

Name of paper	Codelist Management
Theme (e.g. Rules & Guidelines; Improvements etc)	Rules & Guidelines
Lead author	Wendy Rogers
Links to other relevant information (Discuss forum; Website etc)	Discuss: http://discuss.iatistandard.org/t/tech-paper-codelist-management/538

Introduction

The IATI Standard includes a number of [codelists](#). Codelists contain the allowed values for specific IATI Standard element and/or attributes. Each codelist is defined as either 'embedded' or 'non-embedded'.

- Embedded codelists belong to IATI. They can only be updated through the IATI Standard upgrade process because their modification may have functional implications. Embedded codelists are currently versioned in line with the IATI standard: 1.04, 1.05, 2.01, 2.02 etc.
- Non-embedded codelists can belong to either third parties or IATI. They are unversioned. This means that new codes can be added or existing code descriptions amended at the request of the IATI community as required (as long as the change does not infer a change in functionality). The use of non-embedded codelists therefore enables the IATI Standard to be more reactive to external changes.

However, it should be noted that deletion of values from both non-embedded and embedded codelists can only be made as part of an integer IATI Standard upgrade. This is due to non-backward compatibility issues for publishers.

With regard to codelist management the following issues currently require clarification:

- The status of some codelists could be changed from embedded to non-embedded.?
- Should all external vocabularies be imported into their own IATI-duplicated codelist like [OECD DAC CRS Sector Codes](#)
- How to keep non-embedded codelists up to date?
 - Where it is possible to do so, the IATI secretariat will set-up automated process to check codelist values. In practice, this is a human exercise, which creates a large support overhead for the head.

1. Changing the status of some embedded codelists to non-embedded

a. Definitions

i. Proposal

Definition: Embedded codelists contain codes that involve functional logic that impacts on the way in which the standard is interpreted and processed. They are embedded in the standards' schema.

ii. Proposal

Definition: Non-embedded codelists contain codes that qualify data, not processes. They may be generated and maintained by third-parties or IATI.

b. Modifications

i. Proposal

Rule: Embedded codelists can only be changed through the standard upgrade process and are versioned accordingly.

ii. Proposal

Guideline: Non-embedded codelists may be changed at any time subject to appropriate notification and/or consultation and are not versioned.

iii. Proposal

Action: Redefine 30 codelists (as per list below) from embedded to non-embedded during the 2.03 upgrade process

Below is a table of all current embedded and non-embedded codelists.

- Purple indicates codes deemed to be Functional and will not change
- Yellow indicates codes which could change from embedded to non-embedded
- Green indicates codes which are non-embedded and shall remain so.

Codelist	Current Type	Proposed New Type
ActivityDateType	Embedded	Embedded
ActivityStatus	Embedded	Embedded

BudgetType	Embedded	Embedded
OrganisationRole	Embedded	Embedded
OrganisationType	Embedded	Embedded
RelatedActivityType	Embedded	Embedded
TransactionType	Embedded	Embedded
BudgetStatus	Embedded	Embedded
ActivityScope	Embedded	Non-Embedded
BudgetIdentifier	Embedded	Non-Embedded
BudgetIdentifierSector-category	Embedded	Non-Embedded
BudgetIdentifierSector	Embedded	Non-Embedded
BudgetIdentifierVocabulary	Embedded	Non-Embedded
CRSAddOtherFlags	Embedded	Non-Embedded
ConditionType	Embedded	Non-Embedded
ContactType	Embedded	Non-Embedded
DescriptionType	Embedded	Non-Embedded
DisbursementChannel	Embedded	Non-Embedded
DocumentCategory-category	Embedded	Non-Embedded
DocumentCategory	Embedded	Non-Embedded
GazetteerAgency	Embedded	Non-Embedded
GeographicExactness	Embedded	Non-Embedded
GeographicLocationClass	Embedded	Non-Embedded
GeographicLocationReach	Embedded	Non-Embedded
GeographicVocabulary	Embedded	Non-Embedded
GeographicalPrecision	Embedded	Non-Embedded
IndicatorMeasure	Embedded	Non-Embedded
LoanRepaymentPeriod	Embedded	Non-Embedded
LoanRepaymentType	Embedded	Non-Embedded
OtherIdentifierType	Embedded	Non-Embedded
PolicyMarker	Embedded	Non-Embedded
PolicyMarkerVocabulary	Embedded	Non-Embedded
PublisherType	Embedded	Non-Embedded

RegionVocabulary	Embedded	Non-Embedded
ResultType	Embedded	Non-Embedded
SectorVocabulary	Embedded	Non-Embedded
TiedStatus	Embedded	Non-Embedded
VerificationStatus	Embedded	Non-Embedded
AidType-category	Non-Embedded	Non-Embedded
AidType	Non-Embedded	Non-Embedded
CRSChannelCode	Non-Embedded	Non-Embedded
CollaborationType	Non-Embedded	Non-Embedded
Country	Non-Embedded	Non-Embedded
Currency	Non-Embedded	Non-Embedded
FileFormat	Non-Embedded	Non-Embedded
FinanceType-category	Non-Embedded	Non-Embedded
FinanceType	Non-Embedded	Non-Embedded
FlowType	Non-Embedded	Non-Embedded
HumanitarianScopeType	Non-Embedded	Non-Embedded
HumanitarianScopeVocabulary	Non-Embedded	Non-Embedded
IATIOrganisationIdentifier	Non-Embedded	Non-Embedded
IndicatorVocabulary	Non-Embedded	Non-Embedded
Language	Non-Embedded	Non-Embedded
LocationType-category	Non-Embedded	Non-Embedded
LocationType	Non-Embedded	Non-Embedded
OrganisationIdentifier	Non-Embedded	Non-Embedded
OrganisationRegistrationAgency	Non-Embedded	Non-Embedded
PolicySignificance	Non-Embedded	Non-Embedded
Region	Non-Embedded	Non-Embedded
Sector	Non-Embedded	Non-Embedded
SectorCategory	Non-Embedded	Non-Embedded
Version	Non-Embedded	Non-Embedded

