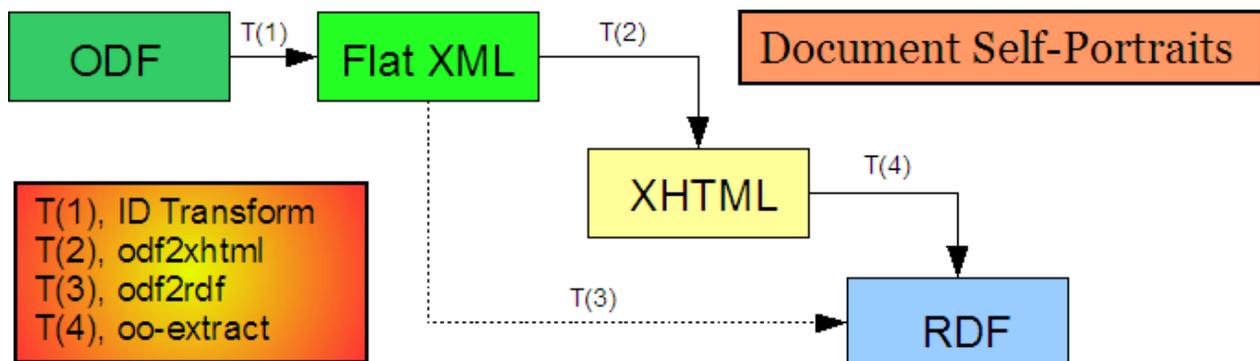


The ODF2XHTML Export Filter

Modified To Use Dublin Core Metadata Initiative (DCMI) Terms.

The transforms were done with Saxon. The export filter does not recognize MSXML node sets. The GRDDL transform needs no "node-set" function and instead includes the RDF Schema as an informal check on the accessibility of Resource Descriptions which should be found.

Modifications were necessary to only one template in header.xsl (aside from the set-up work mentioned below). Techniques for the propagation of meta data in the <body /> of XHTML documents [require other protocols not covered by this contribution](#). The best format to describe document content is not the best format to describe document meta data. Nonetheless, an open source resource like ODF should be able to furnish a "self-portrait" in an ascetically pleasing way, for human consumption, and an efficient way for classification, cataloging, searching etc..



When there are no references in the <body> of the document, the ODF to RDF direct transformation [ODF » T(1)|T(3) » RDF] and the chained transformation via XHTML [ODF » T(1)|T(2) » XHTML » T(4) » RDF] have the same results. There can be only one profile, and the transforms assure that missing meta data is exposed as empty nodes.

The case where the resulting RDF representations are the same is appropriate for collaborative writing, wiki's and so forth where all meta data is public, and shared. A collection of RDF Description Sets can be made available to all, while presentation (XHTML) remains under control of the individual ODF file holder.

The T(4) transform is based on the **G**leaning **R**esource **D**escriptions from **D**ialects of **L**anguages recommendation from the W3C. It predates the DCMI Abstract Model (DCAM) and has been updated (oo-extract.xsl) .

There was a small amount of set-up work – see Appendix 1.

Sample Files

- SampleXHTML.odt
- meta.xml (from the odt package)

- oo-extract.xsl + DCMI RDFS files (T(4))
- [dc-extract.xsl](#) (W3C GRDDL, use oo-extract.xsl instead)
- header.xsl (modified part of T(2))
- flat-plus-ns.xsl (T(1))
- odf2xhtml.xsl (T(3))

Written Files

- SampleXHTML.flt (made with an ID Filter for external (Java) parser use as source)
- SampleXHTML.pdf (to check the “Document Properties”)

- SampleXHTML.old.xhtml (original transform)
- SampleXHTML.dcmi.xhtml (ODF « T(2))
- SampleXHTML.dcmi.rdf (SampleXHTML.dcmi.xhtml « T(4))

- SampleXHTML.rdf (ODF « T(3))

- sample output of W3C RDF Validator (Triples and Graph)

