Paper 171
Taking XML’s Measure: Using SAS® to Read In and Create XML for Analytic Use and Websites

Louise S. Hadden
Abt Associates Inc.
Louise Hadden has been using and loving SAS since the days of punch cards and computers the size of a tiny house. She spends most of her time in support of health policy analytics at Abt Associates Inc. and loves a good SAS reporting challenge. She is an ardent life long learner and reads voraciously, loves photography and volunteers at the MSPCA Boston Adoption Center walking, training and photographing dogs.
Introduction

XML has become a standard over the years for populating websites and transferring information. This presentation demonstrates how to parse mystery XML files, read in XML files that you can’t right-click on, read into Microsoft Excel using SAS®, how to use maps and schemas to input and output various XML representations, and how to construct and output “measure code” data sets from input data to maximize the flexibility of XML data representation and usage.
Introduction

XML

- HTML  Markup Language used to build static web pages
- HTML5  Latest version of HTML with support for multimedia
- XML  Extensible markup language
- XHTML  XML that mirrors HTML in syntax and adds hypertext capability
- DHTML  Dynamic HTML
- KML  ML to display geographic data
Introduction

Markup Languages

Wikipedia tells us “Markup Language is a system for annotating a document in a way that is syntactically distinguishable from the text.”
Introduction

More on XML files

SAS has a very informative document at http://support.sas.com/rnd/base/ods/templateFAQ/Template_xml.html#overview

SAS Tip sheets are also available for both 9.3 and 9.4.

---

**LIBNAME Statement Syntax**

```
LIBNAME libref XML2 <SAS-library|XML-document-path> <options>;
```

**Required Arguments**

- `libref` a valid SAS name to associate with the XML document. A libref cannot exceed eight characters.

- `XML2` the LIBNAME engine name. (Alias: XML92)

- `<SAS-library|XML-document-path>` the physical location of the XML document.

**Options**

- `FORMATACTIVE=NO|YES`

---

**XMLV2 Engine**

XMLV2 accesses enhancements since SAS 9.1.3, which includes the ability to assign a libref to a SAS library in a directory-based environment, and enhanced XMLMap support. XML92 is supported as an alias.

---

**XMLMap Syntax Version 2.1**

```
XMLMap version=number name=XMLMap
description=description

NAMESPACE count=number
NS id=number <prefix=prefix>

OUTPUT
HEADER
ATTRIBUTE name=#name value=value
TABLEREF name=name

TABLE name=set name=table
TABLE-PATH syntax=type
TABLE-END PATH syntax=type
beginend='BEGIN|END'
TABLE-DESCRIPTION
```
Introduction
Basic XML Concepts

• XML documents
  – Nodes
  – Relationships
• DTD
• XML maps or schemas
• XSL style sheets
Methods for Reading XML in

IRL Example

Search 30815 SAS proceedings from SAS Global Forum, SUGI, PharmaSUG, NESUG, SESUG, WUSS, MWSUG, PNWSUG and SCSUG.

This search is based on title and author. You need to specify at least 3 characters in one field.

Searches are case insensitive and can be done in one of 3 ways:

- A literal string within double quotes. Examples: "dynamic data exchange" or "dde"
- Multiple words separated by a plus (+). The results must contain all words. Example: clinical metadata
- Multiple words separated by blanks. The results must contain at least one of the words.

