
InRoads Group of Products

Creating XML Lookup Table Style Sheets

1. Introduction

This document describes steps to create style sheets that look in another XML file as well as the active XML file to obtain data for a combined output. The example in this document produces an ASCII file, but the principles apply to HTML output as well.

This procedure applies to any application within the InRoads Group of products being developed by the Civil Engineering Development group.

This document cannot teach everything needed to create XML style sheets. There are a number of excellent books available on the subject and Bentley also offers a 3-day class on InRoads Reporting with XML. This document only describes the steps necessary to create style sheets that look in another XML file as well as the active XML file to obtain data for a combined output.

2. Workflow

XSL provides a simple way to include one XML document into another, allowing values from both documents to be combined into the final output. The following sections explain how to combine and use multiple XML documents.

2.1. Creating the Additional XML Document

The additional XML document can be another document produced by an InRoads Group command or it can come from another application entirely. It can even be a document created manually by the user. For this document, we are concentrating on creating a "lookup table" containing data to be matched by some value in the active XML. An example of a user-created lookup table document can be found in the `\XML Data\Custom` directory called *SurveyFeature.xml*. An excerpt is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<Document>
  <SurveyFeature numericCode="" alphaCode="BB" pointType="Breakline"
style="BB" description="Bottom of bank"/>
  <SurveyFeature numericCode="" alphaCode="BBERM" pointType="Breakline"
style="BBERM" description="Bottom of berm"/>
  <SurveyFeature numericCode="" alphaCode="BM" pointType="Breakline - Do Not
Triangulate" style="BM" description="Benchmark"/>
  .
  .
  .
  <SurveyFeature numericCode="4" alphaCode="SPOT" pointType="Breakline - Do
Not Triangulate" style="SPOT" description="Spot elevation/surface shot"/>
  <SurveyFeature numericCode="10" alphaCode="CPXYZ" pointType="Breakline - Do
Not Triangulate" style="CPXYZ" description="Primary Control Point XYZ"/>
  <SurveyFeature numericCode="11" alphaCode="CPXY" pointType="Breakline - Do
Not Triangulate" style="CPXY" description="Horizontal Control Point XY"/>
</Document>
```

The declaration on the first line is a required entry. The simplest root element for any user-created XML document is `Document`, but any element name can be used. The child element names and their attribute names can also be anything desired by the user. It is usually best to use a descriptive name with a minimum of abbreviations so that anyone can look at the file and tell what the data represents.

2.2. Creating a Variable to Include the XML Document

A variable must be created to hold the name of the look up table XML document which is then included using the `document()` function available in XSL. This must be done near the top of the style sheet as a child of the `xsl:stylesheet` element so it will be a global variable available to any part of the style sheet. Here is an example from the style sheet called *PointStyleToSurveyNumeric.xsl* found in your `\XML Data\Custom` directory which shows the entry to include the file listed in Section 2.1.

```
<xsl:variable name="featureTableList" select="document('SurveyFeature.xml')"/>
```

Note the single quotes surrounding the file name.

2.3. Referencing the Additional XML Document

Once the file is included, it can be referenced by its variable name, and any data contained within it can be treated in the same way as any data from the active XML document. An example from the *PointStyleToSurveyNumeric.xsl* is shown below:

```
<xsl:variable name="numericCode"
select="$featureTableList/*/SurveyFeature[@style =
current()/@style]/@numericCode"/>
```

In this example, a variable called `numericCode` is being populated with data from any `numericCode` attribute from the included lookup table file (referenced by `$featureTableList`) where the `style` attribute of the `SurveyFeature` element in the lookup table matches the `style` attribute of the current point in the active XML document.

2.4. Sample Output

A portion of the output from the style sheet and XML files detailed above looks like this:

```
* FILE NAME: G:\datasets\kranjiV88\kranji.alg
* DATE: 3/16/2006
* PROJECT NAME: kranji
* DESCRIPTION:
* UNITS: METRIC*
*PT NO. EASTING NORTHING ELEVATION NUMERIC CODE
60 -2650.28626 -3371.17792 0.00000 100
78 -2704.37375 -3326.74208 0.00000 13
80 -1904.32062 -2352.91397 0.00000 13
82 -1841.64327 -2275.74768 0.00000 13
1 1214.13369 -186.14166 620.00000 1000
2 1345.34565 -1170.23131 620.00000 955
3 492.46795 2110.06754 610.00000 150
4 743.95752 2011.65857 610.00000 408
5 820.49783 2350.62279 610.00000
6 1236.00235 2416.22876 609.50000 160
7 -904.56001 1569.95629 567.60000
```

3. References

Harold, Elliotte Rusty. *XML Bible*. IDG Books Worldwide, Inc, Foster City, CA, 1999.

Higginbotham, Debbi. *Creating ASCII Output Style Sheets.doc*.

Kay, Michael. *XSLT Programmer's Reference 2nd Edition*. Wrox Press, Ltd, Birmingham, UK, 2001.

4. Glossary

Terms used in this document are:

ASCII	American Standard Code for Information Interchange – the character set upon which most text files used by modern computers is based
Attribute	A component of an XML or HTML element that provides additional information about a specific instance of the element in the form of a <code>name="value"</code> pair
Child element	An element that is nested (contained) within another element
Comment	An item in an XML or HTML document that is used to carry extraneous information that is not part of the data; written between the delimiters <code><!--</code> and <code>--></code> .

Element	A markup tag, consisting of a start tag and an end tag, and the content, text, or data, contained within the tag
Empty element	An element with no content, although it may have attributes
HTML	<u>H</u> yper <u>t</u> ext <u>M</u> arkup <u>L</u> anguage – an SGML application created for Web documents
Markup	Tags added to a document to define the pieces and parts of the document and to describe the role they play; markup works on any computer
Meta language	A language used for defining other languages
Parent element	An element that has one or more elements nested within it
Root element	The upper level parent element of which all other elements in the document are children
SGML	<u>S</u> tandard <u>G</u> eneralized <u>M</u> arkup <u>L</u> anguage – a meta language created for general document structuring
Tag	HTML and XML code that delineates elements; tags can have three kinds of meaning – structure, semantics and style
XML	<u>E</u> xtensible <u>M</u> arkup <u>L</u> anguage – a text format for storing structured data; a meta language based on simplified SGML created for Web use
XML style sheet	Well-formed XML document that uses XSLT to transform XML data for presentation
XSL / XSLT	<u>E</u> xtensible <u>S</u> tyl <u>S</u> heet <u>L</u> anguage: <u>T</u> ransformations – an advanced style sheet mechanism that provides browsers with formatting and display information