Profile			
ODF Meta	XHTML	RDF	Scheme Comments
-	content-type	-	http-equiv
meta:generator	generator	DCTERMS.source	See DCMI Definition
meta:initial-creator	author	DCTERMS.creator	
meta:creation-date	created	DCTERMS.issued	DCTERMS.W3CDTF
dc:creator	changedby	DCTERMS.contributor	
dc:date	changed	DCTERMS.modified	DCTERMS.W3CDTF
dc:subject + meta:keyword(s)	subject	DCTERMS.subject	xml:lang
dc:description	description	DCTERMS.description	xml:lang
dc:title	title 	DCTERMS.title	xml:lang
dc:language	-	DCTERMS.language	DCTERMS.RFC4646
meta:printed-by + meta:print-date + dc:language 	-	DCTERMS.provenance with [dc:publisher] [dc:date] [dc:language]	[dc:publisher] is ideally a contact point, a mailto: link or email address, but may be a name of a person or organization. xml:lang
	-	DCTERMS.identifier	DCTERMS.URI

-  Since a resource requires an identifier (URI) and location (URL), the base (ODF.base) and file name (ODF.filename) are required in user defined fields.
-  As a literal. This may be enhanced in future versions to use a DCTERMS.Agent class, for example with FOAF etc..
-  XHTML, like HTML requires a title to be valid. Both <title> and DCTERMS.title are written, and identical in content, but the <title> is ignored by GRDDL transforms because it is unqualified, in the XHTML name space.

In keeping with DCMI conventions, “keywords” are taken to be subject headings, and are exposed as a semicolon delimited list. Alternatively, these could be exposed as individual <subject> nodes, or the list merged with (an XPATH of) registered Controlled Vocabularies with a sub-type <subject>, e.g. MeSH, LCC, TGN etc., depending upon resource classification requirements.

The two transformation pathways require different <source> or generator results per DCMI conventions. The <provenance> node differentiates the act of printing or publishing from the act of modification or editing.

“Extra” meta data *elements* such as rights, other authors, other contributing editors etc. can be defined in user defined fields with the proper HTML syntax DCMI Property Name (see meta.xml). Some caution is necessary because

<meta> is plain/flat text and in scope applies to the entire document. If this specificity is inadequate, then there may be no choice but to resort to [more complex encoding](#).

Encoding meta data in the body of a document yields unlimited storage and extensibility, the problem is that meta data is normally not visible and can clutter displays. Beyond the clutter lies a much deeper problem, it is not enough to use style to hide meta data in the <body> of a document when the real intent is to redact (delete) it – see **Pay No Attention Redux** below.

Appendix 1

Errors

1. body.xsl – variable defined twice line 489 – comment out second declaration
2. DOM – dc:language fails to propagate in “flat” ODF. This may be peculiar to OpenOffice. At some point, it will be fixed since the ODF specification is unambiguous.

Irritations

1. Includes xforms and xsd/xsi namespaces – modified “exclude-result-prefixes”. It could be that these name spaces are necessary (along with xlink and mathML). If that is the case, then XHTML 1.1 should be used for DTD validation. Schema validation avoids these problems, but you have to roll your own for XHTML 1.0 (XSD Schema included as a courtesy to those who actually read the notes – there is no official W3C version).
2. “indent” attribute set to “yes”
3. <head> profile attribute set to “http://dublincore.org/documents/dcmi-terms/”[Property URI]
4. XHTML 1.0 (net, DTD) validation turned off. Validation is expensive, and of lesser value outside a development environment. Name space conflicts assure that there is no “one size fits all” syntax. The browsers wars are behind us for the most part, but there are still a few *partisans* in high places.

