Meta Data Round-Tripping & Privacy

Application User Data, Address Books and Web Forms

- Gannon J. Dick (gannon_dick@yahoo.com)

Foreword

The trouble with meta data round trips between formats is the same problem with meta data round trips between "displays" which may hide meta data from view. While the results are common in real life, a similar name on a "no fly" list causing a missed flight (a mis-identification), or a targeted advertisementt way off target (a *mis* –re-identificationn), the root cause is simply that a format conversion confuses the distinction between "exposed" and "unseen". In the case of re-identification, the issue comes up as an *identifier deletion* when a *data redaction* was intended.

The **CSVplus** Data Base is a generalized model of a CSV Format Spreadsheet. Fully exposed Meta data elements are added to the CSV Format to prevent the unintentional deletion of identifiers.

While this format has an obvious application for Application User Data and Address Books, it can also be used on other data sets, for example, Web Forms or Social Networking Site Profiles, or even Paper Forms.

Data View (Spreadsheet View)

…	**…**	…	…	…	…	…
PropName2	**-2**	…	…	…	…	…
PropName1	**-1**	A	B	C	D	…
PropName0	**0**	ColNameA	ColNameB	ColNameC	ColNameD	…
Property	**Sequence**	**ColNameA**	**ColNameB**	**ColNameC**	**ColNameD**	**…**
		A	**B**	**C**	**D**	**[…]**
	1	ColNameA	ColNameB	ColNameC	ColNameD	…
	2	dataA1	dataB1	dataC1	dataD1	…
	3	dataA2	[null]	[null]	dataD2	…
	…	…	…	…	…	…

The CSV format has been expanded to contain several more sets (rows) of column properties to enable an automation of SQL scripts, and generation of an common web formats from **CSVplus** files.

Meta Data Components

A **CSVplus** table contains rows for a model and a reference data set. For flexibility, the model fields contain a row of primary and alternate names as well as an "all" field which is the union. In the simplest configuration, all fields are primary. The reference data set has primary fields as fields matching the model and alternates set to the model field name. The model acts as a filter for the data set, by field.

In this configuration, a ***Round Trip Data Element*** is defined as any two Primary Column Names which occur in the same (either Primary or Auxiliary) Model Column. Any Auxiliary Column Names used will prevent a Round Trip either on import or export (and we do not care much which one). The number of Round Trip Data Elements can only be increased (maybe) by adding Model Columns. Yet, since these additional columns must have been matched from unmatched names in the original sets it may be that there are simply no corresponding primary names in each set. The Primary/Auxiliary designation of a Model Column has no effect and can be used for experimentation with the optimum components of the model.

If two or more data sets "round trip" with the same column in the model, then the data round trips with each, by name.

For example:

Ds1.Name	**Ds1.AltName**		**Model (Union)**		**Ds2.Name**	**Ds2AltName**
FOOT_SIZE			shoeSize		size-of-shoe	
firstName			firstName			firstName
	lastName		lastName		Surname	
	Country		Country			Country

Only (FOOT = size-of-shoe) is true. However there is some wiggle room here. Note that linear combinations can be made to work.

Ds1.Name	**Ds1.AltName**		**Model (Union)**		**Ds2.Name**	**Ds2AltName**
FOOT_SIZE			shoeSize		size-of-shoe	
firstName			*firstName*			firstName
	lastName		*lastName*		Surname	
	Country		Country			Country

With a strict attention to parsing, a new *Name* (=firstName+lastName) element can be added to the model. The result is an additional "round trip".

The Data Base

Changes to the Card Data Base

The model now includes logic for a more fine tuned redaction (4 cases). While each case is computed, it is possible to collapse the result set by manipulation of the meta data properties (ID <= 0). The effect of the "Redaction TOKEN" is the same, if present the cell will contain the TOKEN. However, if the cell was NULL to begin with, it will contain the Redaction NULL TOKEN or the NULL TOKEN.

