

Alaska Soil Data Bank (AKSDB) v2 Documentation

Nic Jelinski

2026-01-05

Table of contents

Home	3
What is AKSDB v2?	3
How to use this documentation	3
1) Read it end-to-end (recommended for contributors)	3
2) Jump in based on your task	4
3) Use it as a reference	4
Quick links	4
The AKSDB object model (bird's-eye view)	4
Public vs private (two-repo worldview)	5
Guarantees (what AKSDB tries to make true)	5
Where to go next	5
Citation and licensing	6
Project status	6
 I Concepts::concepts	 7
1 Concepts	8
1.1 Why these concepts exist	8
1.2 The core chain	8
1.3 In this section	9
1.4 Suggested next pages	9
 II Registry::registry	 10
2 Registry	11
2.1 What the registry is	11
2.2 How to use this section	11
2.3 Registry docs	11
2.4 Design notes	12
 3 packages::Packages	 13
3.1 Columns	14

III Vocabulary::vocab	20
4 Vocab	21
4.1 What belongs here	21
4.2 How vocab is used	21
4.3 Start here	21
5 Controlled vocabularies standard	22
5.1 What a controlled vocabulary CSV represents	22
5.2 Identifiers and namespaces	22
5.3 Standard column set	22
5.3.1 Governance fields	23
5.3.2 Labels and definitions	23
5.3.3 Relationships and mappings	23
5.4 Multi-value encoding convention	24
5.5 CSV quoting and commas	24
5.6 Regex and validation	24
5.7 CSVW metadata sidecars	25
5.7.1 What CSVW is for	25
5.7.2 How to interpret the CSVW file	25
5.7.3 How to use CSVW in this project	25
5.7.4 Canonical multi-value parsing rule	26
5.8 Documentation format for each vocabulary	26
6 eml_responsibility_roles	27
6.1 Controlled vocabulary file	27
6.2 Columns	27
6.3 References	29
7 datacite_contributor_roles	30
7.1 Controlled vocabulary file	30
7.2 Columns	30
7.3 References	32
8 credit_roles	33
8.1 Controlled vocabulary file	33
8.2 Columns	33
8.3 References	35
IV Glossary::glossary	36
9 Glossary	37
9.1 Purpose	37

9.2	In this section	37
9.3	Authoring rule of thumb	37
10	Glossary	38
10.1	Overview	38
10.1.1	Concept	38
10.1.2	Data	39
10.1.3	Data dictionary	40
10.1.4	Data element	41
10.1.5	Dictionary	41
10.1.6	Glossary	42
10.1.7	Metadata	42
10.1.8	Registry	43
10.1.9	Term	43
10.1.10	Value domain	44
10.1.11	Catalog	45
10.1.12	Semantic	45
10.1.13	Element	45
10.1.14	Data element	45
10.1.15	Dictionary	46
10.1.16	Catalog	46
10.1.17	Taxonomy	46
10.1.18	Term	47
10.2	References	50
V	Datasets::datasets	51
11	Datasets	52
11.1	What a dataset page is	52
11.2	What every dataset page should include	52
11.3	Dataset pages	52
VI	Products::products	53
12	Products	54
12.1	What a product is	54
12.2	What every product page should include	54
12.3	Start here	54

VII Mappings::mappings	55
13 Mappings	56
13.1 Why mapping docs matter	56
13.2 What a mapping doc should include	56
13.3 Typical mapping topics	56
VIII Standards::standards	57
14 Standards	58
14.1 Purpose	58
14.2 Pages	58
14.3 Practical output	58
IX Governance::governance	59
15 Governance	60
15.1 What governance covers	60
15.2 Pages	60
15.3 Governance principle (default)	60
X Examples::examples	61
16 Examples	62
16.1 What belongs here	62
16.2 Suggested starter examples	62

Home

i Note

This book is the documentation hub for **AKSDB v2**: a reproducible, citable, and governance-aware system for managing Alaska soil datasets and publishing harmonized products.

What is AKSDB v2?

AKSDB v2 is a **data + metadata + governance** system for assembling many heterogeneous soil datasets into a consistent, auditable set of **harmonized outputs**—while preserving provenance back to source datasets.

It is designed around four ideas:

- **Stable IDs** for long-lived references (machines + citations)
- **Human slugs** for readable URLs and docs navigation
- **Releases** to publish coherent snapshots of schema + vocab + mappings + products
- **Checksums/manifests** to guarantee integrity and reproducibility

If you remember one thing: *AKSDB is built so that you can always answer “what is this, where did it come from, and what changed?”*

How to use this documentation

Use this book in one of three ways:

1) Read it end-to-end (recommended for contributors)

Start with **Concepts** → **Registry** → **Vocab/Glossary** → **Mapping** → **Products** → **Governance**.

2) Jump in based on your task

- Adding a new source dataset? Go to **Datasets** and **Mapping**
- Publishing a harmonized output? Go to **Products**, **Standards**, and **Releases**
- Changing schema/vocab/rules? Go to **Governance**
- Confused by a term? Go to **Glossary**
- Need allowable values? Go to **Vocab**

3) Use it as a reference

Think of **Registry** / **Vocab** / **Glossary** as the “API docs” for AKSDB.

Quick links

- Start here: [Getting Started](#)
- Mental model: [Concepts](#)
- Metadata spine: [Registry](#)
- Allowed values: [Vocab](#)
- Definitions: [Glossary](#)
- Source datasets: [Datasets](#)
- Harmonized outputs: [Products](#)
- Crosswalk rules: [Mapping](#)
- Standards alignment: [Standards](#)
- Change process: [Governance](#)
- Copy/paste patterns: [Examples](#)

The AKSDB object model (bird’s-eye view)

AKSDB documentation is organized around a few “objects” you’ll see everywhere:

Object	Key	Where documented	Purpose
Dataset	ds_iid	Datasets	A source dataset (pre-harmonization)
Product	product_slug (+ canonical ID)	Products	A harmonized publishable output
Entity	canonical entity ID	Registry	A described thing (table/file/layer/artifact)
Party	canonical party ID	Registry	A person or organization

Object	Key	Where documented	Purpose
Role	controlled vocab	Vocab + Standards	Attribution and responsibility
Term	<code>term_id</code>	Glossary	Canonical definition used across docs

Public vs private (two-repo worldview)

AKSDB v2 usually operates as two surfaces:

- **Private workspace:** raw inputs, restricted datasets, internal notes, intermediate artifacts
- **Public repo:** harmonized products, documentation, open metadata, citations

This book aims to describe the system in a way that supports both—without leaking private data.

