Guidelines on using and extending and IFLA namespaces (working draft)

IFLA Namespaces Technical Group

**Version 0.0, August 2013**

# Using namespace elements

IFLA namespaces are maintained in the Open Metadata Registry.

IFLA namespace elements include classes and properties from element sets and concepts from value vocabularies.

## Access

Namespaces elements can be found and identified by:

* Browsing the list of IFLA namespaces in the Registry: <http://iflastandards.info/ns/>
* Browsing the lists of namespaces for specific IFLA standards:
	+ FR family: <http://iflastandards.info/ns/fr/>
	+ ISBD: <http://iflastandards.info/ns/isbd/>
	+ MulDiCat: tba
	+ UNIMARC: tba
* Browsing a specific element set or value vocabulary. [See table]
* De-referencing the element by using its URI as a URL in a normal Web browser.

## RDF

An RDF/XML serialization of an element or its containing element set or value vocabulary can be obtained manually using the Get RDF button on the appropriate page of the Registry.





The same serialization can be obtained automatically by de-referencing the URI of the element, element set, or value vocabulary.

## Unconstrained elements

# Extending a namespace

A namespace can be extended by refining the granularity of an element, by making the granularity of an element more coarse, or by adding an element.

An element may be a class or property from an element set, or a concept from a value vocabulary.

## Refinement

Refinement is the addition of an element that is more granular and narrower in its semantic scope. It is the opposite of coarsening.

For example:

A property for a general bibliographic note can be refined by adding a property for a specific type of note such as a note on printing history. The specific note property is linked to the general note property using the RDFS sub-property relationship:

ex:noteOnPrintingHistory rdfs:subPropertyOf ex:note

A class for persons can be refined by adding a class for a specific type of person such as fictitious persons. The specific type of person class is linked to the person class using the RDFS sub-class relationship:

ex:FictitiousPerson rdfs:sub-class ex:Person

A concept for the electronic resource type computer program can be refined by adding a concept for a specific type of program such as a computer macro. The specific type of program is linked to the general concept using the SKOS broader relationship:

ex:computerMacro skos:broader ex:computerProgram

## Coarsening

Coarsening is the addition of an element that is less granular and broader in its semantic scope. It is the opposite of refining.

### Scenario 1

A broader element is required for compatibility or interoperability; this is the opposite of simple refinement.

For example:

A property for a specific type of note such as a note on system requirements can be coarsened by adding a property for a more general note such as a note on access. The specific note property is linked to the general note property using the RDFS sub-property relationship:

ex:noteOnSystemRequirements rdfs:subPropertyOf ex:noteOnAccess

A class for persons can be coarsened by adding a broader class for agents. The person class is linked to the agent class using the RDFS sub-class relationship:

ex: Person rdfs:sub-class ex:Agent

A concept for the electronic resource type computer program can be coarsened by adding a concept for machine-readable information . The specific type of program is linked to the general concept using the SKOS broader relationship:

ex:computerProgram skos:broader ex:machineInfo

## Scenario 2

A property is required to aggregate a set of existing properties. This need often arises when developing an application profile.

## Scenario 3

An unconstrained property is required for interoperability with a property from another namespace.