- ➔ When all three TOKENS are NULL valued, this is Deletion, which, for reasons stated elsewhere, is not acceptable. A "re-identifier" is free to fill in the blanks.
- ➔ If the Redaction TOKEN is equal to the Redaction NULL TOKEN, then the entire column will be populated with the Redaction Tokens – **RUST**ed (**R**edact **U**nless **S**tatic **T**ext). The entire spreadsheet will be populated to prevent "re-identification".
- ➔ If the NULL TOKEN is equal to the Redaction NULL TOKEN then redacted columns will show a mixture of Redaction Tokens and NULL Tokens and non-redacted columns will show a mixture of cell values and NULL Tokens. The entire spreadsheet will be populated to prevent "re-identification".

Spreadsheet View

```
CASE
   WHEN (Cell Value IS NOT NULL) THEN (Cell Value)        // result (Cell Value)'
   WHEN (Cell Value IS NULL) THEN NULL                    // result NULL
   WHEN (Cell Value IS NULL) THEN NULL                    // never executed
   ELSE (Cell Value)                                      // never executed
END AS Column Name
```

Redaction

```
CASE
   WHEN (Cell Value IS NOT NULL)
        AND (Redaction TOKEN IS NOT NULL) THEN (Redaction TOKEN)       // result '[redacted]'
   WHEN (Cell Value IS NULL)
        AND (Redaction TOKEN IS NOT NULL) THEN (Redaction NULL TOKEN)  // result '[null]'
   WHEN (Cell Value IS NULL) THEN (NULL TOKEN)                         // result '[fallback]'
   ELSE (Cell Value)                                                   // result (Cell Value)
 END AS Column Name
```

Deletion

```
CASE
   WHEN (Cell Value IS NOT NULL)
        AND (Redaction TOKEN IS NOT NULL) THEN NULL                    // result NULL
   WHEN (Cell Value IS NULL)
        AND (Redaction TOKEN IS NOT NULL) THEN NULL                    // result NULL
   WHEN (Cell Value IS NULL) THEN NULL                                 // result NULL
   ELSE (Cell Value)                                                   // result (Cell Value)
END AS Column Name
```

The *PropName* list element has been expanded to (in DTD notation):

<!ELEMENT PropertyName (Property, Owner)+ >
<!ELEMENT Property (#PCDATA) >
<!ELEMENT Owner (#PCDATA) >

Data View (Spreadsheet View)

…	…	…	…	…	…	…	…
$Property_n$	$Owner_n$	**-2**	…	…	…	…	…
$Property_n$	$Owner_n$	**-1**	A	B	C	D	…
$Property_n$	$Owner_n$	**0**	ColNameA	ColNameB	ColNameC	ColNameD	…
Property	**Owner**	**Sequence**	**ColNameA**	**ColNameB**	**ColNameC**	**ColNameD**	**…**
			A	**B**	**C**	**D**	**[…]**
		1	ColNameA	ColNameB	ColNameC	ColNameD	…
		2	dataA1	dataB1	dataC1	dataD1	…
		3	dataA2	[null]	[null]	dataD2	…
		…	…	…	…	…	…

Markup, HTML Family, etc.

The addition of one piece of global meta data is sufficient to enable a transform to a list of HTML format documents.

CSVplus	«=»	D1.html	D2.html	D3.html	D4.html	...
<dc:subject />	«=»	<dc:title />	<dc:title />	<dc:title />	<dc:title />	<dc:title />

CSVplus	ColNameA.html
... <dc:subject xml:lang="en-US"> ColNameA; ColNameB; ColNameC; ColNameD </dc:subject> ...	<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US"> <head> <title>ColNameA</title> </head> <body> <table> <tr> <th>ColNameA</th> </tr> <tr> <td>dataA1</td> </tr> <tr> <td>dataA2</td> </tr> </table> </body> </html>
	ColNameB.html
	...

