SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Generic XSLT Mapping for SAP XI, Part I

Table of Contents

Applies To:

This piece of code was written primarily for XSLT mapping in XI 3.0, but it can be used in any product where XSLT mapping can be used.

Summary

This XSLT code sample is very useful in places where we might need to traverse through the nodes of an XML file, identify certain nodes having common fields, and then aggregate those field values and display them as a single node instead of multiple nodes. There are many requirements in XI where, while mapping, we might be required to do some grouping and aggregation of data in XML nodes; this sample code explains how grouping can be done on an XML document using XSLT.

Any enhancements to this piece of code are welcome.

By: Sameer Rahim Shadab

Company: Intelligroup Asia Private Ltd. (www.intelligroup.com)

Date: 07 July 2005

Files Involved

input.xml: This is the source XML file for the XSLT code.

grouping_xslt.xsl: This is the XSL document that does the grouping.

output.xml: This is how the output file would look after executing the XSLT code.

Requirement Scenario

In the “input.xml” file there are four ‘item’ nodes under the 'salesOrder' node. Out of the four 'item' nodes, three have the same 'itemDescription' value of 'Blazer' and the fourth has value of 'Trouser'.

The requirement is to identify nodes with same 'itemDescription' and then aggregate the subsequent 'itemRetailPrice' and 'itemSalePrice' into a single node, so that the output looks like the XML file below.

Solution

input.xml

```
<?xml version="1.0"?>
<salesOrder>
<item>
<itemDescription>Blazer</itemDescription>
<itemRetailPrice>29.95</itemRetailPrice>
<itemSalePrice>24.95</itemSalePrice>
</item>
<item>
<itemDescription>Blazer</itemDescription>
<itemRetailPrice>29.95</itemRetailPrice>
<itemSalePrice>24.95</itemSalePrice>
</item>
<item>
<itemDescription>Blazer</itemDescription>
<itemRetailPrice>29.95</itemRetailPrice>
<itemSalePrice>24.95</itemSalePrice>
</item>
<item>
```

```
<itemDescription>Trouser</itemDescription>
<itemRetailPrice>10.55</itemRetailPrice>
<itemSalePrice>8.95</itemSalePrice>
</item>
</salesOrder>
```

grouping_xslt.xsl

```
<?xml version='1.0'?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
<!-- Build a key to group item by itemDescription -->
<xsl:key name="itemDescription" match="salesOrder/item"
use="itemDescription"/>
<xsl:template match="/">
<salesOrder>
<!-- iterate on each group -->
<xsl:for-each select="salesOrder/item[generate-id(.) = generate-
id(key('itemDescription', itemDescription)[1]) ]">
<xsl:variable name="group" select="key('itemDescription', itemDescription)"/>
<xsl:variable name="itemDescription">
<xsl:value-of select="itemDescription"/>
</xsl:variable>
<xsl:variable name="itemRetailPrice">
<xsl:value-of select="sum($group/itemRetailPrice)"/>
</xsl:variable>
<xsl:variable name="itemSalePrice">
<xsl:value-of select="sum($group/itemSalePrice)"/>
</xsl:variable>
<!-- Generate the final XML file -->
<item>
<itemDescription>
<xsl:value-of select="$itemDescription"/>
</itemDescription>
<itemRetailPrice>
```

```
<xsl:value-of select="$itemRetailPrice"/>
</itemRetailPrice>
<itemSalePrice>
<xsl:value-of select="$itemSalePrice"/>
</itemSalePrice>
</item>
</xsl:for-each>
</salesOrder>
</xsl:template>
</xsl:stylesheet>
```

output.xml

```
<?xml version='1.0' encoding='UTF-8' ?>
<salesOrder>
  <item>
    <itemDescription>Blazer</itemDescription>
    <itemRetailPrice>89.85</itemRetailPrice>
    <itemSalePrice>74.85</itemSalePrice>
  </item>
  <item>
    <itemDescription>Trouser</itemDescription>
    <itemRetailPrice>10.55</itemRetailPrice>
    <itemSalePrice>8.95</itemSalePrice>
  </item>
</salesOrder>
```

Author Bio

Sameer Rahim Shadab has about seven years of experience in the IT industry. He has been with Intelligroup Asia Pvt Ltd. since January 2003. He is a part of Intelligroup's SAP NetWeaver competency group and has been working on SAP XI for around a year now.