Guarantees (what AKSDB tries to make true)

AKSDB v2 aims for:

- **Referential stability:** IDs don’t change once published
- **Reproducibility:** releases can be rebuilt (same inputs + same rules)
- **Traceability:** products can be traced to datasets and mapping rules
- **Controlled language:** vocabularies and glossary prevent “semantic drift”
- **Governed change:** schema and rules evolve with explicit deprecation/versioning

Where to go next

If you are new to the project:

1. Read [Getting Started](#)
2. Read [Concepts](#) (IDs → slugs → releases → checksums)
3. Skim [Registry](#) to understand the metadata spine
4. Use [Examples](#) to do real work quickly

Citation and licensing

AKSDB v2 is intended to be **citable** at multiple levels:

- the overall project/release
- each harmonized product
- each source dataset (original citation preserved)

(See **Standards** for how we align with DataCite / EML, and how roles map to CRediT.)

Project status

This documentation is evolving alongside: - the registry tables and controlled vocabularies, - mapping rules for key sources, - and the first “stable” harmonized product releases.

If something feels incomplete, check **Governance** for how to propose changes and add missing pages.

Part I

Concepts::concepts

1 Concepts

The AKSDB v2 mental model: IDs \rightarrow slugs \rightarrow releases \rightarrow checksums

1.1 Why these concepts exist

AKSDB v2 is designed for: - reproducibility (same inputs \rightarrow same outputs), - long-lived references (stable identifiers), - safe evolution (versioning + deprecation), - traceability (checksums + provenance).

Everything else in the docs assumes this mental model.

1.2 The core chain

- 1) IDs **IDs** are canonical, stable identifiers for machines and long-term references. They should be:
 - unique,
 - immutable once published,
 - never “meaningful” in a way that invites reinterpretation.
- 2) Slugs **Slugs** are human-friendly names used for URLs and docs navigation. They can change (with redirects / alias rules), but must never break ID-based references.
- 3) Releases A **release** is a named snapshot of:
 - schema,
 - vocab,
 - mappings,
 - products (harmonized outputs), with explicit compatibility expectations.
- 4) Checksums **Checksums** anchor trust:
 - detect unintended change,

- support manifests,
- enable provenance (“this file came from these inputs under this mapping version”).

1.3 In this section

- **IDs & slugs:** what is stable, what can change, and how aliases work
- **Releases:** what constitutes a release, and how we version it
- **Checksums & provenance:** manifests, file integrity, and lineage
- **Validation checks:** what we assert about tables and outputs

1.4 Suggested next pages

- [IDs & slugs](#)
- [Releases & versioning](#)
- [Checksums & provenance](#)
- [Validation checks](#)

Part II

Registry::registry

2 Registry

Metadata registry tables: packages, entities, parties, roles

2.1 What the registry is

The registry is the **metadata spine** of AKSDB v2. It defines the “who/what/where/why” across datasets and products:

- **packages**: publishable units / bundles (datasets, products, releases)
- **entities**: things described by metadata (tables, files, layers, artifacts)
- **parties**: people and organizations
- **roles**: how parties relate to entities (author, curator, funder, etc.)

2.2 How to use this section

1. Read the table docs (structure + field meanings)
2. Use the data dictionary to enforce consistent meanings
3. Refer back here whenever you add a dataset/product page

2.3 Registry docs

- [packages](#)
- [entities](#)
- [parties](#)
- [data dictionary](#)

2.4 Design notes

- Registry rows should be referencable by canonical IDs
- Human-readable display names belong in slugs/labels, not in IDs
- Controlled vocabularies should be referenced, not duplicated

3 packages::Packages

Notes

File: metadata/registry/packages.csv

Grain: 1 row = 1 released snapshot (a frozen package)

Purpose: Dataset/release-level metadata: identity, provenance, coverage, rights, and integrity checks for a snapshot.

Keys & relationships

- **Primary key:** release_id
- **Stable concept key:** dataset_id
- **Immutable handle:** ds_iid (slug; used in scripting; must not change once assigned)
- **Joins**
 - entities.release_id → packages.release_id
 - party_roles.release_id → packages.release_id

Required minimum (recommended)

- dataset_id, ds_iid, release_id, release_slug
- title, abstract
- ingest_date
- license (or explicit intellectual_rights)
- file_manifest_path, manifest_sha256, file_count, total_bytes
- 1 creator + 1 contact via party_roles

The **packages** registry is the authoritative *release ledger* for AKSDB. Each row records a **frozen snapshot** of a dataset (a published package) with a stable identity, provenance pointers, basic descriptive metadata, coverage fields, rights/access information, and integrity checks (manifest + checksums). Downstream, this table drives dataset/release index pages, enables deterministic joins to **entities** and **party_roles**, and supports reproducible citation/export to metadata standards (e.g., DataCite/EML) by providing a single, versioned source of truth for each released package.

3.1 Columns

Column label	Column name	Type	Format	Description
Dataset concept ID	<code>dataset_id</code>	string	UUIDv4	<p>Stable identifier for the dataset concept across time (all releases/snapshots share the same <code>dataset_id</code>). Enables stable joins and lineage even if names/slugs change (though <code>ds_iid</code> should be immutable too). For a given <code>ds_iid</code>, <code>dataset_id</code> must be consistent across all rows.</p> <p><i>Example:</i> <code>550e8400-e29b-41d4-a716-44666</code></p>

Column label	Column name	Type	Format	Description
Dataset handle (slug)	ds_iid	string	REGEX:^[a-z0-9]+(-[a-z0-9]+)*\$	Human-readable dataset handle used in scripting and folder naming. Stable, memorable reference to dataset concept (e.g., in pipelines, CLIs, configuration). Must be globally unique across the registry; never reused. Must not change for a given dataset_id. <i>Example:</i> <i>nrcs-nasis-pedons-ak</i>
Release (snapshot) ID	release_id	string	UUID (recommend UUIDv7)	Unique identifier for a frozen snapshot of a dataset at a point in time.
- Must be a valid UUID.				
Release (snapshot) slug	release_slug	string	parseable slug (project convention)	Human-readable identifier for the snapshot, typically derived from ds_iid + dates + internal counter.
- Must be unique within the registry.				
Examples:				
-				
				nrcs-nasis-pedons-ak--ing20251231--r003

Column label	Column name	Type	Format	Description
Title	title	string		Human-readable dataset/release title appropriate for citation.
Abstract	abstract	string		Short description of what the dataset/release contains and why it exists.
Purpose	purpose	string		Optional statement of intended use / motivation (helpful for EML-style metadata).
Source organization	source_org	string		Organization responsible for producing or stewarding the upstream dataset.
Source citation	source_citation	string		Preferred citation text for the upstream dataset (include DOI if available).
Landing page URL	landing_page_url	string		Public landing page for the dataset or release (repository record, project page).
DOI	doi	string		DOI for the dataset release/version, if minted.

