Data types for string and language

VIVO 1.10 implements RDF 1.1 and Jena 3. These changes impact the datatypes for strings and the use of the lang tag to indicate the language of the string. Please read these recommendations carefully. Jena2tools and Jena3tools will convert from previous representations to the representations recommended here.

Literal Values

Jena 3 improves Jena's RDF 1.1 compatibility. Specifically, literal values are always stored internally with datatypes. "Untyped" string literals are the same as the identical value typed as xsd:string. See the following document for more information

> https://jena.apache.org/documentation/migrate_jena2_jena3.html

For VIVO, this means that the two triples:

```
<subj> <pred> "value"
<subj> <pred> "value"^^xsd:string
```

will be treated as the same triple and only stored once in your triple store. As a result, when using the procedure described below to upgrade your triple store from an earlier version of VIVO, you may find that the number of triples in your triple store after the upgrade is lower than the number before the upgrade.

Any code, tools, parsers, utilities, or queries that expect to differentiate between these two triples will produce results different than produced previously – RDF 1.1 no longer distinguishes between these two triples. In particular, queries that limit results based on the xsd:string datatype will likely produce larger result sets, as previously untyped triples are now typed as xsd:string internally.

Another way of saying this – any triple loaded into VIVO or Vitro that does not have a type will be typed as xsd:string internally.

Exports from VIVO and Vitro will never include the xsd:string datatype. Literal values that do not have explicit types are always assumed to be xsd:string.

As a result, we recommend that input process for VIVO do not include xsd:string datatypes on literals. While they may be correct, and will result in the literal value being typed as xsd:string internally, export processes will not include the xsd:string on output.

In RDF 1.0, a type could not be asserted with a language identifier. In RDF 1.1, a type can be asserted with a language identifier. Untyped input with language identifiers were left as untyped internally in RDF 1.0. In RDF 1.1, untyped input with language identifiers are assumed to have type rdf:langString. Exports from VIVO for triples with language tags will never include the rdf:langString datatype. Literal values with language tags are always assumed to be rdf:langString.

As a result, we recommend that input process for VIVO do not include datatypes on triples with language types. While asserting rdf:langString is correct, and will result in the literal value being typed as rdf:langString internally, export processes will not include the rdf:langString on output.

Code, tools, parsers, utilities based on RDF 1.0 should not be used with Vitro and VIVO 1.10.x All code, tools, parsers, and utilities must support RDF 1.1.

On start-up of version 1.10.x, the triple store is checked to insure that it has been upgraded. If untyped literals are found in the triple store, an error message will appear in the browser and the application will not start. The test applies only to the content store. It is possible that your content store could pass this test, but your configuration triple store remains incompatible with Jena 3 and RDF 1.1. In such a case, your application may become unstable.

Recommendations

1. String literals should be untyped in RDF input to VIVO. Use "xxx" rather than "xxx"^^xsd:string
2. Lang tags should be used with untyped string literals in RDF input to VIVO. Use "xxx"@fr-CA rather than "xxx"@fr-CA^^rdf:langString
3. Lang tags should be used on all strings which might render differently in different languages. Use "United States"@en-US. For strings which are not rendered differently in different languages use a simple untyped string literal. For example "0000-0001-2345-321X"