# Specifying a linked data dataset for Europeana and aggregators

Linked data aggregation in project Europeana Common Culture

(version 0.2)

Nuno Freire (nuno.freire@europeana.eu)

Cultural Heritage Institutions that publish linked data typically publish data that covers more resources than the cultural heritage digital objects that they provide to Europeana. Especially for the case of their digital objects, where only a subset is intended for delivery to Europeana. Therefore, it is necessary that data providers make available linked data descriptions of the datasets for aggregation by Europeana, so that the specific data for Europeana is accurately specified in machine readable form.

Several vocabularies are available nowadays to describe datasets. Europeana supports three vocabularies which are suitable to fulfill the requirements for aggregation of linked data: VoID, DCAT, and Schema.org.

Data providers may use classes and properties from any of the three vocabularies to describe each of their datasets. To enable Europeana to aggregate and ingest a dataset, the linked data resource of the dataset should follow the following points:

- **Must** be accessible by its URI.
- **Must** be encoded in RDF.
- **Must** have a title property.
- **Must** specify the technical mechanism that allows the dataset to be automatically harvested by Europeana.
- **May** specify the most recent date on which the dataset was created or updated.
- **May** specify a machine readable license that applies to all metadata

The sections below provide details on how these points can be provided.

## Dataset RDF resource accessible by its URI

The description of the dataset in RDF must itself be published as linked data.

For ingesting the dataset in Europeana, this URI must be provided to Europeana. It will function as the entry point for the Europeana linked data crawler to reach all RDF descriptions of cultural heritage objects that form the dataset.

The dataset description is used during the first ingestion of the dataset into Europeana, and later, for its incremental updates.

The data provider should maintain the dataset description updated over time, if incremental updates of the dataset are done in the future Europeana.

## Dataset resource encoded in a supported RDF serialization

The Europeana LD Harvester accesses the RDF resource of the dataset by sending an HTTP request to the URI that includes the Accept header with the supported mime-types for RDF serialization. The response may use any of the following supported mime-types for sending the RDF description of the dataset:

| **Format** | **Mime-type** | **Specification** |
|---|---|---|
| RDF/XML | application/rdf+xml | https://www.w3.org/TR/rdf-syntax-grammar/ |
| Turtle | application/x-turtle or text/turtle | https://www.w3.org/TR/turtle/ |
| Notation3 (N3) | text/n3 | https://www.w3.org/TeamSubmission/n3/ |
| N-Triples | application/n-triples | https://www.w3.org/TR/n-triples/ |
| JSON-LD | application/ld+json | https://www.w3.org/TR/json-ld/ |

## Title of the dataset

The RDF resource of the dataset must have a title, and the title may be repeated for other languages. The titles should be provided by one of these properties: dcterms:title, schema:name. The language of the title should be represented in a xml:lang attribute of the title property.

## Specifying the technical mechanism for LD harvesting

A LD dataset for Europeana, is formed, in its core, by RDF descriptions of resources of the class edm:ProvidedCHO. In addition, the dataset contains all other resources used to describe the cultural object and aggregation metadata, as specified in EDM (i.e. resources of type ore:Aggregation, edm:WebResource, edm:Agent, etc.).

Descriptions of all these resources will be harvested by Europeana's LD harvester. The harvester will use the RDF description of the dataset to know which RDF resources to harvest and the mechanism to harvest them.

Data providers may choose one of the mechanisms, typically used for LD:

- Option A - Dataset distribution containing all data within the dataset.
- Option B - Listing of the URIs of all cultural objects's RDF resources in the dataset.
- Option C - SPARQL endpoint that can list the URIs of all cultural objects's RDF resources in the dataset.

The mechanism that should be applied to a LD dataset is indicated by the data provider in the properties of the RDF description of the dataset, using any of the supported vocabularies: VoID, DCAT, and Schema.org.

The following subsections provide details on how each of these mechanisms can be specified.

## Option A - Specifying a downloadable dataset distribution

All three vocabularies are capable of representing the required information for allowing Europeana to automatically obtain a dataset by downloading a distribution containing all data within the dataset.

