International Federation of Reproduction Rights Organisations

ifrro

EDItEUR

ONIX IFRRO Dictionary Notes: Issue 2, May 2010

Part 1: General

1. Many data elements in the ONIX IFRRO formats have controlled (coded) values. These values may be of three kinds:
   (a) Values derived from an external source ("third-party values"). Currently, this category covers a few elements whose values are taken from an ISO standard (eg currency codes) or from an ONIX code list maintained by EDItEUR. Third-party values are given a "namespace" prefix identifying the external source, eg *iso:* or *onix:*
   (b) Values controlled by IFRRO in consultation with EDItEUR ("IFRRO values"). This category covers values that have been considered sufficiently generic and sufficiently well-defined to be usable by IFRRO agencies in general, not just by those agencies that have participated in pilot projects. IFRRO values are given a namespace prefix *ifrro:*
   (c) Local values. This category covers values that have been found necessary in pilot projects, but cannot safely be regarded as generic to other IFRRO agencies. As the formats begin to be implemented for live use between pairs or groups of agencies, it is envisaged that further local values will need to be defined ahead of any possible adoption into generic IFRRO lists, partly because they will be needed for immediate use, and implementers will not be able to wait for wider consultation and approval processes to be completed. Local values will be given a namespace prefix identifying the implementation group concerned. To date, the only such prefix that has been assigned is *ukrro:* for the group comprising CLA, PLS and ALCS. (New implementation groups will need to register an intended namespace prefix.)
2. The XML Schemas for ONIX-DS and ONIX-RP have been set up so that:
   (a) Elements that have third-party values can *only* have third-party values.
   (b) All elements that have IFRRO values can also have local values, provided that the local values have a registered namespace prefix.
   (c) All elements that have local values – which again must have a registered namespace prefix - can also have IFRRO values.
3. Lists of third-party values and IFRRO values, and their definitions, are maintained in the ONIX IFRRO Dictionary. Local values are not carried in the Dictionary, and implementers are responsible for making their own arrangements to manage and maintain them. The code list module maintained by EDItEUR as part of the ONIX IFRRO XML Schemas similarly carries only third-party values and IFRRO values. However, the schemas will validate local values to the extent of checking that they carry a registered namespace prefix. Any validation of individual local values will need to be performed by other means.

4. The ONIX IFRRO Dictionary is in the form of an Excel spreadsheet, listing all third-party and IFRRO values used in the ONIX-DS and ONIX-RP formats. In regular practice, the schema module carrying code lists will be generated automatically from the spreadsheet whenever it needs to be updated. This schema module is separate from the schema files that define the XML structures, and can be replaced without changing the structure files.

5. This means that code list updates can be handled independently of any new versions of the message structures, and such updates would in the normal course of events be expected to occur more frequently than extensions to the structures. As with other ONIX formats, code lists are therefore given their own issue numbering, simply as Issue 1, Issue 2 etc. This numbering is completely independent of the version numbering of the overall schema and documentation (Version 1.0, 1.1 etc).

6. The version of the spreadsheet that accompanies these notes represents Issue 2 of the Dictionary. Future issues may add new values, or clarify the definitions of existing values, or occasionally recommend that usage of an existing value should be deprecated in favour of a new value or values; but, effective from Issue 1, once a value has been published in the Dictionary, it will never be deleted, nor will its meaning be substantively changed.

7. The spreadsheet has eight columns:

   Column A Sequence number of entries within each code list

   Column B Data element name

   Column C Code value

   Column D Definition

   Column E Comments: also used, in third-party lists derived from ONIX, to carry notes that supplement the short definitions

   Column F Whether used in ONIX-RP

   Column G Whether used in ONIX-DS

   Column H The issue number in which a new code was added

   In addition, for greater ease of reference, ONIX-RP elements and their applicable values are shaded light yellow in column 1. ONIX-DS elements and their applicable values are shaded light blue in column 2. Elements that have been used in pilot use cases with local values only are shaded light tan in column 3. Elements that have not been required in any pilot use cases, and for which no values have yet been assigned, are shaded light red in column 3.

8. In addition to simple data elements whose coded values are controlled through the schema and codelists (with additional local codes where necessary, as already described), the ONIX IFRRO formats have a number of composite elements in which there is a second level of coding that depends on the value assigned to another element in the composite. The best example is the “ResourceCategory” composite, which has been designed to provide complete flexibility in accommodating present and future, IFRRO-wide, local, or third-party, resource class coding schemes.

   This composite has the structure:

   <ResourceCategory>
   <ResourceCategoryType>
   <CategoryValue>

The “second-level values” to be used in <CategoryValue> depend on the scheme identified in <ResourceCategoryType>. They could in principle be third-party values, IFRRO values, or local values.

For example, CEDRO and Kopinor have proposed a two-part coding comprising “resource form” and “resource content”, each having a hierarchical numeric coding structure, which might be developed to become agreed IFRRO code lists. CLA and PLS seem likely to require one or more local category types that are significant in their repertoire management.

Currently, any “second-level values”, whether IFRRO-wide, local, or third-party, can only be maintained and validated outside of the XML Schemas.

Part 2: Changes in Issue 2

The following is a summary of the changes made in Issue 2 of the Dictionary. New values added in Issue 2 are shown in red in the spreadsheet, and are numbered '2' in column H.

Deduction Type

A new value *ifrro:GSTChargedOnAdmin* has been added at CAL’s request to enable the amount of any Goods & Services Tax charged against an agency’s administration fees to be specified in cases where the calculation of admin fees and associated tax is not shown in the detailed make-up of the Distribution Amount.

Distribution Type
Distribution Source Qualifier
Imprint ID Type
License ID Type
Mandate ID Type
Mandate Time Point Relator
Name ID Type
Publisher ID Type
Publishing Role
Repertoire Agent Relator
Repertoire Definition
Repertoire ID Type
Repertoire Definition
Resource Category Type
Subject To Payment
Usage Context
Usage Type

In all of the elements listed above, values found necessary for the PLS/CLA implementation of ONIX-DS, and agreed by IFRRO ONIX Governance Board (OGB) to be suitable for general application, have been added.

Resource ID Type

New value *ifrro:URI* has been added at CAL’s request to enable a web resource which features in a Distribution to be identified by a URI.

Tax Type

New values *ifrro:GST* and *ifrro:PAYG* have been added at CAL’s request to enable Goods & Services Tax and Pay As You Go Tax (Australia) to be specified in tax calculations.

Category Value

The new value *ifrro:PublicationClass* in **Resource Category Type** is associated with a new “second level” list for use in the **Category Value** element, also derived from the PLS/CLA implementation. This list appears on a new Sheet 2 of the spreadsheet (“Second level values”).