Pay no attention to the ManBehindTheCurtain

To validate files with the XHTML 1.0 XSD Schema, the XSI name space must be used. This will, of course keep the file from validating against the PUBLIC DTD from the W3C. Dublin Core has a special syntax for XHTML/XML <meta> nodes so that DTD validation is not disrupted – but that does not mean the meta data is validated because the data is in the XHTML name space. This is no mere technicality for the same reason RDF in XML is not quite an RDF Graph.

Two schema files are necessary: xml.xsd, and xhtml1-strict.xsd

As well as changes to the root <html> node.

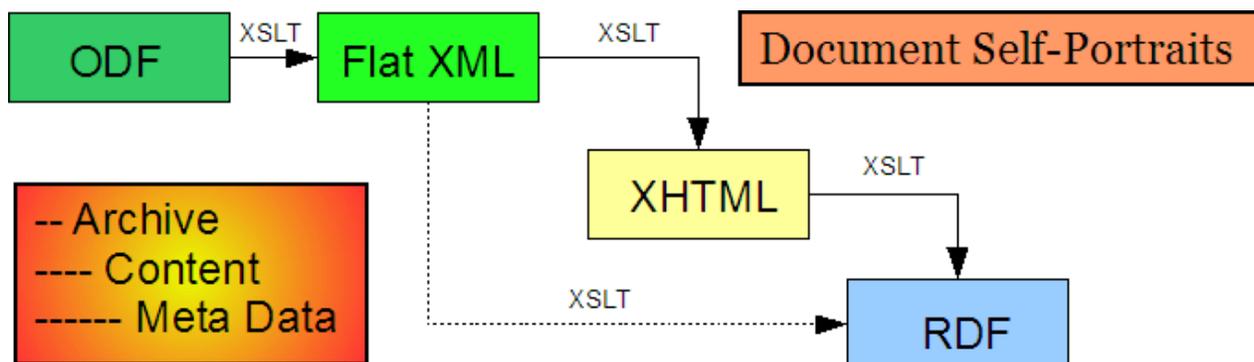
```
<?xml version="1.0" encoding="UTF-8"?>
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en"
  lang="en"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/1999/xhtml ./xhtml1-strict.xsd">
...

```

These files will validate with xerces (schema only) or even MSXML.dll with a suitable wrapper, but you will have to trust me or do it yourself. Validation, when required, is too important to be left to experts at user friendliness.

Pay No Attention Redux

This is derivative work, not part of the contribution provided for explanatory reasons only.



The case where data is not meant to be public and the result of the direct transformation should not be the same as the chain is what the author calls **Redact Unless Static Text Technology**. Although not useful for scholarly work, there are certainly times when data must be masked for privacy reasons and to prevent Identity Theft. In this case, the two modes (different XSLT paths) can be used to control propagation of private data. Moreover, by strict access control to the source ODF or archive flat ODF the private data is masked automatically – an additional level of control. With these two levels of access control, those who must collect Personally Identifiable Information (PII) can choose storage and retention options *at the document level*.

Governments, at all levels, provide examples of the utility of this approach. The difficulty, in the past, has been that the cost of maintaining large domains and systems has prohibited such fine, local control. The key to regulation of the domain is a [uniform system of tokens](#) which although inserted at the document level, do not cause the domain systems to fail. Specific requests from a Federal Authority can be answered with statistical compilations rather than raw data. While a grasping Federal Authority probably would not see this as an advantage over access to raw data, in fact it is an advantage because Confidence Levels can be used to measure compliance between federation membership. Non-compliance should be seen for the political decision it is rather than the fault of Collector(s) – or worse, Collectors making political decisions they are not entitled to make.

The Private Sector can benefit as well. The major problem there is that meta data aggregation causes insurmountable time-shifting concerns for mass marketing and in turn a temptation to store an ever increasing amount of information which may not contribute to the marketing effort at all. For example,

if the phone company were to once acquire your shoe size they require a simple mechanism to [maintain a reference](#) to the acquisition, time, date etc. to live on in their systems even though the information itself is discarded as non-useful. The alternative, collecting and maintaining a data base of everybody's shoe size, is insane.