

XSLT Ingest Example: Make URIs

Make URIs

[Start](#) [Previous](#) [Next](#)

This step produces a pair of XML files that are used to provide unique numbers (UNOs) from which we can construct URIs for the URPs and UROs just counted. The unique numbers were generated by a Perl utility, developed by the author, called **nuno** (included in the **example/bin** directory) that uses an 8 character hexadecimal counter, an Inode number, a Unix user id and a token to ensure uniqueness. The counter is stored in a UNIX file and so can be protected by UNIX security. The Inode number ensures that if the file is moved or copied the unique number sequence is different from the original sequence. The UNIX user id is the ID of the utility caller (not the owner of the counter file) this ensures that if there are multiple users of the utility on the same counter file then each user will obtain sequence different from all other users. The token mentioned above is useful when trying to SPARQL query for individuals from a specific source. In our case this is 'EX' standing for 'Example'.

The calls to the nuno utility are shown next.

```
bin/nuno -t EX- -X -n 11 > EX-URP-UNOs.xml
bin/nuno -t EX- -X -n 5  > EX-URO-UNOs.xml
```

An example output file of the first command, EX-URP-UNOs.xml is shown next (the second file is similar).

```
<?xml version='1.0'?>
<Mapping>
<map n='1' nuno='EX-0203EF6807A00000000' />
<map n='2' nuno='EX-0203EF6807A00000001' />
<map n='3' nuno='EX-0203EF6807A00000002' />
<map n='4' nuno='EX-0203EF6807A00000003' />
...
</Mapping>
```

The attribute **n** will be used to select which URI is assigned when creating new **foaf:Persons** as described in the next section. In our example, the following URIs will be assigned to the URPs.

```
http://vivo.cornell.edu/individual/EX-0203EF6807A00000000
http://vivo.cornell.edu/individual/EX-0203EF6807A00000001
http://vivo.cornell.edu/individual/EX-0203EF6807A00000002
http://vivo.cornell.edu/individual/EX-0203EF6807A00000003 ...
```

It is worth noting that your data source may guarantee that a particular data element *is always present and is always uniquely associated with a single person*. If this is the case, then that element would be appropriate for use as a local name in your URIs provided it also satisfies the URI formation rules. Suppose that we had such guarantees in our example in the case of **NETID**. This would mean that a URI for 'Arthur R. Fuller' (arf72) like that shown next could be assigned during the Gather step.

```
http://vivo.cornell.edu/individual/arf72
```

Unfortunately we can't do this since the **NETID** element might be empty or missing in our source data. While a **NETID** is uniquely associated with a person we are not assured by the source that one will always be present in each record. Hence we will still have to create URIs from scratch.

[Start](#) [Previous](#) [Next](#)