Using SAS® software to shrink the Data used in Apache Flex® Application

Ahmed Al-Attar, AnA Data Warehousing Consulting LLC, McLean, VA

Abstract

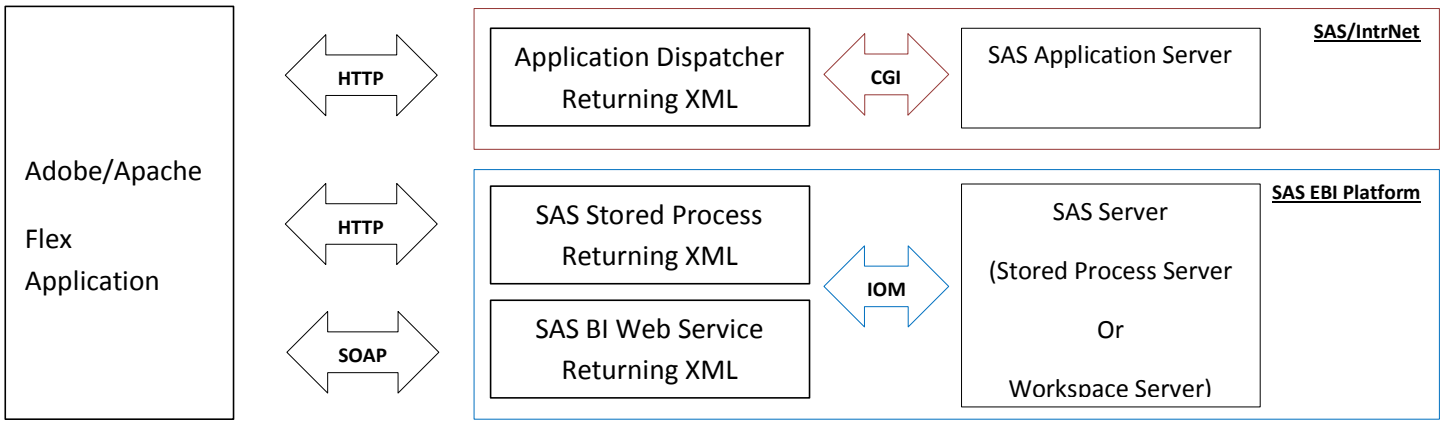
This paper discusses the techniques I used at the Census Bureau to overcome the issue of dealing with large amount of data while modernizing some of their public facing web applications by using Service Oriented Architecture (SOA) to deploy SAS powered Flex web applications. Techniques that resulted in reducing **142,293** XML lines (3.6 MB) down to **15,813** XML lines (1.8 MB) a **50%** size reduction on the server side (HTTP Response), and **196,167** observations down to **283** observations, a reduction of **99.8%** in summarized data on the client side (XML Lookup file).

Introduction

When the Dissemination Internet Staff (DIS) team at the Census Bureau decided to update some of their existing web applications and modernize them, they had to decide on a technology that could

- Integrate with SAS/IntrNet® in phase I
- Get embedded in HTML page
- Operate without Java Application Server
- Integrate with SAS Enterprise Business Intelligence (EBI) Platform in phase II

Based on the above, Adobe/Apache Flex® application framework was selected to provide the required Rich Interactivity, and Integration with SAS® 9.2 through Web Services standard protocols such as REST and SOAP.



While the client application (Flex) can communicate with SAS via submitting either HTTP request or SOAP envelop over HTTP, the response (data) is always returned as XML stream. That's where we started to have issues when dealing with large data!!

