

Reading Multiple XML Files into One SAS® Data Set

Mayumi Takada

Arkansas Foundation for Medical Care, Fort Smith, AR

ABSTRACT

Have you ever needed to create a SAS data set from multiple XML files? You can develop an XML Map using SAS XML Mapper application. It will create a SAS sample program to read in XML file, but it will not take care of reading multiple XML files. This paper presents how to use the SAS XML Mapper application and then demonstrates a SAS macro program to create a SAS data set from multiple XML files.

INTRODUCTION

What is the SAS XML Mapper application? SAS User's Guide states "SAS XML Mapper is a Java-based application that assists you in creating and modifying XMLMaps for use by the XML engine. SAS XML Mapper provides a graphical interface that you can use to generate the appropriate XML elements. SAS XML Mapper analyzes the structure of an XML document or an XML schema and generates basic XML syntax for the XMLMap[...]"

After creating the XMLMap using SAS XML Mapper application, the XMLMap syntax tells the XML engine how to interpret the XML markup into SAS data set, variables and observations by using XMLMAP= option in the program.

The SAS macro program(Figure. 6) which is shown in this paper reads in consecutive order from multiple XML files in the directory and each file is added to a SAS data set using PROC APPEND which performs an update in place on the BASE= data set.

SAS XML MAPPER: How to create XML Map using SAS XML Mapper Application

Below Figure1. is the sample XML file:

Figure 1: Sample XML File

```
<?xml version="1.0" encoding="UTF-8" ?>
<submission action-code="ADD" data="CLINICAL" type="HOSPITAL" version="1.0">
+ <file-audit-data>
- <abstraction-audit-data>
  <abstraction-date>10-10-2007</abstraction-date>
  <abstractor-id>abc</abstractor-id>
  <total-abstraction-time>324</total-abstraction-time>
</abstraction-audit-data>
- <provider>
  <provider-id>000000</provider-id>
  <npi>999999999</npi>
- <patient>
  <first-name>Jane</first-name>
  <last-name>Dow</last-name>
  <birthdate>06-29-1942</birthdate>
  <sex>F</sex>
  <race>1</race>
  <ethnic>N</ethnic>
  <postal-code>00000</postal-code>
- <episode-of-care measure-set="HF">
  <admit-date>08-21-2007</admit-date>
  <discharge-date>08-24-2007</discharge-date>
  <hospital-patient-id>992109</hospital-patient-id>
  <detail answer-code="428.0" question-cd="PRINDX" row-number="0" />
  <detail answer-code="295.90" question-cd="OTHRDX#" row-number="1" />
  <detail answer-code="250.00" question-cd="OTHRDX#" row-number="2" />
  .
  <detail answer-code="Y" question-cd="LVEFEVAL" row-number="0" />
  <detail answer-code="N" question-cd="LVSD" row-number="0" />
  <detail answer-code="N" question-cd="ADSMKHIST" row-number="0" />
  <detail answer-code="630" question-cd="PHYSICIAN_1" row-number="0" />
</episode-of-care>
</patient>
</provider>
</submission>
```

When you open this sample XML file from the tool bar *File*, select *Open XML..* and the screen Figure 2 will be displayed. The bottom part of the screen (green circle) displays the XML source code. To create XML Map you need to drag the elements from top left corner (red circle) to top right corner (blue circle) of the screen. Also it is recommended to change the default map name SMLEMAP to avoid ambiguity. If you don't change it, the statement appears on the *Validate* tab (yellow circle).

