which part of the predecessor's intension it takes.>
- proposed_identifier: <id-2>
  designation: <term-2>
  definition_summary: >-
    <One sentence.>
relation_migration:
- old_relation: { type: <type>, related: <id> }
  goes_to: <id-1>              # which successor inherits it, and why
```

Figure A.4

A.5 Merge (operation: merge)

```
predecessors:
- <identifier-1>
- <identifier-2>
successor:
  proposed_identifier: <id>
  designation: <term>
  definition: >-
    <The merged intension.>
predecessor_designation_mapping:
- predecessor: <identifier-1>
  treatment: preferred | admitted | deprecated
```

Figure A.5

A.6 Localization (operation: localize)

```
source_concept: <identifier>
language: <ISO 639-3>          # e.g. fra, cmn
script: (opt) <ISO 15924>      # mandatory for multiscript or romanized forms
conversion_system: (opt) <ISO 24229 identifier>   # mandatory for converted forms
designations:
- designation: <localized term>
  type: expression
  normative_status: preferred
definition: >-
  <Localized definition.>
```

© EFS 2026 – All rights reserved

```
equivalence_degree: full       # full | partial | none | directional
equivalence_note: (mandatory unless full) >-
  <What differs and for whom; see EFG 2519, 8.3.>
```

Figure A.6

A.7 Retirement (operation: retire)

```
concept: <identifier>
successor: (opt) <identifier>
ground: >-
  <Why the concept should be retired: obsolescence, replacement,
  error in intension, scope change.>
```

Figure A.7

A.8 Submission notes

a) One operation per form. Two operations touching the same concept in one submission is two proposals.

b) Attach or link the concept record draft in the proposal, so review comments can point at lines.

c) If a proposal affects an n-ary relation (adds or removes a rake member, or flips completeness), say so explicitly at the top of the form: this changes the review path (EFG 2519, 7.3).

© EFS 2026 – All rights reserved

Annex B
(normative)

Decision record template

One decision record per control-body decision. Decision records are numbered, immutable once closed, and cross-referenced from the pull request that implements the decision.

B.1 Template

```
id: DR-2026-001                       # sequential, never reused
date: <YYYY-MM-DD>
decision_channel: written-ballot      # written-ballot | meeting | delegated-review
window: {opened: <date>, closed: <date>}   # for ballots
proposal: <link to the proposal issue / pull request>

question: >-
  <The exact question put to the control body, in one sentence.>

outcome: accepted                     # accepted | returned | rejected | withdrawn

rationale: >-
  <Why this outcome, on the technical and procedural merits. This is
  the field that future volunteers will read; write it for them.>

votes:
- member: <name>
  vote: yes                           # yes | no | abstain
  comment: (opt; mandatory for "no")

delegation: (opt, for delegated-review decisions) >-
  <The charter clause under which review consensus was delegated.>

effects:
  records:
    - <identifier(s) created or changed>
  identifiers_issued: <list or "none">
  editions_affected: <list or "next scheduled run">
  related_decisions: (opt) [<DR-YYYY-NNN>]

appeal:                                # filled if appealed
  status: none                         # none | acknowledged | heard | upheld | rejected
  record: (opt) <link to the appeal record>
```

Figure B.1

B.2 Worked example

```
id: DR-2026-014
date: 2026-11-12
decision_channel: written-ballot
window: {opened: 2026-11-02, closed: 2026-11-09}
proposal: https://github.com/example/register/issues/87

question: >-
  Should the register accept "ranks_above"/"ranks_below" as relation
  types in the adopted vocabulary, per proposal #87?
```

© EFS 2026 – All rights reserved

```
outcome: accepted

rationale: >-
  The pair completes the contiguity family per ISO 704:2022 Table 1 and
  is already used informally by two consuming systems. The review found
  the definitions faithful to the source. The identifier "outranks"
  was preferred over "is_senior_to" for brevity; the designation
  improvement path remains open since identifiers are permanent.