**TITLE contains**

AND **AUTHOR contains**
Methods for Reading XML in

IRL Example

   Louise Hadden; Roberta Glass
   SESUG 2017
   Keywords: macro,

   Rick Andrews; Louise Hadden; Robert Allison
Methods for Reading XML in

IRL Example

```xml
<?xml version="1.0" encoding="utf-8"?>
<searchresults CreationDate="2018-04-02T15:03:20Z" SearchTitle="" SearchAuthors="Louise Hadden"
    <paper paperid="sesug2017.psa-73"
        <title>Document and Enhance Your SAS® Code, Data Sets, and Catalogs with SAS Functions, Macro</title>
        <authors>Louise Hadden; Roberta Glass</authors>
        <keywords>macro</keywords>
        <conference>SESUG</conference>
        <month>2017-11</month>
    </paper>

<paper paperid="sesug2017.riv-42"
    <title>Methods for Creating Sparklines using SAS®</title>
    <authors>Rick Andrews; Louise Hadden; Robert Allison</authors>
    <link>http://www.lexjansen.com/sesug/2017/RIV-42.pdf</link>
    <conference>SESUG</conference>
    <month>2017-11</month>
</paper>
```
Methods for Reading XML In
Open Microsoft Excel and Open XML
Methods for Reading XML In Open Microsoft Excel and Open XML
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>paper_ORDI</td>
<td>paper_paperid</td>
<td>title</td>
<td>Comment</td>
<td>TopicArea</td>
<td>Keywords</td>
<td>authors</td>
<td>link</td>
<td>conference</td>
</tr>
<tr>
<td>2</td>
<td>1 sesug2017.psa-73</td>
<td>Document and Enhance Your SAS® Code, Data Sets, and Catalog macro</td>
<td>Louise Hach</td>
<td><a href="http://www">http://www</a> SESUG</td>
<td>2017-11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3 sesug2017.epo-91</td>
<td>Purrfectly Fabulous Feline Functions</td>
<td>Louise Hach</td>
<td><a href="http://www">http://www</a> SESUG</td>
<td>2017-11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5 mwsug2017.tt08</td>
<td>Get Smart! Eliminate Kaos and Stay in Control - Creating a Complex Direct</td>
<td>Louise Hach</td>
<td><a href="http://www">http://www</a> MWSUG</td>
<td>2017-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7 mwsug2017.po01</td>
<td>Red Rover, Red Rover, Send Data Right Over: Exploring External Geographic</td>
<td>Louise Hach</td>
<td><a href="http://www">http://www</a> MWSUG</td>
<td>2017-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8 scsug2017.ih05</td>
<td>Red Rover, Red Rover, Send Data Right Over: Exploring External Geographic</td>
<td>Louise S.</td>
<td><a href="http://www">http://www</a> SCSUG</td>
<td>2017-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>10 wuss2017.043.dm</td>
<td>Get Smart! Eliminate Kaos and Stay in Control Creating a Complex Director</td>
<td>Louise Hach</td>
<td><a href="http://www">http://www</a> WUSS</td>
<td>2017-09</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>12 wuss2017.051.dp</td>
<td>Red Rover, Red Rover, Send Data Right Over: Exploring External Geographic</td>
<td>Louise Hach</td>
<td><a href="http://www">http://www</a> WUSS</td>
<td>2017-09</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Methods for Reading XML In

Read In Using Automap

filename ajw ' .\AlanWhite.xml';
filename map ' .\Map\AJW.map';
libname ajw xmlv2
automap=replace xmlmap=map;

proc contents data=ajw._all_; run;
<xml version="1.0" encoding="UTF-8">
<!-- ###################################################### -->
<!-- 2018-04-07T12:21:39 -->
<!-- SAS XML Libname Engine Map -->
<!-- Generated by XML Mapper, 904000.0.0.20130522190000_v940 -->
<!-- ###################################################### -->
<!-- ### Validation report ### -->
<!-- ###################################################### -->
<!-- XMLMap validation completed successfully. -->
<!-- ###################################################### -->
<xmlmap name="AUTO_GEN" version="2.1">
  <namespaces count="0"/>
  <!-- ###################################################### -->
  <table description="searchresults" name="searchresults">
    <table-path syntax="XPath">/searchresults</table-path>
    <column class="ORDINAL" name="searchresults_ORDINAL">
      <increment-path beginend="BEGIN" syntax="XPath">/searchresults</increment-path>
      <type>numeric</type>
      <datatype>integer</datatype>
    </column>
  </table>
</xmlmap>
Methods for Reading XML In
SAS XML Mapper
Methods for Reading XML In
SAS XML Mapper
Methods for Reading XML In

Read In Using an Existing Map

filename ajw '.\AlanWhite.xml';
filename map '.\Map\AJW.map';
libname ajw xmlv2 xmlmap=map;
Capitalizing on the “Extensible”

Case Study: Nursing Home Compare

Find a nursing home

Nursing Home Compare has detailed information about every Medicare and Medicaid-certified nursing home in the country. A nursing home is a place for people who can’t be cared for at home and need 24-hour nursing care.

Search below to find nursing homes based on a location and compare the quality of care they provide and their staffing.