Column label	Column name	Type	Format	Description
EML package identifier	eml_packageId	string		Repository-specific EML package ID if publishing to an EML-based repository (e.g., EDI/ADC).
Repository identifiers	repo_scope repo_identifier repo_revision	string		Optional split fields for package identification/versioning within a repository.
Coverage window	coverage_start coverage_end	date or string	YYYY-MM-DD (preferred) or YYYY/YYYY-MM if partial	Temporal extent of the observations represented in this release (not ingest date).
Published/updated date (upstream)	published_date	date		Provider-reported publication or last-updated date for the upstream dataset snapshot.
Ingest date	ingest_date	date		Date this snapshot was frozen/ingested into your harmonization pipeline.
	geo_description west east south			

Column label	Column name	Type	Format	Description
Spatial coverage (bbox + description)	north	string + float		Overall spatial foot- print/description for the release; bbox coordinates in decimal degrees.
Spatial reference	spatial_reference	string		EPSG code, WKT, or other SRS description if relevant.
License	license	string		License identifier (prefer SPDX where possible).
Intellectual rights / usage constraints	intellectual_rights	string		Free-text rights statement when a standard license is not sufficient.
Methods & processing	methods_summary processing_summary	string		High-level methods and any key processing steps performed for this release.
Upstream lineage	upstream_release_id	string	##124;- separated list of release_id	Release IDs that this release derives from or depends on (lineage).
File checksum manifest path	file_manifest_path	string		Relative path (within release folder) to the per-file checksum manifest (e.g., check- sums.sha256).

Column label	Column name	Type	Format	Description
Dataset fingerprint (manifest hash)	manifest_sha256	string	sha256 hex	SHA-256 of the checksum manifest file.
File count	file_count	int		Number of files/entities included in this release.
Total bytes	total_bytes	int		Sum of size_bytes across entities for this release.
Notes	notes	string		Free-text notes, known limitations, quirks, etc.

Part III

Vocabulary::vocab

4 Vocab

Controlled vocabularies: enums, roles, status fields

4.1 What belongs here

Anything that should be **standardized** and **validated** belongs in **vocab/**, including: - roles (aligned with CRediT where applicable), - status fields (draft/published/deprecated), - entity types, - license identifiers, - field-level enums.

4.2 How vocab is used

- Registry tables reference vocab values
- Validation checks assert only allowed values appear
- Releases snapshot vocab at a point in time

4.3 Start here

- Roles vocabulary (registry + standards)
- Status vocabulary (draft → published → deprecated)
- Shared enums used across tables

5 Controlled vocabularies standard

This project uses **controlled vocabularies (CVs)** to standardize categorical values across tables, pipelines, and exports. A controlled vocabulary is a curated list of allowed concepts with stable identifiers, labels, definitions, and governance metadata. Using CVs improves interoperability, reduces ambiguity, and enables consistent joins across harmonized datasets.

5.1 What a controlled vocabulary CSV represents

- **One CSV = one vocabulary** (one list of concepts for one topic such as methods, roles, units, properties).
- **One row = one concept**.
- The **machine value** used in data products is stored in `term_code`.
- The **stable project identifier** used for joins and governance is stored in `concept_iid`.

5.2 Identifiers and namespaces

- `concept_iid` is a stable identifier minted by this project (typically in the `aksdb:` namespace).
- Even if a term originates from an external standard (EML, ISO, etc.), the project still mints its own `concept_iid` so the record can be versioned, governed, and referenced consistently.
- External provenance and mappings are captured in `source`, `exact_match_ids`, and `close_match_ids`.

5.3 Standard column set

All controlled vocabulary CSVs in this project use the following columns:

- `concept_iid`
- `term_code`
- `pref_label`
- `alt_labels`

- `definition`
- `status`
- `created`
- `modified`
- `scope_note`
- `related_ids`
- `exact_match_ids`
- `close_match_ids`
- `replaced_by`
- `source`
- `note`

5.3.1 Governance fields

- `status` indicates lifecycle state. Allowed values:
 - `accepted` (current valid term)
 - `proposed` (candidate term under review)
 - `deprecated` (do not use for new data; may have a replacement)
 - `draft` (work-in-progress, not ready for use)
- `created` and `modified` track history and **must be ISO 8601**: YYYY-MM-DD.

5.3.2 Labels and definitions

- `pref_label` is the human-facing label used in UIs and reports.
- `alt_labels` captures synonyms, abbreviations, and common variants to improve search and mapping.
- `definition` is a short, stable description of the concept's meaning.
- `scope_note` clarifies boundaries, intended usage, and edge cases.

5.3.3 Relationships and mappings

- `related_ids` can point to other `concept_iid` values when a useful non-hierarchical relationship exists.
- `exact_match_ids` is for external identifiers that are equivalent in meaning (for example, an authoritative schema or standard identifier).
- `close_match_ids` is for external identifiers that are similar but not strictly equivalent (crosswalks).
- `replaced_by` supports deprecation by pointing to the preferred successor concept when `status=deprecated`.

5.4 Multi-value encoding convention

When a single cell contains multiple values, the project enforces this encoding:

- Enclose the set in curly braces
- Separate values with pipes

Examples:

- `alt_labels = {EC|electrical conductivity|conductivity}`
- `related_ids = {aksdb:foo/a|aksdb:foo/b}`

Empty cells represent “no value”.