SAS continues to enhance and simplify the creation of web services and data conversion to XML via the XML Libname engine. Here is an example of exporting a SAS data set from one library to XML document in another directory/library.

```
LIBNAME sesug14 XML92 "C:\Projects\SESUG14";

PROC COPY IN=work OUT=sesug14;
        SELECT sample;
RUN;
```

The resulting XML would look like this:

```
1  <?xml version="1.0" encoding="windows-1252" ?>
2  <TABLE>
3     <SAMPLE>
4        <CCODE>AA</CCODE>
5        <GEOAREA>Not specified</GEOAREA>
6        <REFDATE>1993</REFDATE>
7        <SUBPOP>Prostitutes</SUBPOP>
8        <POPCODE>P</POPCODE>
9        <SEX>F</SEX>
10       <AGE>18Y</AGE>
11       <SOURCEID>D0178</SOURCEID>
12       <COMMENTS>Sample size (n) approximated was 280.  HIV1 and/or HIV2.</COMMENTS>
13       <DATACODE>P</DATACODE>
14       <COUNTRY>Aruba</COUNTRY>
15       <AUTHOR>De Windt, M., Dept. Salubridad Publica</AUTHOR>
16       <YEAR>1995</YEAR>
17       <TITLE>PAHO/WHO HIV Surveillance</TITLE>
18       <PUB_INFO>June 23, PAHO/WHO.</PUB_INFO>
19       <DATATYPE>HIV1,2</DATATYPE>
20       <NO_CASES>0</NO_CASES>
21       <NO_DEATHS>0</NO_DEATHS>
22       <PREV_RATE>0.00</PREV_RATE>
23       <INC_RATE>N/A</INC_RATE>
24       <SPEC_TYPE>B</SPEC_TYPE>
25       <TESTTYPE>ELISA*2</TESTTYPE>
26       <SAMPSIZE>N/A</SAMPSIZE>
27    </SAMPLE>
28    <SAMPLE>
29       <CCODE>AA</CCODE>
30       <GEOAREA>Not specified</GEOAREA>
31       <REFDATE>1994</REFDATE>
32       <SUBPOP>Prostitutes</SUBPOP>
33       <POPCODE>P</POPCODE>
34       <SEX>F</SEX>
35       <AGE>18Y</AGE>
36       <SOURCEID>D0178</SOURCEID>
37       <COMMENTS>Sample size (n) approximated was 280.  HIV1 and/or HIV2.</COMMENTS>
38       <DATACODE>P</DATACODE>
39       <COUNTRY>Aruba</COUNTRY>
40       <AUTHOR>De Windt, M., Dept. Salubridad Publica</AUTHOR>
41       <YEAR>1995</YEAR>
42       <TITLE>PAHO/WHO HIV Surveillance</TITLE>
43       <PUB_INFO>June 23, PAHO/WHO.</PUB_INFO>
44       <DATATYPE>HIV1,2</DATATYPE>
45       <NO_CASES>0</NO_CASES>
46       <NO_DEATHS>0</NO_DEATHS>
47       <PREV_RATE>0.00</PREV_RATE>
48       <INC_RATE>N/A</INC_RATE>
49       <SPEC_TYPE>B</SPEC_TYPE>
```

Three Tags/Lines added (two at top, One closing Tag at the end of the file)

Columns are transposed in Rows. Sigle Line in converted into multiple lines

Each Observation/Record is wraped by two tagts. Openning & Closing Tags

Examining the resulted output indicated “Total Resulted lines” = [(#Obs * #Columns) + (#Obs * 2) + 3]

	SAS Data Set		XML92 Engine		
Table Name	#Obs	Size in MB	#Lines	Size in MB	Real time
SGF2014.SAMPLE (23 Cols)	131,991	120	3,299,778	125	56.68 seconds

Note: While different SAS data set sizes and structures could result in different storage footprint and ratios, the formula used to generate the XML lines always the same!

That’s where this formula started causing issues especially when we had to transfer the resulted XML data across the network. While all the examples illustrated in the referenced papers used this standard XML output format, we had to find an alternative. Upon further investigation and research, I discovered Flex’s ability to consume XML Attributes[1] in

[1] XML Attribute: “A markup construct consisting of a name/value pair that exists within a start-tag or empty-element tag”. http://en.wikipedia.org/wiki/XML

addition to XML Elements[2]. So I went ahead and created custom SAS macro that would accept a data set name and creates XML output that utilizes XML Attributes instead of XML Elements.

```
FILENAME smplxml "C:\Projects\SESUG14\sample_cust.xml";

%util_reformatTable(p_inDsName=work_r.sample, p_newFormat=xml, p_outFileRef=smplxml, p_dispNumObs_yn=N
, p_numObs=, p_dispByteSize_yn=N, p_byteSize=);
```