A field with an asterisk (*) is required.
* Location
  Example: 45602 or Lima, OH or Ohio
  ZIP code or City, State or State

Nursing home name (optional)
  Full or partial nursing home name

Search

Spotlight
- Get updated state website information, including the ability to electronically file a complaint against a nursing home in some states
- Use these helpful resources when looking for a nursing home:
  - Guide to choosing a nursing home
  - Nursing Home Checklist

Tools and Tips
- First time here?
  - Visit About Nursing Home Compare and the Resources section to learn more about the site and available resources.
- Learn about Medicare coverage of skilled nursing facility care, and swing bed services.
- Get help filing a nursing home

Additional Information
- Nursing Home Compare data last updated: August 23, 2017 (Data are updated on or about the fourth Wednesday of the month).
- Download the database
- Learn how we calculate ratings
- For nursing homes: Update your address, phone number and other administrative data.
Capitalizing on the “Extensible” Concepts

• Many files from many different sources go into the Nursing Home XML

• Original files were at different levels, for example nursing home residents versus providers

• Thousands of elements or nodes
Capitalizing on the “Extensible” XML Output

ODS MARKUP BODY=TEST.XML;

PROC PRINT DATA=TEST;
RUN:

ODS MARKUP CLOSE;
data msr_ownership (keep=provnum msr_cd value occurrence ftnt filedate);
   length provnum $ 6 msr_cd $ 20 value $ 120 ftnt $ 12 occurrence $ 3
   filedate $ 8;
set dd.owner_ocr;

.....
msr_cd = 'ASSOCDATE';
value = assoc_date_text;
if value ne ' ' then output;
Run;
Methods for Reading XML in Creating Measures

proc sql;
create table out.msr_Owners_&fileyear.&filedate. as
    select PROVNUM as PID,
           MSR_CD as MCD,
           occurrence as OCR,
           VALUE as SV,
           ftnt as FN,
           "Text" as ST
    from msr_ownership
order by PID, MCD, OCR;
quit;
Methods for Reading XML in
Creating Measures

PID 015009
MCD ASSOCDATE
OCR 1
SV since 09/01/1969
FN
ST Text
Capitalizing on the “Extensible” XML Output

As with reading XML output in, the SAS XML libname engine and a schema or map are employed.

```plaintext
filename mapt ".\map\MapTemplate.map";
filename map ".\map\MapModified.map";
libname temp1 xml92
xmltype=xmlmap xmlmap= map;
```
Capitalizing on the “Extensible” XML Output

As with reading XML output in, the SAS XML libname engine and a schema or map are employed.

```
filename map ".\map\MapModified.map";
libname temp1 xml92 xmltype=xmlmap xmlmap= map;
filename out ".\XML\&outnm.XMLOut.xml";
```
Capitalizing on the “Extensible”

XML Output

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<!--
SAS XML Libname Engine (SAS92XML)
SAS XMLMap Generated Output
Version 9.04.01M1P12042013
Created 2018-04-06T19:55:49
-->
<PMIDetails xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <HeaderInformation>
    <DataSource>Abt - Nursing Home Compare - Combined File 4 of 10</DataSource>
    <DateCreated>04/01/2018</DateCreated>
    <PMICount>1225</PMICount>
  </HeaderInformation>
</PMIDetails>
Capitalizing on the “Extensible” XML Output

```xml
<PMI PID="175008">
  <M MCD="401">
    <S ST="Percentage">
      <SV >14.7</SV>
    </S>
  </M>
</PMI>
```
Conclusion

SAS has provided many tools to both read XML into SAS and to output XML from SAS. The use of “measure code” transformation greatly extends the power and flexibility of XML generation from SAS.

“Wit beyond measure is man's greatest treasure.”

J.K. Rowling
Acknowledgements

The author wishes to thank Chevell Parker of SAS, a former colleague Fred Pratter, a current colleague Nancy McGarry, Troy Martin Hughes and Lex Jansen for mentoring and inspiring me.

Thanks also to the Nursing Home Compare project team for providing a welcome challenge to encourage finding new ways to improve our data processing and transfers, especially Christianna Williams of Abt Associates and Zach Sarver of CGI.
Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Louise S. Hadden
Abt Associates Inc.
617-349-2385
Louise_hadden@abtassoc.com
abtassociates.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.