Exercise:

RDF Authority Data for Homegrown Reconciliation Service

**Summary**

In this exercise, you’ll convert name authority data in tabular format to RDF, for the purposes of name reconciliation in our TEI demo dataset of Ambrose Bierce letters. Using OpenRefine with RDF plugin, we can clean/adjust name authority data before expressing as RDF.

**Dataset**

*exercise\_nameAuthority\_uri.csv*

|  |  |
| --- | --- |
| Name | URI |
| Bierce, Ambrose, 1842-1914? | http://viaf.org/viaf/232144783035948358100/ |
| Walsh, Elizabeth, 1872-1895 | http://dummyuri/5431789082754235 |
| Walsh, Myles, 1874-1968 | http://dummyuri/5431789082754236 |

**Basic RDF triple**

Predicate

Our RDF structure should look something like this:

|  |
| --- |
| <http://dummyuri/5431789082754235/> ⇒ RDF:type ⇒ foaf:person  <http://dummyuri/5431789082754235/> ⇒ foaf:name ⇒ “Walsh, Elizabeth” |

**Exercise**

First, we will prep the OpenRefine project with the data and vocabularies needed to express these authorities as RDF.

1. Download the dataset and import into a new OpenRefine project. Accept all of the file import defaults and *Create Project.*
2. Split the name field into two columns to separate names and birth-death range.

We need to identify which vocabularies and schemas we will be using in our RDF. We do this by declaring the URIs for the various vocabularies we want to use and assigning them to a prefix at the beginning of the document. The RDF plugin for OpenRefine allows us to select prefixes and automatically includes the declarations on export.

1. Beside *Extensions,* click the *RDF* dropdown and select *Edit RDF Skeleton*
2. Edit Base URI to something like `http://localhost/`
3. Click *add prefix,* next to *Available Prefixes*
4. In *New Prefix* dialog, add `foaf` to prefix field and the URI field should auto populate. Click `Ok` and `foaf` should now appear under available prefixes
5. Follow the steps from above to add the `rdf` AND `rdfs` prefixes.

Now we can begin to fill out the RDF skeleton: we align the fields from the authorities csv with appropriate RDF terms. We’ll be using RDF to make machine readable statements about each entry in the authorities csv. Within RDF we use text or URI to represent concepts. In this example, we’ll be using both. We use the **foaf:person** term to identify each entry as a person. Further, we’ll use the **foaf:name** term to point to the text of each name, to associate the name with the person object.

1. In the main area of the skeleton editor, the dialog will include a default skeleton that we will modify
2. To the left, click *ID* and select the *URI* radio button from *Use content from cell*
3. Select *URI* radio button under *The cell’s content is used as...*
4. Click OK (figure 1)

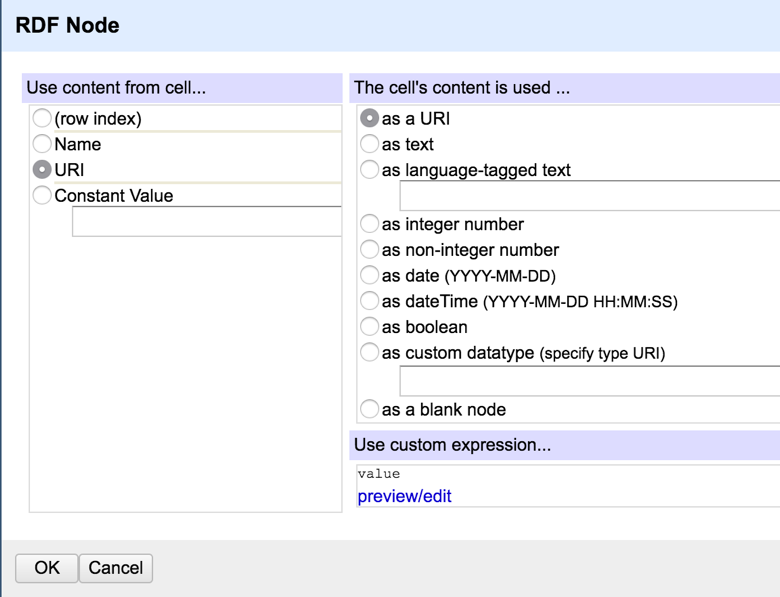


Figure 1 Your final configuration should look like this.

