

A Generalizable, XSLT Based RDF Ingest Example

Author: Joseph R. McEnerney (jrm424@cornell.edu)

The purpose of this example is to illustrate XSLT based techniques that have been used successfully to ingest data from more than a dozen sources into VIVO at Cornell. Instead of a simplified 'toy' example, the source data used will display many of the data quality problems often found in practice. The goal is to transform this source data into RDF that conforms to a specific data model and can be loaded into a VIVO instance. The example is based on educational credentials and the central objects in the RDF data model are instances of the class **vivo:EducationalTraining**. In addition, we want to prevent duplication of Person and Organization RDF. Experience has taught us that this XSLT transform methodology performs well in terms of processing time and is scalable to tens of thousands of source data records.

- [Download full paper](#)
- [Download example code and data](#)
- [Download poster from 2013 VIVO Conference](#)
- [Flowchart](#) for this example

Assumptions

This example expects the reader to have some familiarity with the following notions and technologies:

1. VIVO via [this wiki](#), vivoweb.org, and the [VIVO Project on GitHub](#)
2. [XML](#), [RDF](#)
3. [XSLT 2.0 \(Saxon 9.x he\)](#), [XPath 2.0](#)
4. [SPARQL 1.1](#)
5. [Unix/Linux Bash Shell Command Line](#)

Use the following links to get to the various sections of this paper.

- [The Source Data](#)
- [The Accumulator Classes](#)
- [The Process](#)
- [Gather](#)
- [Count](#)
- [Make URIs](#)
- [Create New Persons and Organizations](#)
- [Fill in URPs and UROs](#)
- [Create RDF and add to VIVO](#)
- [Final Considerations](#)

Appendices

- [Appendix A: Example Directory Layout](#)
- [Appendix B: Bash Commands to Execute Example Code](#)
- [Appendix C: The Example Source Data XML](#)
- [Appendix D: The Cumulative Sum Recursive Template](#)
- [Appendix E: The Sparql Construct and XSLT for Per0.xml](#)
- [Appendix F: Suggested Reading](#)

[Next](#)