AGENTS.md

Purpose

Use this file as the replication playbook for building a new bilingual academic vocabulary project with the same pipeline used here.

The core pattern is:

1. Store each subject and grade band in a canonical CSV.
2. Validate the CSV against a shared schema.
3. Generate a sorted export for review and lookup.
4. Generate a XeLaTeX booklet source from the canonical CSV.
5. Compile the booklet to PDF.
6. Audit overlap across grade bands and curate higher bands so they are not just copies of lower bands.

This repo proves that workflow for:

- Subjects: `math`, `science`, `social-studies`, `ela`
- Grade bands: `K-2`, `3-5`, `4-5`, `6-8`, `9-12`
- Language pair: English-Japanese

Recommended Repo Structure

Replicate this layout in a new project:

```text
data/
  <subject>/
    grades-k-2/
    grades-3-5/
    grades-4-5/
    grades-6-8/
    grades-9-12/
build/
  <subject>/
    <grade-band>/
  reports/
docs/
scripts/
AGENTS.md
```

Naming convention:

- Canonical dataset:
  - `data/<subject>/grades-<band>/<subject>EnJa<Region>Grade<band>.csv`
- Sorted export:
  - `build/<subject>/grades-<band>/<subject>EnJa<Region>Grade<band>Sorted.csv`
- Booklet TeX:

  - `build/<subject>/grades-<band>/<subject>EnJa<Region>Grade<band>Booklet.tex`
- Booklet PDF:
  - `build/<subject>/grades-<band>/<subject>EnJa<Region>Grade<band>Booklet.pdf`

Example:

- `data/science/grades-6-8/scienceEnJaNyGrade68.csv`

Canonical Dataset Schema

Each canonical CSV should use these columns:

- `term_id`
- `subject`
- `domain`
- `subdomain`
- `grade_band`
- `grade_detail`
- `english`
- `japanese`
- `reading`
- `priority`
- `standards_ref`
- `student_note`
- `teacher_note`
- `example_context`
- `source_status`

Field expectations:

- `term_id`: stable unique identifier, ideally `<subject>-<band>-NNN`
- `subject`: one of `math`, `science`, `social-studies`, `ela`
- `grade_band`: one of `K-2`, `3-5`, `4-5`, `6-8`, `9-12`
- `grade_detail`: more precise target like `K`, `1-2`, `4`, or `9-12`
- `english`: canonical English term for lookup and overlap auditing
- `japanese`: target-language rendering
- `reading`: kana reading for Japanese output
- `priority`: `Core` or `Useful`
- `source_status`: values like `draft`, `reviewed`, `migrated-from-legacy`

Validation expectations used in this repo:

- no duplicate `term_id`
- no duplicate `subject/domain/subdomain/english` combinations
- no missing `reading`
- no invalid `priority` values
- no malformed CSV rows with shifted columns

Scripts To Carry Forward

These scripts are the minimum reusable toolchain:

- `scripts/validate_canonical_dataset.py`
  - validates one canonical dataset
- `scripts/build_sorted_export.py`
  - creates a sorted export in `build/`
- `scripts/build_booklet_tex.py`
  - creates booklet TeX from a canonical dataset
- `scripts/build_overlap_report.py`
  - audits repeated normalized English terms across grade bands within a subject

Optional migration helper from this repo:

- `scripts/build_math_grade45_assets.py`
  - specific to the original legacy math source here
  - do not treat it as general infrastructure unless you are migrating the same legacy format

New Project Setup

For a fresh project, copy:

- `scripts/validate_canonical_dataset.py`
- `scripts/build_sorted_export.py`
- `scripts/build_booklet_tex.py`
- `scripts/build_overlap_report.py`
- `docs/schema.md`
- `docs/workflow.md`
- this `AGENTS.md`

Then update the copied docs for:

- target language pair
- target curriculum or standards region
- subject list
- grade-band coverage
- output naming convention if it differs

Build Sequence For One New Dataset

Create one canonical CSV first. Do not start by trying to build every subject and grade at once.

Recommended order:

1. Create one subject and one grade band.
2. Validate the canonical CSV.
3. Generate the sorted export.
4. Generate the booklet TeX.
5. Compile the PDF.
6. Review the PDF for translation quality, line breaking, and readability.

Commands:

```bash
python3.12 scripts/validate_canonical_dataset.py data/<subject>/grades-<band>/<file>.csv
python3.12 scripts/build_sorted_export.py data/<subject>/grades-<band>/<file>.csv
python3.12 scripts/build_booklet_tex.py data/<subject>/grades-<band>/<file>.csv
xelatex -interaction=nonstopmode -halt-on-error -output-directory=build/<subject>/grades-<band>
build/<subject>/grades-<band>/<file>Booklet.tex
xelatex -interaction=nonstopmode -halt-on-error -output-directory=build/<subject>/grades-<band>
build/<subject>/grades-<band>/<file>Booklet.tex
```

Why two XeLaTeX passes:

- cross-references and table layout settle on the second pass
- the final warning scan should be taken from the second-pass log

Recommended final warning scan:

```bash
rg -n -F \
  -e 'Overfull \\hbox' \
  -e 'Underfull \\hbox' \
  -e 'xeCJK Warning' \
  -e 'undefined references' \
  -e 'Label(s) may have changed' \
  -e 'Rerun to get' \
  build/<subject>/grades-<band>/<file>Booklet.log
```

An empty result is the target.