Figure 2. Application Screen 1

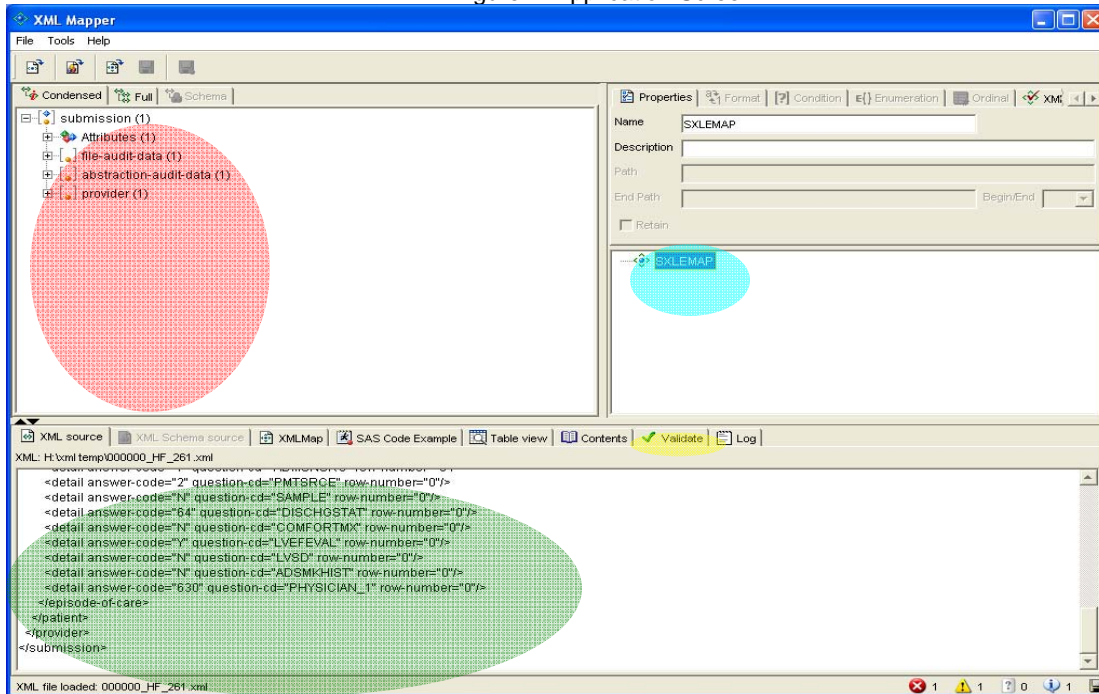


Figure 3: The sample dataset has fifteen observations (the element *detail* indicates fifteen – blue circle). Select the element *detail* and drag to the top right corner (pink circle) first, then select the elements you need. But remember, when you drop the elements to the top right corner (pink circle), drop over on *detail*. The order of the elements can be changed with a right-click.