2. Replication of external vocabularies

- a. Not all external vocabularies are easy to find, interpret, or use. Some may only be accessible in a single format or have layouts which require data cleaning before use. Therefore importing the external list into its own IATI codelist is useful for both publisher and data users as the lists are subsequently published in machine readable and multiple formats (eg. json, csv & xml). However, some of the external vocabularies are very large and/or are difficult for machines to access and/or change often so in some cases importing the data is not actually practical.

As of Dec 2016, there are 31 distinct vocabularies available across the following vocabulary lists:

[BudgetIdentifierVocabulary.xml](#) (5 vocabs but possibly to be deprecated)

[GeographicVocabulary.xml](#) (6 vocabs)

[HumanitarianScopeVocabulary.xml](#) (2 vocabs)

[IndicatorVocabulary.xml](#) (7 vocabs)

[PolicyMarkerVocabulary.xml](#)

[RegionVocabulary.xml](#)

[SectorVocabulary.xml](#)

- i. Proposal

Action: Create a new metadata category for Vocabulary Codes , one being 'replicated' and the other 'non-replicated'.

- ii. Proposal

Action: Create a new metadata category for Codelists - Update Frequency

- iii. Proposal

Guideline: The decision on which codelists should be replicated needs to be a pragmatic one which reconciles the widespread use of a list, the length of the list and the overhead required to keep the list up to date.

- iv. Proposal

Guideline: Manually replicated codelists should be updated by the Technical Team in line with the agreed Update Frequency

3. Deprecation of retired codes

- a. There's an interesting issue where external codelists delete / retire codes that are still important to be able to use in IATI. The CRS purpose codes are maybe the most obvious example - though countries are an interesting case too. In both cases, having a complete set of codes replicated on IATI sites (so that they are useful for users of the data) should really include historical codes, even if they have been deleted / retired by the source.

- i. Proposal

Guideline: Retired codes should be deprecated but never deleted from codelists maintained or replicated by IATI

4. Shared governance of Organisation Identifier Lists codelist

- a. The org-id project is a collaborative effort between Open Contracting, IATI, 360 Giving, NRGI and other partners to maintain a shared list of organisation identifier lists. It builds on the IATI Organisation Registration Agency list, but adds additional meta-data, and introduces a [new schema](#) and [repository](#) for code information.

See [this post](#) for details of changes brought about.

Adopting this as the replacement to IATI's OrganisationRegistrationAgency codelist will involve:

- Linking to, or maintaining a copy of, the org-id confirmed codelist (all codes with the 'confirmed' property set to true). An API output in IATI XML format is available <http://org-id.guide/download>;
- Pointing users to the search interface for locating an appropriate organisation identifier lists from the IATI documentation. (This is now available at <http://org-id.guide>);
- Submitting all new requests for identifier lists to be included via the org-id project, either through pull-requests to the repository;
- IATI Technical Team staff engaging in the review process for new identifier lists submitted.

Before this can happen:

- A small change is needed to the org-id schema to replace the boolean 'deprecated' flag for codes with a date of deprecation. This will allow the org-id API to provide a backwards compatible list that only removes codes with each IATI version update;
- The repository for org-id needs to move to it's own organisation repository;
- A governance document between the partners in the project needs to be agreed for ongoing management of the list;

Proposal

- Action: The `OrganisationRegistrationAgency` codelist should be renamed to `'OrganisationIdentifierList'`
- Action: Maintenance of this list should be carried out by the 'org-id project'
- Action: The above changes should take place as part of the 2.03 upgrade process

- The codelist xsd allows values for `activation-date` and `withdrawal-date` to be specified
 - It would be useful to allow the `activation-version` and `withdrawal-version` to be specified
 - i. This would allow for easier auto-generation of documentation and changelogs about when codelists were modified
 - Specifying activation and withdrawal dates that correspond with version numbers may also work, and not require a change to the standard
- Codelist mappings
 - At the moment Codelists are mapped to Datasets rather than Schemas
 - i. <https://github.com/IATI/IATI-Codelists/blob/version-2.02/mapping.xml>
 - ii. This does not make sense conceptually
 - A Codelist is part of a Schema, not part of a Dataset
 - iii. Elements are generally defined at a global scope
 - This does not match attributes that have a non-standard root
 - Codelist mappings should refer to a Schema, not a Dataset
 - i. They may then be validated using enumeration facets and other features of XSD
 - ii. These should be included in new versions of the standard
 - iii. These should be added to previous versions of the standard
 - They do not add anything new, merely programmatically describing the correct mapping for validation purposes