1. Note that the xml:lang (above as RFC4646) is "optional" for the CSV format, but a requirement for an HTML representation. Less obvious, the language of the document (HTML) lang is a synonym for xml:lang. The real model language value of the document is in the *hreflang attribute* of the schema link. Every HTML document must do this, in case you ever wanted to know what *XSD Schema of Schemas* meant:

Name 1 or 2	AltName 1 or 2		Model (Union)		Name 2 or 1	AltName 2 or 1
lang	(xml:lang)		link@hreflang		lang	(xml:lang)
(lang)	xml:lang		link@hreflang		(lang)	xml:lang
xml:lang	(lang)		link@hreflang		xml:lang	(lang)
(xml:lang)	lang		link@hreflang		(xml:lang)	lang

2. Note that the *Property* and *Owner* information is likewise "optional" for the HTML but required for the CSVplus format. This deficiency can be remedied with the (optional) HTML profile attribute (see below). At present, web browsers are not set up to display multiple HTML files, so this is an academic exercise. Nonetheless, this is a necessary "afterthought" which should not come as any shock. In fact Dublin Core has solved the problem, in a somewhat awkward fashion, but without modification to the HTML legacy specifications.

CSVplus

```
            ...
<dc:subject
    xml:lang="en-US">
ColNameA;
ColNameB;
ColNameC;
ColNameD
</dc:subject>

<csv:Property>
  <ColNameA>A</ColNameA>
  <ColNameB>B</ColNameB>
  <ColNameC>C</ColNameC>
  <ColNameD>D</ColNameD>
</csv:Property>

<csv:Owner>
  <ColNameA>E</ColNameA>
  <ColNameB>F</ColNameB>
  <ColNameC>G</ColNameC>
  <ColNameD>H</ColNameD>
</csv:Owner>

            ...
```

ColNameA.html

```
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en-US">
  <head
      profile="http://www.rustprivacy.org/meta/property-names#">
     <title>ColNameA</title>
     <meta name="CSV.Property" content="A" />
     <meta name="CSV.Owner" content="E" />
     <link
        rel="schema.CSV"
        href="http://www.rustprivacy.org/meta/property-names#"
        hreflang="en-US"/>
  </head>
  <body>
    <table>
     <tr> <th>ColNameA</th> </tr>
     <tr> <td>dataA1</td> </tr>
     <tr> <td>dataA2</td> </tr>
    </table>
  </body>
</html>
```

ColNameB.html

...

In a data base, a CSVplus file is one table. It can be rendered in a spreadsheet, XML (as above) as well. The conversion to RDF is general, but depends upon a "hidden" identifierfor the nodes (rdf:nodeID).

In Conclusion

A (tentative) list of elements for the **CSVplus** name space is below.

Property (value)	Owner (value)	Description
csv:nodeID	target	rdf:nodeID (target list)
csv:order	target	A cardinal sort order (target)
csv:dsAux	target	Alternate Name List (target)
csv:dsPri	target	Primary Name List (target)
csv:dsAll	target	List Union (target)
csv:nodeID	model	rdf:nodeID (core list)
csv:typeof	model	List of Compact URI's (CURIE) rdfa typeof
csv:tokenNULL	model	token
csv:tokenRNULL	model	token
csv:tokenR	model	token
csv:modelDefault	model	Value Defaults (model)
csv:modelAux	model	Alternate Name List (model)
csv:modelPri	model	Primary Name List (model)
csv:modelBase26	model	Spreadsheet numbering, [A_ ... ZZZ]
csv:modelNames	model	List Union (model)

Models and Targets share four elements, Models have six elements unique to models. Targets have one element unique to Targets. The Spreadsheet Numbering is a shifted base 26 number line, considerably easier to list than to calculate with SQL. The author is properly ashamed. Redaction and Deletion behavior is inherited by the target from the model. The behavior is not strictly related to the use, filtering an rdf:bag of CURIEs from the Personally Identifiable Information (PII) name space <http://purl.org/pii/terms/> URI's.