5.5 CSV quoting and commas

CSV uses commas to separate fields. If a cell contains a comma, a newline, or a double quote, the value must be quoted using double quotes. Where possible, keep definitions and notes free of commas to minimize quoting and reduce noisy diffs.

5.6 Regex and validation

Regular expressions (regex) are used in validation to ensure consistent value shapes. Common uses:

- Enforce identifier formats (for example, `concept_iid` prefix rules)
- Enforce machine code formats (`term_code`)
- Validate multi-value fields (either empty or `{...}` with `|` separators)
- Validate date formats (`created`, `modified`)

Recommended regex checks:

- Multi-value cell is either empty or wrapped in braces:
 - `^(\{[^\}]*\}|)$`
- Basic term code (letters only, supports CamelCase):
 - `^[A-Za-z][A-Za-z]*$`
- ISO date:
 - `^\d{4}-\d{2}-\d{2}$`

5.7 CSVW metadata sidecars

Each controlled vocabulary CSV should include a matching CSVW metadata file:

- `<vocab>.csv-metadata.json`

The CSVW file provides machine-readable descriptions of: - column datatypes - required fields - controlled enums (such as `status`) - separators for multi-value fields

This supports automated validation and consistent parsing across tools.

5.7.1 What CSVW is for

CSVW (CSV on the Web) is a W3C standard for describing CSV files with machine-readable metadata. In this project, the CSVW sidecar is used as the **authoritative schema** for each vocabulary so that scripts and pipelines do not have to guess how to interpret the CSV.

5.7.2 How to interpret the CSVW file

Key fields you will see:

- `url`: which CSV file this metadata describes.
- `dialect`: how the CSV is formatted (delimiter, quote character, encoding, and whether there is a header row).
- `tableSchema.primaryKey`: the column or columns that uniquely identify rows.
- `tableSchema.columns`: per-column rules:
 - `name`: the exact column header in the CSV.
 - `required`: whether empty values are allowed.
 - `null`: which values are treated as “missing” (this project uses `""`).
 - `datatype`: the expected datatype (for example `date`, `string`, `anyURI`) and optionally a `format` pattern (regex-like).
 - `separator`: how to split multi-valued cells (this project uses `|` in columns whose CSV values are encoded as `{a|b|c}`).

5.7.3 How to use CSVW in this project

Use the CSVW file in any workflow that reads vocabularies:

- **Validation (recommended in CI):**
 - check required fields are present
 - check `status` is one of the allowed enum values

- check `created` and `modified` are valid ISO dates
- check multi-value fields conform to `{...}` and split cleanly on `|`
- **Parsing (ETL and ingestion):**
 - use the `separator` metadata to parse multi-value fields into arrays/lists consistently
 - use `datatype` to parse and type-cast fields consistently (for example dates)
- **Documentation generation (optional):**
 - column names and constraints can be derived from CSVW to keep validation rules synchronized with the CSV

5.7.4 Canonical multi-value parsing rule

For any column that is both: - encoded as `{a|b|c}` in the CSV, and - has `"separator": "|"` in CSVW,

the canonical parse is:

- 1) if blank → empty list
- 2) else strip leading `{` and trailing `}`
- 3) split on `|`
- 4) trim whitespace on each value

5.8 Documentation format for each vocabulary

Each vocabulary must have a companion Markdown documentation page that includes:

1. A brief narrative introduction (rationale + what the vocabulary is for)
2. A “Controlled vocabulary file” section with:
 - controlled vocabulary name / table name (filename)
 - label (human-readable label)
3. A “Columns” section with a table:
 - Column label (human-readable)
 - Column name (exact CSV header)
 - Description (detailed)
4. A “References” section that points to the authority defining the terms and any mapping standards

6 eml_responsibility_roles

This controlled vocabulary enumerates the allowed **EML Party responsibility roles** (EML **RoleType**) used to describe how a person or organization is associated with a dataset or other resource (for example, author, originator, custodian/steward, distributor, and point of contact). Each row provides a stable AKSDB identifier (**concept_iid**) for joining and governance, and the exact EML machine value (**term_code**) that should be written into EML metadata.

The vocabulary also includes optional crosswalks to ISO 19115 **CI_RoleCode** values via **close_match_ids** when a reasonable “close match” exists.

6.1 Controlled vocabulary file

- **Controlled vocabulary name / table name:** `/metadata/vocab/eml_responsibility_roles.csv`
- **Label:** EML responsibility roles

6.2 Columns

Column label	Column name	Description
Concept identifier	concept_iid	Stable, project-minted identifier for the concept. Used as the primary key for joins and internal governance (do not recycle). Pattern in this vocab: <code>aksdb:eml_responsibility_role/<RoleType</code>
EML role code	term_code	The exact machine-readable EML RoleType literal value to write into EML (case sensitive). Examples include <code>principalInvestigator</code> and <code>pointOfContact</code> .

Column label	Column name	Description
Preferred label	<code>pref_label</code>	Human-readable display label for the role.
Alternative labels	<code>alt_labels</code>	Optional synonyms and variants for search and display. Multi-values are encoded as <code>{value1 value2 ...}</code> .
Definition	<code>definition</code>	Short definition of the role concept (project-readable).
Status	<code>status</code>	Governance status for the concept. Allowed values: <code>accepted</code> , <code>proposed</code> , <code>deprecated</code> , <code>draft</code> .
Created	<code>created</code>	Date the row/concept was introduced to the vocabulary in ISO format YYYY-MM-DD.
Modified	<code>modified</code>	Date the row/concept was last changed in ISO format YYYY-MM-DD.
Scope note	<code>scope_note</code>	Optional usage guidance and boundaries for applying the role.
Related identifiers	<code>related_ids</code>	Optional related concept identifiers (non-hierarchical). Multi-values are encoded as <code>{value1 value2 ...}</code> .
Exact match identifiers	<code>exact_match_ids</code>	External identifier(s) that are considered exact matches. In this vocab this is the authoritative EML XSD URL used as the source definition for RoleType.
Close match identifiers	<code>close_match_ids</code>	External identifier(s) that are close (not exact) matches. Here this is used for ISO 19115 <code>CI_RoleCode</code> crosswalks when applicable. Multi-values are encoded as <code>{value1 value2 ...}</code> .