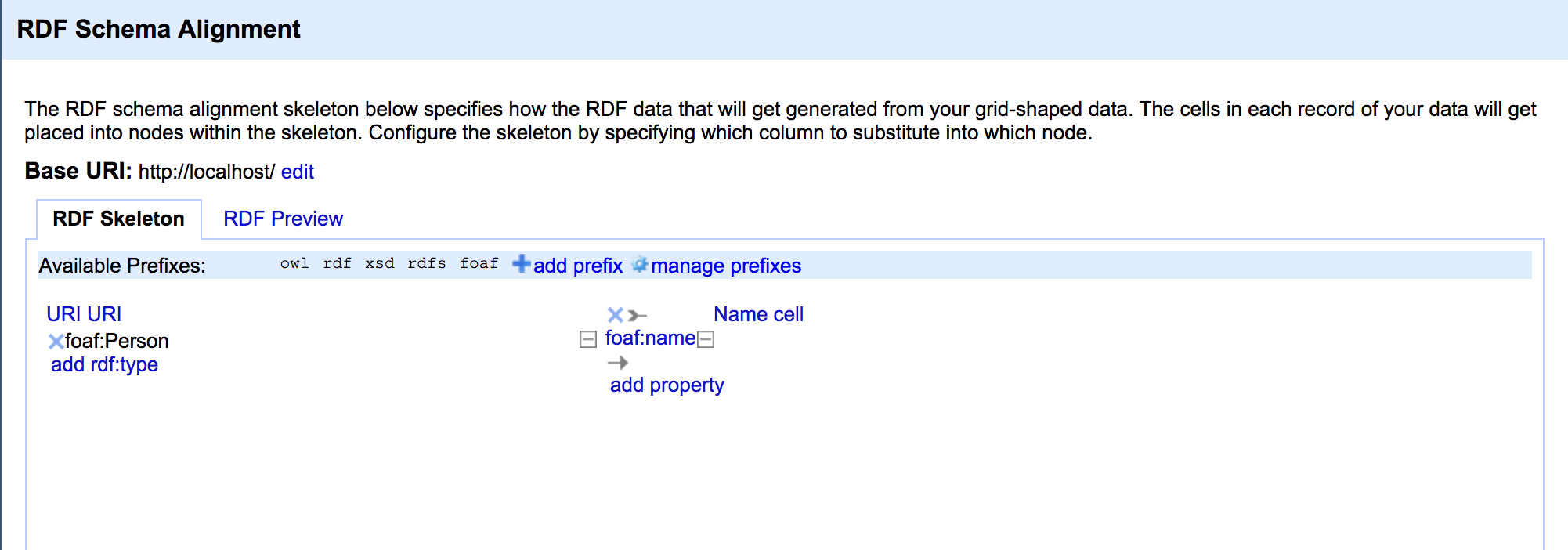
1. On the left of the RDF skeleton, under `URI`, click *add rdf:type* and type “foaf:person” into *search for property* and select or hit enter to apply (if you don’t see it as an option in the dropdown, press shift+enter to add it.)
2. On the right side of the RDF skeleton, click all of the little blue x’s to remove the default properties.
3. On the right side of the RDF skeleton, click *property?* and type “foaf:person” into *search for property* and select or hit enter to apply (if you don’t see it as an option in the dropdown, press shift+enter to add it.)
4. On the right side of the RDF skeleton, click *Configure?*
5. Select the *Name* radio button under *Use content from cell...* and select the *as text* radio button under *The cell’s content is used...*
6. Click OK (figure 2) \*Toggle the RDF Preview tab if OK button not showing.

Figure - Example RDF schema alignment

Now that we have defined the RDF schema for our data, we can use the RDF plugin to export for use in other applications. The RDF plugin allows to export either as RDF/XML or Turtle (ttl) format. We will export in turtle and use the file in another exercise for data reconciliation (also with the RDF OpenRefine plugin.

1. Click the *Export* dropdown and select *RDF as Turtle*
2. This should open a new tab with the text; copy and paste the text into a text editor and save with .ttl file extension (figure 3)

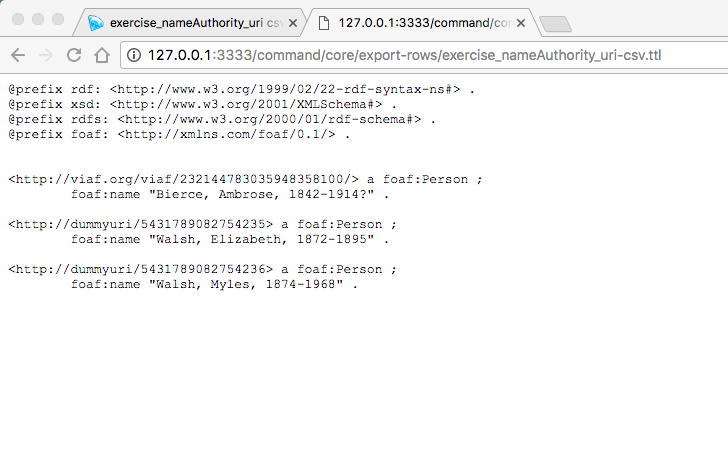


Figure 3 - Turtle output