# Technical Project Cycle SDMX\_CL\_Generator

## Initiation and Data Preparation

**Source files:** The list of SDMX code lists is defined in a CSV file (in/Code\_Lists\_GR\_SDMX.csv). The structure includes:

* **agency\_id** — agency code;
* **codelistID** — code list identifier;
* **Name** — name;
* **URL** — link to the XML resource.

**XML download:** The module download\_xml\_gr\_cl.py retrieves files from the global SDMX repository. For correct access, the urn\_to\_sdmx\_url function is implemented, which converts a URN into a direct URL.

**Objective:** ensure centralized control over input sources and create a reproducible base for further processing.

## Code List Parsing

**Module:** parse\_save\_cl.py

**Main function:** parse\_codelist\_v3(xml\_file)

**Tasks performed:**

* Reading XML using ElementTree and SDMX 3.0 namespaces;
* Extracting metadata (agency, version, date, description);
* Building a list of codes (Code) with their descriptions (Description).

**Results are saved into structured tables:**

* **Codes table (DF\_CODE\_COLUMNS):** contains code, agency, description;
* **Code lists table (DF\_CODE\_LISTS\_COLUMNS):** stores scheme characteristics and aggregated indicators.

**Objective:** convert and unify data from SDMX-XML into a structured tabular format for further analytics.

## Analytical Processing

**Module:** analyze\_sdmx\_cl.py

**Main function:** cl\_analysis(...)

**Includes:**

* Counting unique codes and identifying intersections (evaluate\_code\_uniqueness, common\_codes\_def);
* Determining technical characteristics (number of codes, presence of duplicates, degree of duplication);
* Classifying code lists by templates (analyze\_templates.py, SINGLE\_CODELISTS, GROUP\_CODELISTS);
* Saving results to CSV (analysis/all\_cl\_data.csv, analysis/cl\_table.csv).

**Objective:** detect patterns and anomalies in code lists, preparing the ground for efficient RDF generation.

## RDF/TTL Generation

**Module:** gen\_cl\_ttl.py

**Main function:** generation\_ttl(...)

**Helper functions (from get\_funcs.py):**

* get\_concept\_str — generate a triple for an individual concept;
* get\_concept\_scheme\_str — generate a scheme block;
* get\_scheme\_dict and get\_from\_scheme\_dict — dictionaries for connectivity.

**Formation principles:**

* Each agency and code list receives its own URI, generated using a template (gen\_template.py, NEW\_PURL\_CODE, NEW\_PREF\_CODE);
* A consolidated file (cl\_out/code.ttl) and individual files for each code list (sip-sdmx-code-\*.ttl) are created;
* Uses **SKOS**: ConceptScheme, Concept, prefLabel, notation, inScheme.

**Objective:** represent code lists as RDF graphs compatible with SDMX, SKOS, and Linked Open Data.

## RDF Quality Control

**Module:** quality\_check.py

**Main function:** check\_rdf\_quality(file\_path)

**Checks performed:**

* Correctness of prefixes (rdfs, skos, custom);
* Presence of required concept properties (skos:prefLabel, skos:notation, skos:inScheme);
* Completeness of reference properties (rdfs:seeAlso, skos:exactMatch);
* Statistical indicators: number of concepts, distribution across schemes.

**Results:**

* Quality scores are calculated (internal scale: quality\_score\_10);
* A value score (value\_score\_10) is generated, reflecting the practical usefulness of the graph.

**Objective:** automate RDF model auditing and ensure reproducible publication-quality control.

## Output Results

* **CSV reports** in the analysis/ folder — for analysis and duplicate control;
* **RDF/TTL files** in the cl\_out/ folder — for subsequent publication;
* **Quality assessments** — for decision-making on whether to include a file in production or send it for revision.

## Cycle Efficiency

* The cycle ensures full automation of SDMX code list processing.
* Modular architecture: each stage can be executed separately (download, parse, analyze, generate, check).
* Use of templates and centralized namespaces makes the process robust and easily scalable.
* Quality control provides a transparent mechanism for approving files for publication.