Column label	Column name	Description
Replaced by	<code>replaced_by</code>	If <code>status=deprecated</code> , the <code>concept_iid</code> of the preferred replacement concept.
Source	<code>source</code>	Human-readable documentation URL for the authority defining the vocabulary values (here: EML Party/RoleType documentation).
Note	<code>note</code>	Optional editorial notes and provenance details. Multi-values are encoded as <code>{value1 value2 ...}</code> .

6.3 References

- EML Party / RoleType documentation: see `source` column in `eml_responsibility_roles_iso.csv`.
- EML 2.1.1 Party schema (XSD): see `exact_match_ids` column in `eml_responsibility_roles_iso.csv`.
- ISO 19115 `CI_RoleCode` codelist (used for close matches): see the ISO codelist URL referenced in the `note` column for each mapped term.

7 datacite_contributor_roles

This controlled vocabulary enumerates the allowed **DataCite Contributor contributorType** values (roles) that may be used to describe the contribution made by a person or organization in DataCite metadata.

The vocabulary also includes optional crosswalks to the **CRedit contributor roles taxonomy** via `close_match_ids` when a reasonable “close match” exists.

7.1 Controlled vocabulary file

- **Controlled vocabulary name / table name:** `/metadata/vocab/datacite_contributor_roles.csv`
- **Label:** DataCite contributor roles

7.2 Columns

Column label	Column name	Description
Concept identifier	<code>concept_iid</code>	Stable, project-minted identifier for the concept. Recommended pattern for this vocab: <code>aksdb:datacite_contributor_role/<ContributorType></code>
Preferred label	<code>label</code>	Human-readable label for the role (typically the same as the DataCite controlled value, optionally spaced for readability).
Term code	<code>term_code</code>	The exact machine-readable DataCite <code>contributorType</code> literal value that should be written into DataCite metadata.

Column label	Column name	Description
Definition	<code>definition</code>	Short definition of the role (sourced from DataCite controlled list definitions where available).
Status	<code>status</code>	Governance status for the concept. Allowed values: <code>accepted</code> , <code>proposed</code> , <code>deprecated</code> , <code>draft</code> .
Created	<code>created</code>	Date the row/concept was introduced to the vocabulary in ISO format YYYY-MM-DD.
Modified	<code>modified</code>	Date the row/concept was last changed in ISO format YYYY-MM-DD.
Scope note	<code>scope_note</code>	Optional usage guidance and boundaries for applying the role.
Related identifiers	<code>related_ids</code>	Optional related concept identifiers (e.g., broader/narrower/hierarchical). Multi-values are encoded as <code>{value1 value2 ...}</code> .
Exact match identifiers	<code>exact_match_ids</code>	External identifier(s) for the authoritative DataCite XSD enumeration and/or per-term documentation anchor used as the source definition for <code>contributorType</code> . Multi-values are encoded as <code>{value1 value2 ...}</code> .
Close match identifiers	<code>close_match_ids</code>	External identifier(s) for close matches (e.g., CRediT role URLs) where applicable. Multi-values are encoded as <code>{value1 value2 ...}</code> .
Replaced by	<code>replaced_by</code>	If <code>status=deprecated</code> , the <code>concept_iid</code> of the preferred replacement concept.

Column label	Column name	Description
Source	source	Human-readable documentation URL for the authoritative definition of the vocabulary values (here: DataCite contributorType documentation).
Note	note	Optional editorial notes and provenance details. Multi-values are encoded as {value1 value2 ...}.

7.3 References

- DataCite **contributorType** controlled list definitions: <https://datacite-metadata-schema.readthedocs.io/en/latest/appendices/appendix-1/contributorType/>
- DataCite **contributorType** schema enumeration (XSD include): <https://schema.datacite.org/meta/kernel/4.6/include/datacite-contributorType-v4.xsd>
- CRediT role descriptors (used for close matches): <https://credit.niso.org/contributor-roles-defined/>

8 credit_roles

This controlled vocabulary enumerates the allowed **CRediT (Contributor Role Taxonomy)** contributor roles.

The vocabulary uses a stable internal identifier (`concept_iid`) and a machine-friendly code (`term_code`) that can be used in structured metadata. The authoritative role descriptor pages are referenced via `exact_match_ids`.

8.1 Controlled vocabulary file

- **Controlled vocabulary name / table name:** `/metadata/vocab/credit_roles.csv`
- **Label:** CRediT roles

8.2 Columns

Column label	Column name	Description
Concept identifier	<code>concept_iid</code>	Stable, project-minted identifier for the concept. Recommended pattern for this vocab: <code>aksdb:credit_role/<RoleCode></code> .
Preferred label	<code>pref_label</code>	Human-readable label for the role (as shown in CRediT).
Term code	<code>term_code</code>	Machine-friendly role code (letters only) used for clean programmatic identifiers.
Alternate labels	<code>alt_labels</code>	Optional synonyms / label variants. Multi-values are encoded as <code>{value1 value2 ...}</code> .

Column label	Column name	Description
Definition	definition	Short definition of the role (from the authoritative CRediT role descriptors).
Status	status	Governance status for the concept. Allowed values: accepted , proposed , deprecated , draft .
Created	created	Date the row/concept was introduced to the vocabulary in ISO format YYYY-MM-DD.
Modified	modified	Date the row/concept was last changed in ISO format YYYY-MM-DD.
Scope note	scope_note	Optional usage guidance and boundaries for applying the role.
Related identifiers	related_ids	Optional related concept identifiers (e.g., broader/narrower/hierarchical). Multi-values are encoded as {value1 value2 ...}.
Exact match identifiers	exact_match_ids	External identifier for the authoritative role descriptor (CRediT per-role URL).
Close match identifiers	close_match_ids	Optional external identifiers for close matches (e.g., other role taxonomies). Multi-values are encoded as {value1 value2 ...}.
Replaced by	replaced_by	If status=deprecated , the concept_iid of the preferred replacement concept.
Source	source	Human-readable documentation URL for the authority defining the vocabulary values (here: CRediT role descriptors).

Note

note

Optional editorial notes and
provenance details.
Multi-values are encoded as
{value1|value2|...}.