votes:
- member: A. Reviewer
  vote: yes
- member: B. Reviewer
  vote: yes
- member: C. Reviewer
  vote: yes

effects:
  records:
    - ranks_above
    - ranks_below
  identifiers_issued: ["urn:example:relation-type:ranks_above",
                       "urn:example:relation-type:ranks_below"]
  editions_affected: [next scheduled run]
  related_decisions: [DR-2026-009]

appeal:
  status: none
```

Figure B.2

B.3 Practice notes

— Write the rationale last, and write it for a reader who was not in the discussion. The vote tally is the easiest field and the least useful; the rationale is the opposite.

— When a decision delegates acceptance to review consensus, the record must cite the charter clause that allows the delegation — an unanchored delegation is a nonconformity at audit (EFG 2519, 11.3).

— Superseding a decision is a new record with `related_decisions: [<contested DR>]` and `outcome` describing the new state; the contested record is annotated in its `appeal.record` field, never edited.

© EFS 2026 – All rights reserved

Annex C
(normative)

Self-assessment checklist against EFS 2513

Run this checklist per register. For each line: mark Conforming ©, Partial (P), or Nonconforming (N); for P and N, record the finding and the action. The completed checklist, dated and signed, is the record that supports a conformance claim (EFS 2513, 11.4).

C.1 Clause 4 — Organization and context

Table C.1

#	Question	Ref (EFS 2513)
4.1-1	Are the external and internal issues that affect the register's concepts documented, and current?	4.1
4.2-1	Does a concept system scope statement exist (domains, purposes, boundaries)?	4.2
4.2-2	Is it reviewed on context change, with review dates recorded?	4.2
4.3-1	Are consumer classes identified with their stability, citability and format requirements?	4.3
4.3-2	Are changes assessed against consumer requirements before being applied?	4.3

C.2 Clause 5 — Governance

Table C.2

#	Question	Ref
5.1-1	Is management commitment documented (policy established, resources available, bodies appointed)?	5.1
5.2-1	Does a concept management policy exist, and is it communicated to participants?	5.2
5.3-1	Does the control body have a written charter with composition, quorum, voting, records, and appeals?	5.3
5.3-2	Are all control-body decisions recorded as decision records?	5.3
5.3-3	Is there an appeals process, and does it exclude original decision-makers from hearing appeals?	5.3
5.5-1	Are roles assigned and recorded (register manager, concept owners, submitters, reviewers)?	5.5
5.5-2	Does the register manager avoid being sole reviewer of their own proposals?	5.5
5.6-1	Are participation rules (membership classes, external parties, confidentiality) documented?	5.6

C.3 Clause 6 — Planning

Table C.3

#	Question	Ref
6.1-1	Are breadth and depth of each concept system planned and recorded?	6.1
6.1-2	Does every subdivision record its criterion of subdivision?	6.1
6.2-1	Is there a written procedure for extending the concept system (criterion recorded before subordinate concepts)?	6.2

© EFS 2026 – All rights reserved

(continued)

#	Question	Ref
6.3-1	Is the lifecycle model defined (states, transitions), aligned with ISO 19135 statuses where a register is operated?	6.3
6.3-2	Is there a versioning policy: new version vs new concept, reference stability across versions?	6.3
6.3-3	Is a review interval planned per concept system?	6.3
6.4-1	Is the identifier scheme planned, stable, and independent of terms, states and locations?	6.4

C.4 Clause 7 — Operations

Table C.4

#	Question	Ref
7.1-1	Do proposals state designations, definition, placement (superordinate + criterion), relations, and sources?	7.1.1