Figure 3. Application Screen 2

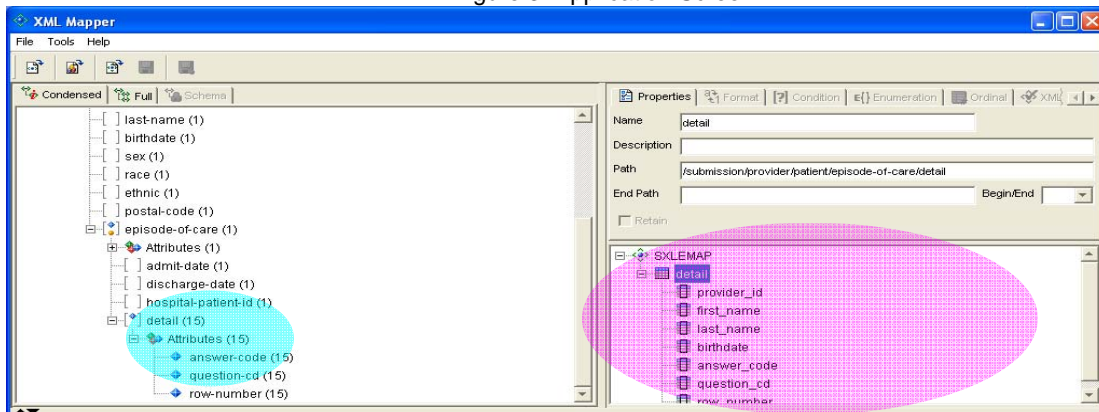
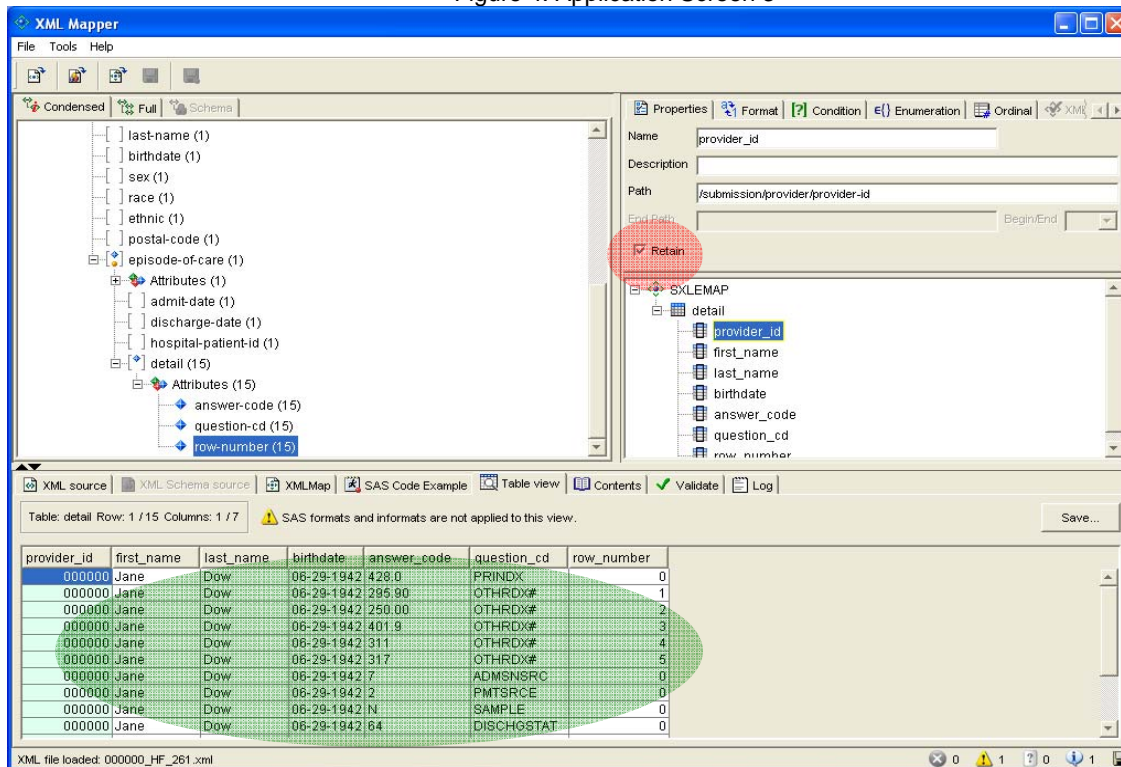


Figure 4 shows the property of the element *provider_id* which has fifteen observations so the box next to the retain has a check-mark (it is automatically checked. Red circle). The name *detail* can be changed because it will be the SAS data set name (the name can be changed anytime after you create the data set).

The bottom box (green circle) of Figure 4 shows *Table View* of SAS data set . You can view the XML Map by clicking the tab *XMLMap* and will be able to save by selecting the *Save XMLMap as...* under *File* menu. Also you can see the sample SAS program by clicking the tab *SAS Code Example* and will be able to save by selecting the *Save SAS* under *File* menu.

Figure 4. Application Screen 3



Below Figure 5 is the XML Map which is created from SAS XML Mapper and will be used to read XML into SAS data set.

Figure 5. XML Map Output

```

<?xml version="1.0" encoding="windows-1252"?>

<!-- ##### -->
<!-- 2008-06-12T12:27:32 -->
<!-- SAS XML Libname Engine Map -->
<!-- Generated by XML Mapper, 9.1.0300.20040709.2028 -->
<!-- ##### -->
<SXLEMAP xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="sample_map" version="1.2"
xsi:noNamespaceSchemaLocation="http://www.sas.com/xml/schema/sxle12.xsd" > ..... 1.
  <!-- ##### -->
  <TABLE name="detail"> ..... 2.
    <TABLE-PATH syntax="XPath"/>/submission/provider/patient/episode-of-care/detail</TABLE-PATH> ..... 3.
    <COLUMN name="provider_id" retain="YES"> ..... 4.
      <PATH syntax="XPath"/>/submission/provider/provider-id</PATH> ..... 5.
      <TYPE>numeric</TYPE> ..... 6.
      <DATATYPE>integer</DATATYPE> ..... 7.
    </COLUMN> ..... 4.
    <COLUMN name="first_name" retain="YES">
      <PATH syntax="XPath"/>/submission/provider/patient/first-name</PATH>
      <TYPE>character</TYPE>
      <DATATYPE>string</DATATYPE>
      <LENGTH>4</LENGTH>
    </COLUMN>
    <COLUMN name="last_name" retain="YES">
      <PATH syntax="XPath"/>/submission/provider/patient/last-name</PATH>
      <TYPE>character</TYPE>
      <DATATYPE>string</DATATYPE>
      <LENGTH>3</LENGTH>
    </COLUMN>
    <COLUMN name="birthdate" retain="YES">
      <PATH syntax="XPath"/>/submission/provider/patient/birthdate</PATH>
      <TYPE>numeric</TYPE>