8.3 References

- CRediT role descriptors (authoritative list + definitions): <https://credit.niso.org/contributor-roles-defined/>
- CRediT home (overview + role list): <https://credit.niso.org/>
- Guidance on using/implementing CRediT: <https://credit.niso.org/implementing-credit/>
- CRediT as an ANSI/NISO standard (background): <https://casrai.org/credit/>

Part IV

Glossary::glossary

9 Glossary

Canonical definitions of terms used across AKSDB

9.1 Purpose

The glossary prevents “same word, different meaning” drift.

A glossary entry should: - define the term precisely, - list synonyms/aliases, - specify scope (where it applies), - link to related registry fields and standards mappings.

9.2 In this section

- [All terms](#)
- [Organic carbon](#)

9.3 Authoring rule of thumb

If a term appears in: - registry field descriptions, - mapping rules, - product documentation, and could be interpreted multiple ways—add a glossary entry.

10 Glossary

10.1 Overview

This glossary ([?@sec-term-glossary](#)) defines terms used throughout the AKSDB project documentation. It provides one or more externally sourced definitions of the term (**Definitions**), the specific usage of the term in the AKSDB project (**AKSDB usage**), and relevant comments, including associated terms and overlapping terms or concepts (**Comments**) and a list of relevant references or other relevant resources (**References/Other Resources**).

10.1.1 Concept

Definitions:

1. a general idea or notion, a universal; a mental representation of the essential or typical properties of something, considered without regard to the peculiar properties of any specific instance or example. Later often (frequently with of): the meaning that is realized by a word or expression. (Press 2026)
2. a unit of knowledge created by a unique combination of characteristics. (Earley 2011)

AKSDB usage:

In AKSDB a concept is a stable unit “thing you mean”, which is an abstract representation and distinct from the term used to define it. Terms can change without changing the concept and terms can have synonyms - concepts must be distinct.

Comments:

In concept-centric systems (thesauri/SKOS), a concept is the abstract meaning; terms/labels are the words/phrases used to denote it (preferred label, alternative label/synonym) (**w3cSKOSSimpleKnowledge2009?**). ISO/IEC 11179 distinguishes the concept (“data element concept”) from its representation (“data element” with a value domain) (ISO 2023). “Term registry” often becomes ambiguous if it stores both concept records and labels—prefer naming it concept registry if one record = one concept.

References/Other Resources:

SKOS (Miles and Bechofer 2009) ISO/IEC 11179 Metadata Registries (MDR) Family (ISO 2023)

10.1.2 Data**Definitions:**

1. Facts represented as text, numbers, graphics, images, sound, or video. Data is the raw material used to represent information, or from which information can be derived (G. Everest, 2010) (Earley 2011)
2. The individual facts that are out of context, and have no meaning by themselves. They are often referred to as raw data, such as [the decimal number] 123.45. Data have historically been defined a plural; datum is the singular form (Brackett, 2011). (Earley 2011)

AKSDB usage:

Same as definitions above.

Comments:

“Data become ‘information’ when analyzed and possibly combined with other data in order to extract meaning, and to provide context. The meaning of data can vary depending on its context. ‘Data’ includes, but is not limited to, 1) geospatial data 2) unstructured data, 3) structured data, etc. Information, as defined in OMB Circular A-130, means any communication or representation of knowledge such as facts, data, or opinions in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audiovisual forms.” (Technology Transformation Service (TTS), Office of Government and Information Services (OGIS), and Office of Management and Budget (OMB) 2026)

References/Other Resources:

10.1.3 Data dictionary

Definitions:

1. a document that “defines and describes the elements of a dataset so that it can be understood and used at a later date” (often variable/field/column-focused) (School 2026).
2. any place where...technical terms and definitions are stored. Typically, data dictionaries are designed to store a limited set of available [metadata], concentrating on the names and definitions relating to the physical data and related objects...(Earley 2011)
3. A collection of metadata describing the contents, format, and structure of a database and the relationship between its elements, used for controlling access to and manipulation of the database. (Press 2026)

AKSDB usage:

A table/field-level technical reference, generated from CSV registries (e.g., `fields.csv`, `value_domains.csv`), rendered into docs.

Comments:

Overlaps with metadata (technical + structural metadata). Differs from glossary/thesaurus in that a dictionary describes how a field is encoded (type/units/allowed values), not the broader concept system. In ecology-oriented metadata systems, the “[attribute](#)” ([variable](#)) [description](#) includes name, definition, domain, coded values, missing values, etc.—functionally a data dictionary embedded in metadata (Jones et al. 2019). There are several broad types of data dictionaries: *active* (interacts with software environment to update metadata in real time), *integrated* (serves as a store for metadata for multiple software tools), and *passive* (requires user entry and update of metadata)(Earley 2011). Based on these classes, the AKSDB data dictionary should be considered a passive data dictionary requiring manual update (although the documentation for the data dictionary is automatically generated from the dictionary csvs themselves, those csvs must be manually updated).

References/Other Resources:

Harvard Medical School - Longwood Research Data Management. [Data dictionary](#).

Matthew B. Jones, Margaret O’Brien, Bryce Mecum, Carl Boettiger, Mark Schildhauer, Mitchell Maier, Timothy Whiteaker, Stevan Earl, Steven Chong. 2019. *Ecological Metadata Language version 2.2.0*. KNB Data Repository. doi:10.5063/F11834T2

National Cancer Institute - Proteomic data commons. [Data dictionary](#). {A very minimal data dictionary example}.

Kristin Briney’s [Create a Data Dictionary exercise](#) in *The Research Data Management Workbook*

Penn Libraries Research Data & Digital Scholarship’s [Data Dictionary Blank Template](#)

Open Science Framework's [How to Make a Data Dictionary](#)

ICPSR's [What is a codebook?](#)

Harvard Biomedical Data Management's [Metadata Worksheet](#)

10.1.4 Data element

Definitions:

1. an item of data within a database or other collection of data. (Press 2026)

AKSDB usage:

Comments:

References/Other Resources:

10.1.5 Dictionary

Definitions:

1. a collection of definitions for words, terms, and phrases that differentiate closely related words. (Earley 2011)
2. a book or electronic resource that lists the words of a language (typically in alphabetical order) and gives their meaning, or gives the equivalent words in a different language, often also providing information about pronunciation, origin, and usage. (Press 2026)

AKSDB usage:

In AKSDB this term is used in the context of **data dictionary**

Comments:

None

References/Other Resources:

None

10.1.6 Glossary

Definitions:

1. ...a dictionary covering a limited subject area. (Earley 2011)
2. a collection of glosses; a list with explanations of abstruse, antiquated, dialectal, or technical terms; a partial dictionary. (Press 2026)

AKSDB usage:

Glossary = human-facing canonical definitions and usage notes for terms used in AKSDB documentation.