7.1.2-1	Are definitions drafted per ISO 704 / ISO 10241-1 (intension, no requirements, no circularity)?	7.1.2
7.1.2-2	For extensional subdivisions, is member exhaustiveness verified before publication?	7.1.2
7.1.3-1	Does every published concept pass review (placement, uniqueness, relations, entries, sources)?	7.1.3
7.2-1	Does each published concept record carry identifier, designations, definition with sources, relations, status, and history?	7.2
7.2-2	Are records machine-readable per EFS 2516?	7.2
7.5.1-1	Do changes to published concepts go through proposal and review, with editorial corrections recorded by the register manager?	7.5.1
7.5.2-1	Are extending, splitting, merging, and instantiating operations decided by the control body and recorded per the record patterns?	7.5.2
7.5.3-1	Is retirement a status change with successor linkage, never a deletion?	7.5.3
7.5.4-1	Do identifiers remain resolvable across every operation?	7.5.4
7.6-1	Where a register is operated: are ISO 19135 procedures and statuses in use, and are register events mirrored as concept-record relations?	7.6

C.5 Clause 8 — Localization

Table C.5

#	Question	Ref
8.2-1	Does every localization record carry ISO 639-3 language, script where applicable, and ISO 24229 conversion system for converted forms?	8.2
8.3-1	Is an equivalence degree recorded per localization, with a note for anything other than `full`?	8.3
8.3-2	Are non-`full` acceptances recorded decisions with reasons?	8.3
8.4-1	Does a workflow exist that triggers localization review on source-concept changes?	8.4

C.6 Clause 9 — Relations

Table C.6

#	Question	Ref
9.1-1	Is an adopted relation vocabulary in force, and are unknown identifiers treated as errors?	9.1

© EFS 2026 – All rights reserved

(continued)

#	Question	Ref
9.2-1	Are the EFS 2510 structural rules validated (acyclicity, criteria, coordination, n-ary first-class, lifecycle duality)?	9.2
9.3-1	Are cross-vocabulary links recorded with mapping types and the external vocabulary named?	9.3
9.4-1	Are relations included in the periodic review?	9.4

C.7 Clause 10 — Support

Table C.7

#	Question	Ref
10.1-1	Are resources (roles, tooling, publication infrastructure) provided?	10.1
10.2-1	Are participants competent (ISO 704 analysis, drafting, vocabulary) and aware of policy and roles?	10.2
10.3-1	Are governance records, criteria records, identifier scheme, records, and histories controlled?	10.3
10.4-1	Are machine-readable publications faithful (identifiers, relations, n-ary metadata, localizations, statuses)?	10.4

C.8 Clause 11 — Evaluation

Table C.8

#	Question	Ref
11.1-1	Is monitoring in place (intake, decision times, error rates, consumer reports)?	11.1
11.2-1	Is the periodic review performed at planned intervals, with the EFS 2510 checks and findings recorded?	11.2
11.3-1	Is an internal audit performed, with criteria, findings, and dispositions recorded?	11.3
11.4-1	Do conformance claims state the class, scope, and assessment basis?	11.4

C.9 Clause 12 — Improvement

Table C.9

#	Question	Ref
12.1-1	Are nonconformities corrected, causes addressed, and sweeps performed?	12.1
12.2-1	Is an improvement log kept, with effects verified at review?	12.2

C.10 Scoring guidance

— **C on all lines of a class** → the class may be claimed (Annex E).

— **P on any line** → claim only with a recorded remediation plan and date; auditors treat stale P entries as N.

— **N on any line of a claimed class** → the claim is withdrawn or narrowed until the finding is dispositioned; publishing a claim with a known N is the fastest way to lose the trust the register exists to earn.

C.11 Assessment record template

The completed assessment is itself a controlled record:

