XSLT Ingest Example: Count

Count

Start Previous Next

The next phase is to count the unresolved person (URPs) matches and the unresolved organizations (UROs). To do this we apply the transforms **countURPs.xsl** and **countUROs.xsl**. Figure 9 shows the more complex URP case. The URO case is much simpler since organizations are compared based on a single string.

- [F9H0] Notice that this transform outputs text not XML.
- [F9H1] This code builds up a variable **gcCount** that contains a sequence of integers each of which represents the number of distinct names /netids in each group formed by name parts. This will also include names without **NETID**s in the source. The grouping order used here does not handle the case of name variants with the same netid in the same result set. Such cases can arise because of change of marital status which is not easily distinguished, in an automated way, from other cases such as a mistyped netid. Any such problems can be fixed by post processing to correct any misattributions based on common netid. Changing the grouping order (i.e. by netid then name parts) also works but with additional coding complexity. Note that we are only interested in **EduRecords** where the **personURI** is empty those are the URPs in **ED0.xml**.
  - [F9H1a] Collect all nodes in the current group formed by grouping by name parts and then group them by **nid** (the tag we use in **ED0.xml** to represent a source **NETID**). The variable **gp** refers to this sequence of nodes.
  - [F9H1b] Count the nodes in each sub-group and append the count to the **cgCounts** sequence.
- [F9H2] Output the grand total by summing the sequence **cgCounts**. This is the number of distinct URPs found in **ED0.xml**.

```
<?xml version="1.0"?>
<xsl:stylesheet version='2.0'
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:vfx='http://vivoweb.org/ext/functions'
        xmlns:xs='http://www.w3.org/2001/XMLSchema'
        exclude-result-prefixes='xs vfx xsl'
>

<xsl:output method='text'/>                                          0

<xsl:template match="/EduRecords">

<xsl:variable name='cgCounts' as='xs:integer*'>                      1
<xsl:for-each-group select='EduRecord[personUri=""]'
        group-by='lower-case(concat(normalize-space(ln),"|",
                                    normalize-space(fn),"|",
                                    normalize-space(mn)))'>

<xsl:variable name='gp'>                                             1a
<xsl:for-each-group select='current-group()'
                            group-by='vfx:adjust(nid)'>
<xsl:copy-of select='.'/>
</xsl:for-each-group>
</xsl:variable>

<xsl:value-of select='count($gp/EduRecord)'/>                        1b
</xsl:for-each-group>
</xsl:variable>

<xsl:value-of select='concat(sum($cgCounts), "&#x0A;")'/>            2

</xsl:template>

<xsl:include href='auxfuncs.xsl'/>
</xsl:stylesheet>
```

countURPs.xsl - Figure 9

In our example, we find 11 unresolved people URIs and 5 unresolved organization URIs. Next we will construct enough URIs to fill in the URPs and UROs.

Start Previous Next