Comments:

May overlap with controlled vocabularies although the AKSDB glossary defines terms in use throughout the human-facing project documentation, whereas controlled vocabularies are machine-readable, offer even more brevity, and specify terms used in the data itself, not the project documentation. A glossary is a type of **dictionary**.

References/Other Resources:

None

10.1.7 Metadata

Definitions:

1. metadata is “data about other data” / resource description used for discovery and management. (DCMI 2005)
2. metadata as structured documentation for a dataset (content, meaning, provenance, access, variables, etc.) (Alliance 2012)
3. Literally, “data about data”; data that defines and describes the characteristics of other data, used to improve both business and technical understanding of data and data-related processes. Because the term ‘metadata’ is a trademark of The Metadata Company, DAMA specifically uses the term *meta-data*. (Earley 2011)

AKSDB usage:

The metadata in the AKSDB project consists of:....

Comments:

References/Other Resources:

10.1.8 Registry

Definitions:

1. generally, a permanent collection of data related to some topic or collected through some process (Earley 2011)
2. in metadata management, an application which stores metadata for querying...(Earley 2011)
3. a registry that stores information about data elements and associated concepts (e.g., data element concepts, conceptual domains, value domains). (ISO 2023)

AKSDB usage: The AKSDB metadata registry is the authoritative tables (CSVs) that collectively define identifiers, terms/concepts, fields, domains, mappings, and releases.

Comments: The AKSDB registry is the official system-of-record; any associated docs are a presentation layer. A registry can be “semantic” (concepts) without describing physical storage (ISO 11179 explicitly separates semantic registry objects from physical tables/columns). (ISO 2023)

References/Other Resources:

International Organization for Standardization. (n.d.). ISO/IEC 11179-3 (Metadata registries). (ISO 2023)

10.1.9 Term

Definitions:

1. a word or phrase used in a precise sense in a particular subject or field, or by a particular group of people; a technical expression; a piece of jargon. (Press 2026)
2. more widely: any word or phrase expressing a particular idea or concept, or denoting a particular object; an expression (for something). (Press 2026)

AKSDB usage:

A term is a word or phrase used to denote a concept.

Comments:

Terms can have synonyms. There can be many terms to a single concept. Terms can change without changing the concept. *synonym*: label.

References/Other Resources: ISO/IEC 11179 Metadata Registries (MDR) Family (ISO 2023)

10.1.10 Value domain**Definitions:**

1. a value domain is the permitted set/range of values for representing a concept (including enumerations and ranges). (ISO 2023)

AKSDB usage:

In AKSDB, the value domain is the “how it’s represented” rule set: datatype, units, valid range, code list, missing codes, precision/scale rules. This complements concept (meaning) with representation constraints.

Comments:

The EML attribute supports enumerated domains, missing-value definitions, etc., as part of attribute description. (Jones et al. 2019)

References/Other Resources:

International Organization for Standardization. (ISO/IEC 11179 family overview).

Ecological Metadata Language. (n.d.). Enumerated domain (eml-attribute).

ontology, taxonomy, thesaurus - three forms of ??? where was that from?

10.1.11 Catalog

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.12 Semantic

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.13 Element

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.14 Data element

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.15 Dictionary

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.16 Catalog

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.17 Taxonomy

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.17.1 Data thesaurus

DEFINITION: ISO 25964 is the international standard for thesauri; it focuses on selecting concepts/terms and expressing relationships to form a thesaurus (NISO 2026). SKOS positions thesauri/taxonomies/classification schemes as “knowledge organization systems” with shared structure that can be published and linked.

AKSDB USAGE: Data thesaurus = curated term system that supports indexing/search and synonym control across many datasets (Miles and Bechofer 2009).

COMMENTS: A thesaurus is typically concept-centric (preferred labels + synonyms + broader/narrower/related), which overlaps with controlled vocabulary and can be expressed using SKOS. Less formal than an ontology (usually fewer typed relations/constraints)(Miles and Bechofer 2009).

REFERENCES:

International Organization for Standardization. (2011). ISO 25964-1: Information and documentation—Thesauri and interoperability with other vocabularies—Part 1.

Miles, A., & Bechofer, S. (2009). SKOS Simple Knowledge Organization System Reference (W3C Recommendation).

10.1.18 Term

Definitions:

AKSDB usage:

Comments:

References/Other Resources:

10.1.18.1 Glossary

DEFINITION: A glossary is typically a curated list of terms with definitions and usage notes (often human-facing; not necessarily machine-actionable).

AKSDB USAGE: Glossary = human-facing canonical definitions and usage notes for terms you use in docs (this page / section).

COMMENTS: Overlaps with thesaurus/controlled vocab, but a glossary may not include synonyms/reasons or stable identifiers.

REFERENCES: (Project-local; you may also cite ISO 25964 / SKOS if your glossary adopts their structure.).

Miles, A., & Bechhofer, S. (2009). SKOS Simple Knowledge Organization System Reference (W3C Recommendation).

term	definitions	aksdb usage	synonyms-overlaps	comments	references
Data dictionary	A document that “defines and describes the elements of a dataset so that it can be understood and used at a later date” (often variable/column-focused) (School 2026).	Data dictionary = table/field-level technical reference, generated from CSV registries (e.g., <code>fields.csv</code> , <code>value_domains.csv</code>), rendered into docs.	Overlaps with metadata (technical + structural metadata). Differs from glossary/thesaurus in that a dictionary describes how a field is encoded (type/units/allowed values), not the broader concept system. In ecology-oriented metadata systems, the “ attribute ” (variable) description includes name, definition, domain, coded values, missing values, etc.—functionally a data dictionary embedded in metadata (Jones et al. 2019). Add - more information on different conceptual types of data dictionary).	- Harvard Medical School - Longwood Research Data Management. Data dictionary .	