```
assessment:
  register: <dataset-id>
```

© EFS 2026 – All rights reserved

```
date: <YYYY-MM-DD>
assessor: <name, role>
classes_assessed: [governed]      # per the adoption worksheet, Annex E
lines_total: 46
conforming: 44
partial: 2
nonconforming: 0
partial_findings:
  - line: "8.2-1"
    finding: >-
      Conversion system field recorded for romanizations, but not
      yet validated automatically at publication.
    action: add localization field validation to CI
    target_date: <date>
nonconforming_findings: []
claim:
  statement: >-
    This register claims conformity to EFS 2513 requirements class
    "governed" on the basis of this assessment.
  valid_until: <next review date>
```

Figure C.1

Keep every completed assessment; the sequence of assessments is the register's conformity history, and improvement decisions (clause 12) cite it.

© EFS 2026 – All rights reserved

Annex D
(normative)

Worked example — the ES Relation Vocabulary

This annex walks one real register end to end: the ES Relation Vocabulary, the controlled vocabulary of concept relation types defined in EFS 2510, clause 6, published by ES at the Enosema concept register. Everything below is taken from the register as it operates.

D.1 Purpose and audience (clause 4)

Purpose sentence: "The register provides the authoritative set of concept relation types used in ES concept systems and exchanged between terminology management systems."

Register type: vocabulary register (clause 4), with a standards-derivative aspect: the items are normative content of EFS 2510, clause 6, registered as items in their own right.

Audiences:

Table D.1

Audience	Decision supported	Format needed
Terminology management systems	Validate relation type identifiers in exchanged records	TTL, JSON-LD, CSV bulk exports
Standards authors	Cite relation types when drafting ES documents	Browsable site, identifiers
SKOS consumers	Project vocabularies using the mapping types	TTL

Boundaries: the register holds the relation type vocabulary; it does not hold concept datasets, nor designation-level relations (those live in the datasets that use the vocabulary). Adoption of concepts from other bodies is out of scope; new relation types arrive as proposals through EFD 101 processes.

D.2 Governance (clause 5)

Control body: the Terminology Technical Committee acting under a charter whose essentials are:

> Proposals are submitted as issues in the register repository. The Technical Committee decides by consensus recorded in the proposal thread; where consensus is unclear, a written ballot of seven days is opened and passes by two-thirds of votes cast excluding abstentions. Decision records are written by the register manager within five working days of each decision and are numbered DR-YYYY-NNN. Appeals are addressed to the Board of Directors per EFD 101, clause 7.

Roles: register manager (runs publication runs, writes decision records, maintains register metadata); reviewers (Terminology TC members assigned per proposal); submitters (TC members; external parties per EFD 101 external-member rules).

Why governance matters here, concretely: relation type identifiers are consumed by validating systems. A rename would break every consumer. The control body therefore applies the never-rename rule as a standing decision — recorded as DR-2026-002 — which is exactly the kind of decision a register exists to make and keep visible.

© EFS 2026 – All rights reserved

D.3 Planning (clause 6)

Identifier rule:

Table D.2

Item	Form
Dataset URN	`urn:enosema:relation-vocabulary:<year>`
Relation type URN	`urn:enosema:relation-type:<identifier>`
Relation type identifier	snake_case verb or verb-pair form per EFS 2510, clause 6

Editions: the dataset id carries the year (`es-relation-vocabulary-2026`); a future revision with substantive changes publishes a new edition id; item identifiers never change.

Criteria: the register subdivides its types by **family** (hierarchical, equivalence, associative kinds, lexical, lifecycle, register, versioning), recorded in the generator's canonical table and visible as SKOS collections.

D.4 The dataset (clause 7)

Repository layout:

```
concept-register/
├── datasets/
│   └── es-relation-vocabulary-2026/
│       ├── register.yaml
│       ├── about-eng.md
│       └── concepts/            # 109 concept records, one file each
│           ├── broader_generic.yaml
│           ├── supersedes.yaml
│           └── …
├── site-config.yml
├── skos/relation-type.ttl       # SKOS projection (generated)
├── scripts/generate.py          # canonical table → records + SKOS
└── public/data/es-relation-vocabulary-2026/.gitkeep
```

Figure D.1

register.yaml (verbatim):

```yaml
---
schema_type: glossarist
schema_version: '3'
id: es-relation-vocabulary-2026
ref: ES Relation Vocabulary (2026)
year: 2026
urn: urn:enosema:relation-vocabulary:2026
status: current
owner: Enosema Foundation
source_repo: https://github.com/enosema/concept-register
tags:
- enosema
- relation-types
- concept-relations
- EFS-2510
- SKOS
languages:
- eng
language_order:
- eng
```

Figure D.2

One concept record (`concepts/broader_generic.yaml`, verbatim):

© EFS 2026 – All rights reserved