```

```

<DATATYPE>date</DATATYPE>
<FORMAT width="10">IS8601DA</FORMAT>
<INFORMAT width="8">MMDDYY</INFORMAT>
</COLUMN>
<COLUMN name="answer_code"> .....8.
  <PATH syntax="XPath">/submission/provider/patient/episode-of-care/detail/@answer-code</PATH>
  <TYPE>character</TYPE>
  <DATATYPE>string</DATATYPE>
  <LENGTH>6</LENGTH>
</COLUMN>
<COLUMN name="question_cd"> .....9.
  <PATH syntax="XPath">/submission/provider/patient/episode-of-care/detail/@question-cd</PATH>
  <TYPE>character</TYPE>
  <DATATYPE>string</DATATYPE>
  <LENGTH>11</LENGTH>
</COLUMN>
<COLUMN name="row_number"> .....10.
  <PATH syntax="XPath">/submission/provider/patient/episode-of-care/detail/@row-number</PATH>
  <TYPE>numeric</TYPE>
  <DATATYPE>integer</DATATYPE>
</COLUMN>
</TABLE> .....2.
</SXLEMAP> .....1.

```

Recap from 1 through 10 above Figure 5.

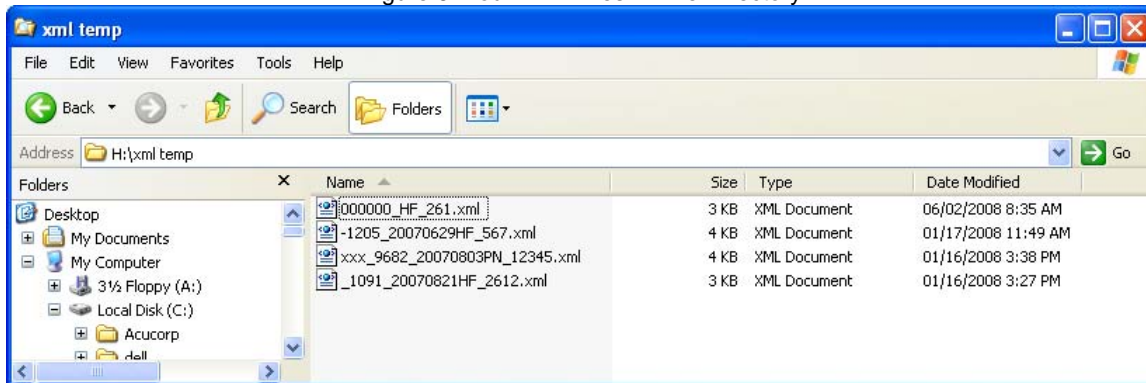
1. SXLEMAP is the root-enclosing element that contains the definition.
2. TABLE element that contains *detail* data set definition. *name=...* specifies the name for the SAS data Set.
3. Location path that tells the XML engine where in the XML document to locate and access specific elements in order to collect variables for SAS data set.
4. COLUMN elements contain the attributes for the *provider_id*. *retain="YES"* keeps the current value until the end of answer code (8), question cd (9) and row number (10).
5. Location path that tells the XML engine where in the XML document to locate and access a specific tag for the variable.
6. SAS data type for the variable.
7. The type of data being read from the XML for the variable.
8. 8-10. explained in 4.