10.1.18.2 slug

DEFINITION:

COMMENT:

REFERENCE:

10.1.18.3 UUIDv4

DEFINITION:

COMMENT:

REFERENCE:

10.1.18.4 UUIDv7

DEFINITION:

COMMENT:

REFERENCE:

10.2 References

Briney, Kristin A. 2020. “Project Close-Out Checklist for Research Data”. May 19. <https://doi.org/10.7907/yjph-sa32>.

Briney, Kristin A. 2023. “Leveling Up Data Management”. June 27. <https://doi.org/10.7907/syk7-3z92>.

Part V

Datasets::datasets

11 Datasets

Per-source dataset documentation (keyed by `ds_iid`)

11.1 What a dataset page is

A dataset page documents a **source dataset** before (or alongside) harmonization: - what it is and where it came from, - spatial/temporal coverage, - collection and processing notes, - access constraints (public vs private), - how it maps into AKSDB v2 entities/products.

Each dataset page is keyed by `ds_iid`.

11.2 What every dataset page should include

- Summary + citation
- Access + license
- Provenance (who, when, how acquired)
- Primary artifacts (tables/files/layers)
- Known issues and QC notes
- Mapping links (crosswalks used)

11.3 Dataset pages

- [MTJ Permafrost](#)
- [KPU GAAR](#)

Part VI

Products::products

12 Products

Harmonized outputs (keyed by `product_slug`)

12.1 What a product is

A **product** is a harmonized, documented output intended for reuse: - tables, - geospatial layers, - derived indicators, - releases of cleaned/harmonized datasets.

Products are keyed by `product_slug` (human-friendly), but should also carry canonical IDs and checksums.

12.2 What every product page should include

- What it contains (schema + files)
- Intended use + non-use cases
- Inputs (datasets + versions)
- Mapping rules used
- Validation checks passed
- Change notes across releases

12.3 Start here

Create product pages once: - mapping rules stabilize, - validation checks are defined, - and you are ready to publish/release.

Part VII

Mappings::mappings

13 Mappings

Crosswalks, transformation rules, and harmonization logic

13.1 Why mapping docs matter

Mappings are where “interpretation” happens: - column crosswalks, - unit conversions, - depth/horizon rules, - categorical harmonization, - join keys and entity resolution.

These docs make harmonized products defensible and reproducible.

13.2 What a mapping doc should include

- Inputs and outputs (tables/fields)
- Rule statements (human-readable)
- Edge cases and precedence rules
- Examples (before/after)
- Version notes (what changed and why)

13.3 Typical mapping topics

- Source → canonical field crosswalks
- Controlled vocab normalization
- ID assignment rules
- Checksums/manifests generated per run

Part VIII

Standards::standards

14 Standards

How AKSDB aligns with EML, DataCite, and CRediT

14.1 Purpose

Standards provide interoperability and citation-quality metadata.

This section documents: - which standards you reference, - what parts you implement, - how registry fields map to them, - where you intentionally diverge (and why).

14.2 Pages

- [EML](#)
- [DataCite](#)
- [CRediT](#)
- [CF Conventions](#)

14.3 Practical output

The goal is to support: - a strong `CITATION.cff`, - dataset/product citations, - machine-readable exports (where appropriate), - consistent attribution via roles.

Part IX

Governance::governance

15 Governance

How schema, docs, vocab, and mappings change over time

15.1 What governance covers

Governance defines: - how you propose changes, - how you version them, - how you deprecate safely, - what guarantees users can rely on.

This is what keeps AKSDB v2 coherent as it grows.

15.2 Pages

- [Schema versioning](#)
- [Adding a field](#)
- [Deprecating a field](#)
- [Glossary rules](#)
- [Roles and attributes](#)

15.3 Governance principle (default)

Prefer: - additive changes, - explicit deprecations, - compatibility notes per release, over silent breaking changes.

Part X

Examples::examples

16 Examples

Minimal worked patterns you can copy-paste

16.1 What belongs here

Examples are small, opinionated templates for: - adding a dataset page, - adding a product page, - writing a mapping doc, - defining a new controlled vocabulary, - running validation checks, - producing a release manifest (checksums + provenance).

16.2 Suggested starter examples

- “Add a dataset” template
- “Add a product” template
- “Mapping doc skeleton”
- “Registry row examples” (packages/entities/parties/roles)
- “Release checklist” (what must be true to publish)

Alliance, Data Documentation Initiative (DDI). 2012. “DDI-Codebook.” *DDI-Codebook (DDI-C)*. <https://ddialliance.org/ddi-codebook>.

DCMI. 2005. “Dublin Core Metadata Initiative : Usage Guide.” *Using Dublin Core*. <https://www.dublincore.org/specifications/dublin-core/usageguide/>.

Earley, Susan, ed. 2011. *The DAMA Dictionary of Data Management*. 2nd ed. New Jersey: Technics Publications, LLC.

ISO. 2023. “ISO/IEC 11179 – Metadata Registry (MDR) Standard | ISO Website.” <https://www.iso.org/standard/78914.html>.

Jones, Matthew, Margaret O’Brien, Bryce Mecum, Carl Boettiger, Mark Schildhauer, Mitchell Maier, Timothy Whiteaker, Stevan Earl, and Steven Chong. 2019. “Ecological Metadata Language Version 2.2.0.” <https://doi.org/10.5063/f11834t2>.

Miles, Alistair, and Sean Bechofer. 2009. “SKOS Simple Knowledge Organization System Reference.” <https://www.w3.org/TR/skos-reference/>.

- NISO. 2026. “ISO 25964 – the International Standard for Thesauri and Interoperability with Other Vocabularies | NISO Website.” https://www.niso.org/schemas/iso25964?utm_source=chatgpt.com.
- Press, Oxford University. 2026. *Oxford English Dictionary (OED)*. Oxford, England: Oxford University Press.
- School, Harvard Medical. 2026. “Data Dictionary | Data Management.” *Data Dictionary*. <https://datamanagement.hms.harvard.edu/collect-analyze/documentation-metadata/data-dictionary>.
- Technology TRanformation Service (TTS), General Services Administration (GSA), Office of Government and Information Services (OGIS), and Office of Management and Budget (OMB). 2026. “Glossary: Federal Enterprise Data Resources.” *Resources.data.gov : A Repository of Federal Enterprise Data Resources*. <https://resources.data.gov/>.