```yaml
---
id: broader_generic
data:
  identifier: broader_generic
  uri: urn:enosema:relation-type:broader_generic
status: valid
date_accepted: '2026-11-11T00:00:00+00:00'
sources:
- type: authoritative
  origin:
    ref:
      source: EFS 2510:2026
      clause: 6
    link: https://www.enosema.org/standards/docs/efs-2510/2026/
related:
- type: related_concept
  content:
    eng: "inverse relation type"
  ref:
    source: ES
    id: narrower_generic
---
id: broader_generic-eng
termid: broader_generic
data:
  language_code: eng
  terms:
  - type: expression
    designation: broader_generic
    normative_status: preferred
  definition:
  - content: "The current concept is a kind of the related concept (generic/specific)."
  notes:
  - content: "Inverse relation type: `narrower_generic`. Relation family: hierarchical-
generic. Source: ISO 25964-1 (BTG/NTG); EFS 2510:2026, 6.2."
status: valid
```

Figure D.3

Note the self-referential discipline: the register's own items use the register's vocabulary (`related_concept` for the inverse-of pointer, typed and directed), and every record cites its normative source.

Generation: all 109 records and the SKOS projection regenerate from one canonical table:

```
python3 scripts/generate.py
```

Figure D.4

The generator is in the repository on purpose (EFS 2513, 10.3): the dataset is reproducible from recorded inputs.

D.5 Publication (clause 7, runbook in practice)

site-config.yml (essentials):

```yaml
id: enosema-concept-register
domain: www.enosema.org
uri_base: https://www.enosema.org
base_path: /concept-register
title: Enosema Concept Register
datasets:
  - id: es-relation-vocabulary-2026
    title: ES Relation Vocabulary (2026)
    local_path: datasets/es-relation-vocabulary-2026
```

© EFS 2026 – All rights reserved

```
bulk_exports: [ttl, jsonld, csv]
```

Figure D.5

The publication run, as CI executes it:

a) **Validate and build**: `npx @glossarist/concept-browser build` — ingests the dataset, validates records, processes 109 concepts, builds the site.

b) **Check**: exports exist under `public/data/<dataset-id>/`; row counts match concept counts; the TTL parses.

c) **Publish**: the register build is mounted at https://www.enosema.org/concept-register/ by the main site pipeline, alongside the register's own build-check workflow in its repository.

What a failure looks like: a definition quoting a standard without YAML quoting (`ISO 704:2022`) breaks the parser — the run fails, the log names the file and line, nothing publishes, and the fix is itself a recorded change. This failure has occurred in practice; the quoting rule of EFS 2516, 4.4 exists because of it.

D.6 SKOS projection (EFS 2516, clause 6)

The projection keeps what SKOS can carry and annotates the rest:

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix eno: <urn:enosema:relation-type:> .

<urn:enosema:relation-vocabulary:2026> a skos:ConceptScheme ;
    dcterms:title "ES Relation Vocabulary"@en ;
    dcterms:creator "Enosema Foundation" ;
    skos:hasTopConcept eno:broader_generic, eno:broader_partitive,
        eno:equivalent, eno:deprecates, eno:has_version .

eno:broader_generic a skos:Concept ;
    skos:prefLabel "broader_generic"@en ;
    skos:definition "The current concept is a kind of the related
        concept (generic/specific)."@en ;
    skos:inScheme <urn:enosema:relation-vocabulary:2026> ;
    owl:inverseOf eno:narrower_generic ;
    dcterms:source "ISO 25964-1 (BTG/NTG); EFS 2510:2026, 6.2" .
```

Figure D.6

The scheme, families as collections, and per-type inverse statements are generated; the full records remain the record of authority (round-trip prohibition, EFS 2516, 6.3).

