# Generating Ontologies from External Code Definitions

This Annex describes how the OWL (RDF/XML) files in this specification are generated from the published ISO and UN Sources. This allows the automated generation of updated ontologies when ISO/UN publish their updates.

# ISO-3166

The source is the XML file published by ISO (and available via subscription) as iso\_country\_codes.xml.

This is processed by two separate XSL files (details below) to produce the OWL files.

There are common algorithms used by each of these files, as follows:

## Camel Case

This turns a published country, subdivision or territory name into a camel case name used for the URI of the ontology element (a NamedIndividual). The steps are as follows:

* Split the name into tokens using the space character
* Convert initial character of each token to uppercase
* Normalize Unicode characters using the NFD algorithm and omit any characters outside the Basic Latin character set
* Remove apostrophes and periods
* Truncate the string at the first character that is not alphanumeric or hyphen

## Country Name Overrides

In general the URIs for countries use the above Camel Case algorithm applied to the published short name of the country (the English short name if there are many).

To ensure uniqueness of URIs, following countries are overridden before applying the above algorithm. This table shows the ISO 3166-1 two character code and the name used:

CC Cocos Keeling Islands

CD Congo Democratic Republic Of

KP Korea Democratic Peoples Republic Of

KR Korea Republic Of

VG Virgin Islands British

VI Virgin Islands US

## Country Codes Processing

The file ISO-3166-1.rdf is produced using the XSL file ISO-3166-Countries.xsl.

The outline of processing as follows:

* Generate the Ontology element
  + Include boiler plate information using OMG’s Specification metadata ontology.
  + Insert the timestamp from the ISO XML file as the Dublin Core issued date of the ontology.
  + Generate a versionIRI using a hard-coded OMG-format timestamp (needs to be updated each version)
* Generate Individuals for the two CodeSets (2 and 3 character alphacodes)
* Generate sameAs statements to allow use of URIs in the adopted version of the LCC Standard: United States, UnitedKingdom and CzechRepublic
* Generate Individuals for 3 Languages which are referenced by Country elements in ISO-3166 but not yet included in the ISO-367-2 Languages standard. The have language codes of 001, 002 and crs. The latter is included in ISO-367-3 but that is not part of the LCC language ontologies.
* Process each Country as follows:
  + Generate URI using Country Name Overrides and Camel Case algorithms in previous section
  + Process labels for country using specific ontology properties for English and French names (short, long, upper case) with other labels represented using hasLocalName
  + Process each language marked as Administrative. Look up the 3 character language code from the ISO file in the LCC languages ontology file in order to find the correct LCC URI to link to.
  + Include any Remarks in English as values of the hasRemarks property.
  + Process the 2 and 3 character codes as separate Individuals of type Alpha2Code and Alpha3Code respectively.

## Subdivision Codes Processing

The XSL file ISO-3166-Subdivisions.xsl produces many ontology files in the subdirectory Regions, each with the name ISO3166-2-SubdivisionCodes-XX.rdf where XX is the 2 character code for the country. A file is only produced if the country has subdivisions or territories.

A further output is the list of such files which is incorporated into the About file by adding boilerplate.

The outline of processing is as follows:

* For each country that has a subdivision or territory
  + Create a file in the Regions subdirectory
  + Generate the Ontology element
    - Include boiler plate information using OMG’s Specification metadata ontology.
    - Insert the timestamp from the ISO XML file as the Dublin Core issued date of the ontology.
    - Generate a versionIRI using a hard-coded OMG-format timestamp (needs to be updated each version)
    - For each category (e.g. “county’, “district”) for the country in the ISO XML file:
      * Generate an Individual of class GeographicRegionKind. The URI is the English name of the category converted to Camel Case and appended to the URI of the country-specific ontology
    - For each subdivision in the ISO XML file
      * Generate an individual of class CountrySubdivision. The URI is the English name of the subdivision converted to Camel Case and appended to the URI of the country-specific ontology. The following overrides are required to ensure uniqueness, where a country has subdivisions with the same name at different levels. The following list the code and the actual name used:

AZ-SA Ski-Municipality

AZ-YE Yevlax-Municipality

AZ-LA Lnkran-Municipality

BG-22 SofiaStolitsa

HU-VM Veszprem-City

LA-VT Viangchan-Prefecture

MU-PU PortLouis-City

MZ-MPM Maputo-City

TW-HSZ Hsinchu-City

TW-CYI Chiayi-City

UZ-TK Toshkent-City

* + - * Link to the category via isClassifiedBy property
      * Link to the country and any parent subdivision
      * If the ISO file has property subdivision-related-country then create a sameAs link to the element in the ISO-3166-1 ontology
      * Process the subdivision code as an Individual of class GeographicRegionIdentifier
      * Recursively process any subdivisions of the subdivision
    - For each territory in the ISO XML file
      * Generate an individual of class Territory
      * Link to the category “Territory” via isClassifiedBy property
      * Link to the country

## UN M49 region Codes

The UN M49 information is used to create regions at different levels.

The processing is as follows, and makes use of two tools prior to applying XSL.

* TARQL, to convert from CSV to RDF (Turtle) . Available at <https://github.com/tarql/tarql/releases>. Note that release 1.2 or later is required.
* Rapper (part of Raptor), to convert from turtle to RDF/XML. Available from <http://download.librdf.org/source/>. Version used was 2.0.15.

The steps are:

* Download the English CSV file as M49.csv from the UN site <https://unstats.un.org/unsd/methodology/m49/overview/>
* Edit the column headers in Row 1 to remove all spaces and hyphens
* Run TARQL on the CSV file using the SPARQL file M49.sparql included in this specification. The command line is as follows:

tarql –dedup 1000 M49.sparql M49.csv >m49.rdf

* Run rapper to convert to RDF/XML. The command line is as follows.

rapper -I turtle -o rdfxml-abbrev m49.rdf >m49.xml

* Apply M49-Format.xsl to the output to clean up the ontology and add boiler plate.

Overall the conversion is as follows. In general each populated column in the CSV file results in an extra level of GeographicRegion Individual; each is linked to its parent using property isSubregionOf.

At the lowest level, that of countries, new individuals are not created, but triples are added linking the counties in the ISO3166-CountryCodes ontology to the M49 ontology. The 3 letter country code from the CSV file is used to look up the correct country URI in the ISO3166-CountryCodes.rdf file.

For each column in the original CSV file the following table states the corresponding ontology element. Columns not mentioned here are ignored. Where a value is repeated only one element is created (e.g. Africa appears in many rows but only one Individual is created). The ontology also includes declarations of Individuals for four GeogarphicRegionKinds which are used as per this table.

|  |  |
| --- | --- |
| **Column** | **Ontology Mapping** |
| Global Name | GeographicRegionKind = Planet |
| Region Name | GeographicRegionKind = Continent |
| Subregion Name | GeographicRegionKind = Region |
| Intermediate Region Name | GeographicRegionKind = Subregion |
| Country or Area | Used to lookup country in ISO3166-CountryCodes.rdf |
| M49 Code | NumericRegionCode property |
|  |  |