There is no particular reason a list of CURIEs from another descriptive meta data name spaces such as the Dublin Core Terms, Dublin Core Elements, or FOAF could not be used as a "Core".

A Very Important Aside: There is a logical equivalence between the the csv:typeof element and "global meta data" (e.g. the xml:lang attribute). The typeof CURIE would be a list of the word 'xml:lang' once for every column or

<csv:typeof csv:typeof-typeof="document-global">xml:lang</csv:typeof>
<csv:typeof csv:typeof-typeof="document-local">URI, URI, URI ... </csv:typeof>

This fiercely ugly syntax, as much as anyone hates to admit it, is what links the semantic web together. In almost all markup , HTML, XML, RDF etc., the locale is factored out, with a global attribute. Not all meta data types will be so lucky.

Namespace Notes

Model

nsURL	nsPrefix	Note
http://www.rustprivacy.org/meta/property-names/	csv:	CSV Properties
http://www.rustprivacy.org/meta/property-names/core.rdf#	csv:	Auxilliary Table Name
http://www.rustprivacy.org/meta/property-names/core.rdf#	csv:	Primary Table Name
http://purl.org/pii/terms/	pii:	Personally Identifiable Information
http://www.w3.org/1999/02/22-rdf-syntax-ns#	rdf:	xmlns:rdf
http://www.w3.org/2004/02/skos/core#	skos:	xmlns:skos
http://purl.org/dc/elements/1.1/	dc:	xmlns:dc

Target(s)

nsURI (Names)	nsPrefix	Note @rdf:nodeID
http://www.w3.org/XML/1998/namespace#	xml:	WC3 (w3c.rdf)
http://www.w3.org/1999/xlink#	xlink:	W3C (w3c.rdf)
http://purl.org/dc/elements/1.1/	dc:	Dublin Core (dc.rdf)
http://purl.org/dc/terms/	dct:	Dublin Core (dc.rdf)
http://purl.org/dc/dcam/	dcam:	Dublin Core (dc.rdf)
http://docs.oasis-open.org/opendocument#	ooud:	OpenOffice (User Data) (oo.rdf)
http://docs.oasis-open.org/opendocument/meta/rdfa#	meta:	OpenOffice ODF (office:meta) (odf.rdf)
http://docs.oasis-open.org/opendocument/biblio#	oobi:	OpenOffice BIBLIO Database (oo.rdf)
http://mxr.mozilla.org/comm-central/source/mailnews/addrbook/public/nsIAbCard/idl#	nabc:	Netscape Address Book Card (abook.rdf)
https://developer.mozilla.org/en/nsIAbCard#	mabc:	Modzilla Address Book Card (abook.rdf)
http://www.modzilla.org/apps-thunderbird/address-book#	tbdc:	Thunderbird Address Book (abook.rdf)
http://www.microsoft.com/winmail#	wnmc:	WinMail Address Book (abook.rdf)

Example A

core.rdf – An RDF Data Base for the RUST Core ID's available now

core-type.rdf – An RDF File listing Core Elements by PII Type available now

source.rdf – An RDF Data Base for all the targets above available now

source-type.rdf – An RDF File listing Target Elements by PII Type available now

CSVplus-sample.zip

Three Sheet Spreadsheets (CSVplus, Model + Target, Query Output) available now

ODB Data Base (preloaded with Model Below) *Jan. 2010*

ODS Spreadsheet *Jan. 2010*

Model

(1 Name, 4 Physical Addresses, 4 Communications (Phone, Email, URL))

PERL Script *Jan. 2010*

Table

Queries

Spreadsheet View (Model)

Redacted View (Model)

Deleted View (Model)

Round Trip I View (Target Column Names, Data Deleted)

Round Trip II View (Target Column Names, Data Redacted)

PERL Script (CSVplus to XML headers only) *Jan. 2010*

XSLT (XML headers only to RDF headers only) *Jan. 2010*

XSLT Bulk nodeID Lookup *Jan. 2010*