The following table points to the most relevant parts of the vocabularies that specify how a dataset distribution can be represented.

| **Vocabulary** | **Specifications parts** |
|---|---|
| VoID | See section "3.3 RDF data dumps" describing the void:dataDump property. |
| DCAT | See section "6.7 Class: Distribution", particularly the properties dcat:downloadURL and dcat:mediaType. |
| Schema.org | see the definition of the property schema:distribution of the schema:Dataset class.<br>see also the class schema:DataDownload and its properties schema:contentUrl and schema:encodingFormat |

For the requirements of Europeana, when using dataset distributions, data providers must follow the following points:

- The files that constitute the data dump of the dataset, must contain the RDF data encoded in one the RDF encodings supported by Europeana: RDF/XML, Notation3, N-Triples, Turtle or JSON-LD.
- The files may be compressed. Currently, Europeana supports only the Zip and GZip compression algorithms.
- When using DCAT or Schema.org, the values of properties dcat:mediaType and schema:encodingFormat should only use mime-types supported by Europeana for RDF encoding: 'application/rdf+xml', 'text/n3', 'application/n-triples', 'application/x-turtle', or 'application/ld+son'.

Examples of dataset metadata with a downloadable distribution are shown in the section Example of a dataset available via a downloadable distribution.

## Option B - Specifying a listing of URIs

Only the VoID vocabulary includes a property to indicate an RDF resource that lists all the resources within a dataset.

VoID defines the property void:rootResource, that may be used by Europeana data providers to provide this information. See section "3.4 Root resources" describing the void:rootResource property, for the general use of the property.

For the requirements of Europeana, when using a listing of URIs, data providers must provide void:rootResource properties that contain the URIs of the cultural objects.

These URIs should point to RDF resources in EDM or in Schema.org. For linked data in EDM, the RDF resources must have one of the types ore:Aggregation, edm:ProvidedCHO. For data in Schema.org, these URIs should point to instances of schema:CreativeWork or one of its subclasses (e.g., schema:Painting, schema:Book, schema:Sculpture, etc.). The list of supported Schema:CreativeWork subclasses is documented in "Guidelines for providing and handling Schema.org metadata in compliance with Europeana".

Note that while DCAT does not offer the same features as VoID's listings of URIs, it is possible to add VoID void:rootResource statements to an existing DCAT description, which would then use the two vocabularies at once. It is also possible to connect a DCAT description of a dataset to a VoID file that would contain a set of void:rootResource statements, as a separate resource. This would be done using the "Loosely structured catalog" pattern, in which a dataset would be linked via dcterms:relation to a VoID file, described as a file that is itself said to comply (using dcterms:conformsTo) with the VoID standard (see this DXWG discussion). This pattern must however be refined so that a data consuming client knows for sure that the file contains a listing of root resources it can exploit. Since all this would in any case not remove the need for VoID statements, we have decided not to investigate this option further.

Examples of dataset metadata with listings of URIs are shown in the section Example of a dataset available via a listing of URIs.

## Option C - Specifying a SPARQL Service

VoID and DCAT provide ways to specify the endpoint URL of a SPARQL service that serves the dataset. Schema.org, however, is not able to accurately express all the required technical details.

Using VoID, the URL of the SPARQL endpoint is specified using the property void:sparqlEndpoint in the dataset RDF resource.

When using DCAT, there are two valid options to specify the URL of the SPARQL endpoint. The simplest option is to specify the URL in the dcat:Distribution using necessarily two properties: dcat:accessURL and dcterms:conformsTo. dcat:accessURL specifies the SPARQL endpoint URL, and dcterms:conformsTo specifies that the standard served by the dcat:accessURL is SPARQL.

The second DCAT option allows more details to be provided about the SPARQL endpoint, by describing it as a dcat:DataService, and reference in the dcat:Distribution the dcat:DataService with the dcat:accessService property. The RDF resource of the dcat:DataService, must have at least two properties that provide the same information as in the first option: dcat:endpointURL and dcterms:conformsTo.

Regarding the value to be used in the dcterms:conformsTo properties, the Europeana LD harvester recognizes the URL of W3C's recommendation SPARQL 1.1 Query Language (http://www.w3.org/TR/sparql11-query/).

Examples of dataset metadata with a sparql endpoint are shown in the section Example of a dataset available via a SPARQL endpoint.

### Specifying a subset of the SPARQL endpoint

In some cases, a SPARQL endpoint may contain additional data that is not part of the dataset intended for delivery to Europeana, that is, the dataset for Europeana is just a subset of all the data available in the SPARQL endpoint.

In these cases, the dataset distribution must specify a SPARQL query, in addition to the SPARQL endpoint URL. The query must have one variable in its SELECT part. The variable may have any name but there must be one, and only one variable in the SELECT part of the query. When executed in the SPARQL endpoint, the query result must be all the URIs of the RDF resources about the cultural heritage digital objects for Europeana.