Booklet Requirements

The booklet generator in this repo already solved the main Japanese output issues. Replicate these decisions in a new project:

- use XeLaTeX, not pdfLaTeX
- use `xeCJK` so Japanese can line-break correctly
- set a Japanese-capable sans font explicitly
- keep Japanese and reading columns wider than English
- use slightly smaller type in Japanese-heavy table cells when needed

Without that setup, long Japanese strings tend to overflow or behave as unbreakable runs.

Content Creation Strategy

Do not treat every band as a fixed clone with minor edits. That was the initial prototype shortcut, and it created heavy overlap.

Better sequence:

1. Build `3-5` or `4-5` first as the prototype band.
2. Add `6-8`.
3. Add `9-12`.
4. Add `K-2`.
5. Run overlap auditing.
6. Curate upper bands so they become genuinely more advanced.

Suggested per-subject emphasis:

- `math`
  - lower bands: number sense, operations, shapes, measurement
  - middle bands: ratios, equations, geometry, statistics
  - upper bands: functions, proof, trigonometry, regression, probability, modeling
- `science`
  - lower bands: living things, weather, materials, observation
  - middle bands: cells, forces, Earth systems, investigations
  - upper bands: genetics, stoichiometry, kinematics, fields, climate systems, modeling
- `social-studies`
  - lower bands: community, maps, rules, citizenship, chronology
  - middle bands: geography, civics, economics, historical thinking
  - upper bands: constitutional interpretation, macroeconomics, historiography, source-based argument
- `ela`
  - lower bands: story language, speaking, beginning writing, print concepts
  - middle bands: theme, evidence, informational text, paragraph structure
  - upper bands: rhetoric, disciplinary discourse, synthesis, research argument, revision

Quality Control Loop

After a subject has multiple grade bands, run:

```bash
python3.12 scripts/build_overlap_report.py
```

That produces:

- `build/reports/grade_band_overlap.md`
- `build/reports/grade_band_overlap_details.csv`

Use the report to find:

- repeated English terms across 2+ grade bands
- especially heavy `6-8` versus `9-12` overlap

The right response is not to remove every repeated term. Some vertical continuity is correct. The goal is to reduce lazy duplication and make higher bands more specialized.

Practical curation rule:

- keep foundational cross-band terms only when they are genuinely essential
- replace generic carryover terms in `9-12` with discipline-specific and task-specific vocabulary

What Worked Best In This Repo

These patterns were effective and should be reused:

- keeping every dataset in the same canonical schema
- generating artifacts into `build/` rather than writing beside source data
- validating before every export/build step
- compiling booklet PDFs immediately after generating TeX
- using overlap auditing after breadth expansion
- curating `9-12` bands last, because that is where duplication was easiest to spot and fix

What To Avoid

- storing source-of-truth data only in ad hoc root-level CSVs
- mixing generated files back into `data/`
- creating new grade bands by simple copy-forward without later curation
- relying on manual sorting instead of generated sorted exports
- using a LaTeX setup that is not Japanese-aware
- expanding breadth indefinitely without running overlap audits

Definition Of Done For A New Subject/Band

A dataset is not done when the CSV merely exists. Treat it as done only when all of these are true:

1. The canonical CSV validates cleanly.
2. The sorted export is generated.
3. The booklet TeX is generated.
4. The PDF compiles successfully after the second XeLaTeX pass.
5. The final log has no significant layout or rerun warnings.
6. The content is reasonable for the intended grade band.
7. If sibling grade bands exist, overlap has been audited and accepted or curated.

Minimal Replication Checklist

For a new project, do this in order:

1. Copy the reusable scripts and docs into the new repo.
2. Create the `data/`, `build/`, `docs/`, and `scripts/` structure.
3. Create one canonical CSV for one subject and one band.
4. Validate it.
5. Build sorted CSV, TeX, and PDF.
6. Confirm Japanese rendering and line breaking work.
7. Add the next grade band for the same subject.
8. Run overlap auditing.
9. Curate the upper band to reduce duplication.
10. Only then expand to more subjects.

Current Repo As Reference Implementation

Use this repo as the reference implementation for:

- canonical CSV structure
- validation rules
- sorted export generation
- Japanese-capable booklet generation
- overlap auditing
- post-prototype curation of upper grade bands

If starting a new project, the process above matters more than the exact current inventory of this repo.