

XSLT Ingest Example: Final Considerations

Final Considerations

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

It is unlikely that you will use this code 'as is'. Your data source is likely to be very different and possibly much more complex. In fact this is the case for many of the data sources ingested here at Cornell. Consider, for example, a data source of Academic Articles where there are typically several co-authors listed with each article and since many co-authors are not at Cornell, missing netids is a commonplace. However these techniques easily extend to handle such cases. Until now no mention has been made about how to update information once in VIVO. Two possibilities suggest themselves:

- Wholesale replacement and
- Incremental update

Both have advantages and disadvantages depending on how complex the source is. Wholesale replacement is attractive because all you have to do is retract what was once asserted; build new RDF which is then asserted. If you keep (i.e. don't retract) the new people and organization RDF then chances are they will be available in [Per0.xml](#) or [Org0.xml](#) when needed. On the other hand, wholesale replacement can be very time consuming. To replace the many tens of thousands of Academic Articles in VIVO at Cornell would take several hours just to retract. However, for small sources wholesale replacement is a good choice.

Incremental update involves a differencing process where the source data for one query result set is compared to the data for the next. Differences in terms of additions, changes and deletes are tabulated to determine what must be retracted (deletes and changes) and what must be asserted (changes and additions). The granularity of the differencing process depends on the complexity of the source. Deeply nested XML structures can be tough to compare. Look for 'last modified dates' and similar elements, that signal a change, to simplify the problem.

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