Neither VoID nor DCAT define a modeling construct applicable to this case of specifying a subset using SPARQL queries. Given this, we have designed a solution for Europeana that defines a modeling construct reusing standard vocabularies: DCAT, Schema.org and the PROV Ontology (PROV).

This modeling construct elaborates on the second DCAT option described in the subsection above, where the SPARQL endpoint is represented as a dcat:DataService referred from a dcat:Distribution. When specifying a SPARQL query, the dcat:DataService is not referred directly from the dcat:Distribution, but instead, it is referred from a schema:SearchAction, in which the SPARQL query is specified. This schema:SearchAction is referred from the dcat:Distribution.

The properties that should be used to link the three objects are:

- A prov:wasGeneratedBy property links the dcat:Distribution to the schema:SearchAction.
- A prov:used property links the schema:SearchAction to the dcat:DataService.

Regarding the properties of the dcat:DataService, these should be the same we mentioned earlier (i.e., dcat:accessURL and dcterms:conformsTo) but it should also have one additional rdf:type property with the value prov:Entity, so that it can be a valid target of prov:used.

A property of the schema:SearchAction object should contain the SPARQL query. This should be a schema:query property, with a literal value containing the SPARQL query string.

An example using this modeling construct is shown in the last example of section Example of a dataset available via a SPARQL endpoint.

## Dataset level license

The RDF resource of the dataset may indicate a license that applies to the whole dataset. Specifying a license is optional since any provision of datasets to Europeana requires the agreement that the metadata can be licensed under CC0.

The license for a whole dataset should be specified in dcterms:license or schema:license (with Europeana supported licenses' URIs). Following the recommendations of DCAT, the property should be applied to the Distribution of the dataset. The property may also be applied

in the Dataset resource: this option may be required when specifying the dataset using VoID, where no Distribution resource exists.

Note that If the dataset provides the licensing information, individual metadata records may still override it, by specifying a license as defined in EDM. Nevertheless, Europeana requires the agreement that the metadata can be licensed under CC0.

# Examples

This section contains illustrative examples of RDF descriptions of datasets, prepared accordingly to the requirements of Europeana.

## Example of a dataset available via a downloadable distribution

The next example contains an RDF description of a dataset available via a downloadable distribution, using the DCAT vocabulary.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dcat="http://www.w3.org/ns/dcat#"
         xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description rdf:about="http://example.org/dataset/children_books">
      <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Dataset">
      <dcterms:title>Children books</dcterms:title>
      <dcat:distribution>
        <rdf:Description
rdf:about="http://example.org/dataset_distribution/children_books/">
          <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Distribution">
          <dcat:downloadURL
rdf:resource="http://example.org/downloads/our_dataset_2018-April.xml.gz"/>
          <dcat:mediaType>application/rdf+xml</dcat:mediaType>
          <dcterms:license
rdf:resource="http://creativecommons.org/publicdomain/mark/1.0/"/>
        </rdf:Description>
      </dcat:distribution>
  </rdf:Description>
</rdf:RDF>
```

The next example contains the description of the same dataset available via a downloadable distribution, using the Schema.org vocabulary.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:schema="http://schema.org/">
  <rdf:Description rdf:about="http://example.org/dataset/children_books">
      <rdf:type rdf:resource="http://schema.org/Dataset">
      <schema:name>Children books</schema:name>
      <schema:distribution>
```

```
        <rdf:Description
rdf:about="http://example.org/dataset_distribution/children_books/">
          <rdf:type rdf:resource="http://schema.org/DataDownload">
          <schema:contentUrl
rdf:resource="http://example.org/downloads/our_dataset_2018-April.xml.gz"/>
          <schema:encodingFormat>application/rdf+xml</schema:encodingFormat>
          <schema:license
rdf:resource="http://creativecommons.org/publicdomain/mark/1.0/"/>
        </rdf:Description>
      </schema:distribution>
  </rdf:Description>
</rdf:RDF>
```