D.7 Improvement in practice (clause 12)

Two real improvement decisions from this register's short life, as decision records would record them:

— The registry initially could not categorize the items: the concept records carried a doctype the aggregation mapping did not know. The fix spanned the vocabulary spec, the record format, and the registry configuration — one improvement decision, three coordinated changes, one verification run.

— The quoting defect (clause 7 of this annex) produced a build failure that stopped a publication run; the sweep added the quoting rule to the interchange spec and the lint expectation to review checklists. Record loop and process loop, in one entry.

© EFS 2026 – All rights reserved

Annex E
(normative)

Requirements class adoption worksheet

EFS 2513, Annex A defines requirements classes: `basic`, `governed`, `multilingual`, `registered`, `full`. This worksheet turns the choice into a recorded decision, which the self-assessment (Annex C) then examines.

E.1 Step 1 — Answer the scoping questions

Table E.1

Question	Why it matters
Does the organization make decisions by a control body, with recorded decisions?	If no: start at `basic`; governance adds `governed`.
Does it publish concepts in more than one language?	If yes: `multilingual` (includes `governed`).
Does it operate or participate in a register under ISO 19135 / FERIN?	If yes: `registered` (includes `governed`).
Is the register public and machine-consumed?	Bulk exports and the publication profile (EFS 2516) are expected of `registered` and `full` adopters in practice.

E.2 Step 2 — Record the claim

```
organization: <name>
date: <YYYY-MM-DD>
class_claimed: governed            # basic | governed | multilingual | registered | full
scope:
  registers:
    - <register or dataset identifiers covered>
  concept_systems:
    - <concept system scopes covered>
exclusions:
  - <what is explicitly out of the claim>
assessment:
  basis: self-assessment            # or internal audit / third party
  checklist_record: <link to the completed Annex C checklist>
  next_review: <date>
```

Figure E.1

E.3 Step 3 — Gap plan

For every Partial (P) or Nonconforming (N) line of the self-assessment within the claimed class:

Table E.2

Checklist line	Finding	Action and owner	Target date

Rules of thumb, learned the hard way:

© EFS 2026 – All rights reserved

— A class claim with open N lines is not a claim; narrow the claim or fix the lines first.

— `basic` is honest and useful: most organizations fail `governed` first at the decision-records line (5.3-2), because “we decide things in chat” is not a decision record.

— Upgrading the class later is easy — the ladder has no cost except the lines themselves; downgrading in public is the expensive direction.

E.4 Step 4 — Re-assess

Set `next_review` to the periodic review of clause 11, and re-run Annex C against the claimed class each time. The claim renewal cites the new checklist record (EFS 2513, 11.4).

© EFS 2026 – All rights reserved

Bibliography

[1] ISO 9001, INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). _Quality management systems — Requirements_. Report no. 9001. Sixth edition. Geneva: International Organization for Standardization, ISO 9001. iso-reference ISO 9001(E). URN urn:iso:std:iso:9001. Available from: https://www.iso.org/standard/88464.html.

[2] ISO 9004, *Quality management — Quality of an organization — Guidance to achieve sustained success*.

[3] ISO 25964-1, *Information and documentation — Thesauri and interoperability with other vocabularies — Part 1: Thesauri for information retrieval*.

[4] ISO 30042, *Management of terminology resources — TermBase eXchange (TBX)*.

[5] Enosema concept register, *ES Relation Vocabulary, published register*, https://www.enosema.org/concept-register/

[6] FERIN, *Framework for Extensible Registration of INformation*, https://www.ferin.org/

[7] Glossarist concept model, *Glossarist concept model, dataset schema v3*, https://github.com/glossarist/concept-model

[8] concept-browser, *Concept-browser publication toolchain*, https://github.com/glossarist/concept-browser

© EFS 2026 – All rights reserved

enosema

Price based on 41 pages

© EFS 2026
All rights reserved

Making meaning meaningful