See reference below on 'Reading and Writing XML files from SAS' from SUGI 29 Hands-on Workshop for more explanation on how to use the application, and it will be very helpful.

SAS MACRO PROGRAM: Read Multiple XML files and create one SAS data set

There are four XML sample files in h:\XML Temp directory which was used here as below Figure 6;

Figure 6. Four XML Files in The Directory



The Sample SAS program was created from the SAS Mapper Application, but it will read only one XML file. You need to modify the program to read in all of the XML files in the directory. Read first XML file and append to the base data set, then read next XML file and append to the base data set, then continuously go to the next XML file. Please do not forget at the beginning of the program to delete all of the data set in the directory which you are going to save the

SAS data set. This is necessary because you use PROC APPEND function in this program and you don't want to append the data set to the data set that already exists in the library. The program below will read the XML files.

Figure 6. SAS Macro Program which reads multiple XML files.

```

options symbolgen;
libname outfile 'H:\test';
proc datasets lib=outfile kill; ..... 1.
run; quit;

%macro test(dir);
%let filerf=mydir;
%let rc=%sysfunc(filename(filerf,&dir)); ..... 2.
%let did=%sysfunc(dopen(&filerf)); ..... 3.
%let memcnt=%sysfunc(dnum(&did)); ..... 4.
%let cnt=0;
%do i=1 %to &memcnt;
  %let fname=%scan(%qsysfunc(dread(&did,&i)),1,'); ..... 5.
  %let cnt=%eval(&cnt+1);
  %let name&cnt=&fname; ..... 6.
%end;
filename SXLEMAP 'S:\program\case selection\iqi_xml_sfy2008.map'; ..... 7.
%do i=1 %to &cnt; ..... 8.
  libname xml&i xml "&dir.\&&name&i...xml" xmlmap=sxlemap access=readonly; ..... 9.
  proc datasets lib=xml&i; ..... 10.
  run;
  proc append base=outfile.detail data= xml&i..detail force; ..... 11.
  run;
  libname xml&i clear; ..... 12.
%end;
/* close the directory */
%let rc=%sysfunc(dclose(&did)); ..... 13.
%mend test;
%test(h:\xml temp); ..... 14.

```

Recap from 1 through 14 in the program above Figure 6.

1. Delete all SAS files in the library *outfile*.
2. FILENAME function which assigns a filerf of mydir to an external directory here.
3. Open the directory *h:\xml temp*.
4. Count how many XML files are in the directory. (There are four files in here.)
5. Read the XML file name without the extension .xml.
6. Rename the file name to *name1.....name4*.
7. List the XML Map *iqi_xml_SF2008.map* which was created from SAS XML Mapper.
8. Start reading from the first XML file.
9. *&&name&i* – first pass is *&name1..4*, then the second pass is *fname* which is the xml file name in the directory.
10. Library *xml1, xml2....* which contains the SAS data set *detail*.
11. Append the data set *detail* to the data set *detail* in *outfile*.
12. Delete the library *xml1....4*.
13. Close the directory.
14. State the name of directory in which the XML files are in.

CONCLUSIONS

SAS XML Mapper application is a user friendly, drag and drop interface. It will create an XML Map and SAS sample program for you in a simple manner. The macro program shown here is one of the ways that can be used when reading multiple xml files. This paper can be helpful to anyone needing a way to solve reading multiple xml files.

REFERENCES

- Miriam Cisternas, and Richardo Cisternas "Reading and Writing XML files from SAS". Paper 119-29. SUGI 29(2003), <http://www2.sas.com/proceedings/sugi29/119-29.pdf>
- Frederick Pratter. "XML for SAS Programmers". Paper 042-2008. SUGI 2008, <http://www2.sas.com/proceedings/forum2008/042-2008.pdf>

SAS® 9.2 XML Libname Engine: User's Guide.

ACKNOWLEDGEMENTS

Thank you to Chevell (SAS technical support) helped me read the external xml files.

CONTACT INFORMATION

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

Mayumi Takada
Data Analyst
Arkansas Foundation for Medical Care
Fort Smith, Arkansas
Work Phone: 479-573-7686
E-mail: mtakada@afmc.org

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.