The next example contains the description of the same dataset available via a downloadable distribution, using the VoID vocabulary.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:void="http://rdfs.org/ns/void#"
         xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description rdf:about="http://example.org/dataset/children_books">
      <rdf:type rdf:resource="http://rdfs.org/ns/void#Dataset">
      <dcterms:title>Children books</dcterms:title>
      <void:void:dataDump
rdf:about="http://example.org/downloads/our_dataset_2018-April.xml.gz"/>
      <dcterms:license
rdf:resource="http://creativecommons.org/publicdomain/mark/1.0/"/>
  </rdf:Description>
</rdf:RDF>
```

## Example of a dataset available via a listing of URIs

The next example contains an RDF description of a dataset available via a listing of URIs. In this example, using the VoID vocabulary.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:void="http://rdfs.org/ns/void#"
         xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description rdf:about="http://example.org/dataset/children_books">
      <rdf:type rdf:resource="http://rdfs.org/ns/void#Dataset">
      <dcterms:title>Children books</dc:title>
      <dcterms:license
rdf:resource="http://creativecommons.org/publicdomain/mark/1.0/"/>
      <void:rootResource rdf:about="http://example.org/Aggregation/cho_abc"/>
      <void:rootResource rdf:about="http://example.org/Aggregation/cho_def"/>
      <void:rootResource rdf:about="http://example.org/Aggregation/cho_zyz"/>
```

```
  </rdf:Description>
</rdf:RDF>
```

## Example of a dataset available via a SPARQL endpoint

The next example contains an RDF description of a dataset available via a SPARQL endpoint, using the VoID vocabulary.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dcat="http://www.w3.org/ns/dcat#"
         xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description rdf:about="http://example.org/dataset/fennica">
      <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Dataset">
      <dcterms:title>Fennica - The Finnish National
                     Bibliography</dcterms:title>
      <void:sparqlEndpoint
                rdf:resource="http://data.nationallibrary.fi/bib/sparql"/>
  </rdf:Description>
</rdf:RDF>
```

The next example contains the description of the same dataset available via a SPARQL endpoint, using the DCAT vocabulary.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dcat="http://www.w3.org/ns/dcat#"
         xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description rdf:about="http://example.org/dataset/fennica">
      <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Dataset">
      <dcterms:title>Fennica - The Finnish National
                     Bibliography</dcterms:title>
      <dcat:distribution>
        <rdf:Description
             rdf:about="http://example.org/dataset_distribution/fennica/">
          <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Distribution">
          <dcat:dataService>
              <rdf:Description
                     rdf:about="http://example.org/data_service/finna/">
                  <rdf:type rdf:resource="dcat:Dataservice">
                  <dcat:endpointURL
                    rdf:resource="http://data.nationallibrary.fi/bib/sparql"/>
                  <dcterms:conformsTo
            rdf:resource="http://www.w3.org/TR/sparql11-query/"/>
                </rdf:Description>
          </dcat:dataService>
          <dcterms:license
```

```
            rdf:resource="http://creativecommons.org/publicdomain/mark/1.0/"/>
        </rdf:Description>
      </dcat:distribution>
  </rdf:Description>
</rdf:RDF>
```

The next example contains an RDF description of a dataset available via a SPARQL endpoint, for those cases where the dataset is a partial subset of the data available in the SPARQL endpoint. This example extends the previous one, which is using the DCAT vocabulary, with Schema.org and the PROV Ontology.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dcat="http://www.w3.org/ns/dcat#"
         xmlns:dcterms="http://purl.org/dc/terms/">
  <rdf:Description rdf:about="http://example.org/dataset/fennica">
      <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Dataset">
      <dcterms:title>Fennica - The Finnish National
                     Bibliography</dcterms:title>
      <dcat:distribution>
        <rdf:Description
             rdf:about="http://example.org/dataset_distribution/fennica/">
          <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Distribution">
          <dcterms:license
            rdf:resource="http://creativecommons.org/publicdomain/mark/1.0/"/>
          <prov:wasGeneratedBy>
            <rdf:Description>
              <rdf:type rdf:resource="schema:SearchAction">
              <schema:query>PREFIX schema: &lt;http://schema.org/&gt;. SELECT
?uri WHERE { ?uri a schema:CreativeWork. ?uri schema:url ?url }</schema:query>
              <prov:used>
                <rdf:Description
                     rdf:about=”http://example.org/data_service/finna/”>
                  <rdf:type rdf:resource="dcat:Dataservice">
                  <rdf:type rdf:resource="prov:Entity">
                  <dcat:endpointURL
                    rdf:resource=”http://data.nationallibrary.fi/bib/sparql”/>
                  <dcterms:conformsTo
            rdf:resource=”http://www.w3.org/TR/sparql11-query/”/>
                </rdf:Description>
              </prov:used>
            </rdf:Description>
          </prov:wasGeneratedBy>
        </rdf:Description>
      </dcat:distribution>
  </rdf:Description>
</rdf:RDF>
```