The resulting XML would look like this:

```
<?xml version="1.0" encoding="windows-1252" ?>
<TABLE>
   <XMLRECORD CCODE="AA" GEOAREA="Not specified" REFDATE="1993" SUBPOP="Prostitutes" POPCODE="P" SEX="F" AGE="18Y" SOURCEID="D0
   <XMLRECORD CCODE="AA" GEOAREA="Not specified" REFDATE="1994" SUBPOP="Prostitutes" POPCODE="P" SEX="F" AGE="18Y" SOURCEID="D0
   <XMLRECORD CCODE="AA" GEOAREA="Not specified" REFDATE="2005" SUBPOP="Blood donors - volunteer" POPCODE="B" SEX="B" AGE="ALL"
   <XMLRECORD CCODE="AC" GEOAREA="Not specified" REFDATE="2004" SUBPOP="Blood donors - volunteer" POPCODE="B" SEX="B" AGE="ALL"
   <XMLRECORD CCODE="AC" GEOAREA="Not specified" REFDATE="2004-2005" SUBPOP="Prisoners" POPCODE="J" SEX="M" AGE="ALL" SOURCEID=
   <XMLRECORD CCODE="AC" GEOAREA="National" REFDATE="2004" SUBPOP="Pregnant women" POPCODE="G" SEX="F" AGE="ALL" SOURCEID="C064
   <XMLRECORD CCODE="AC" GEOAREA="National" REFDATE="2000" SUBPOP="Pregnant women" POPCODE="G" SEX="F" AGE="ALL" SOURCEID="C064
   <XMLRECORD CCODE="AC" GEOAREA="National" REFDATE="2003" SUBPOP="Pregnant women" POPCODE="G" SEX="F" AGE="ALL" SOURCEID="C064
   <XMLRECORD CCODE="AC" GEOAREA="National" REFDATE="2002" SUBPOP="Pregnant women" POPCODE="G" SEX="F" AGE="ALL" SOURCEID="C064
   <XMLRECORD CCODE="AC" GEOAREA="National" REFDATE="2005" SUBPOP="Pregnant women" POPCODE="G" SEX="F" AGE="ALL" SOURCEID="C064
   <XMLRECORD CCODE="AC" GEOAREA="National" REFDATE="2001" SUBPOP="Pregnant women" POPCODE="G" SEX="F" AGE="ALL" SOURCEID="C064
   <XMLRECORD CCODE="AC" GEOAREA="Not specified" REFDATE="1986-1990" SUBPOP="Prostitutes" POPCODE="P" SEX="F" AGE="ALL" SOURCEI
   <XMLRECORD CCODE="AC" GEOAREA="Not specified" REFDATE="1986-1988" SUBPOP="Prostitutes" POPCODE="P" SEX="F" AGE="ALL" SOURCEI
   <XMLRECORD CCODE="AE" GEOAREA="Not specified" REFDATE="2006" SUBPOP="TB pts." POPCODE="U" SEX="B" AGE="ALL" SOURCEID="W0359"
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2002" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="M1023" CO
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2005" SUBPOP="IVDU" POPCODE="I" SEX="B" AGE="ALL" SOURCEID="M1023" COMMENTS="
   <XMLRECORD CCODE="AF" GEOAREA="Herat Province" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2000" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="M1023" CO
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="M1023" CO
   <XMLRECORD CCODE="AF" GEOAREA="Nangarhar Province" REFDATE="2003" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCE
   <XMLRECORD CCODE="AF" GEOAREA="Mazar Province" REFDATE="2002" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2003" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="M1023" CO
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2001" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="M1023" CO
   <XMLRECORD CCODE="AF" GEOAREA="Nangarhar Province" REFDATE="2004" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCE
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2004" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="M1023" CO
   <XMLRECORD CCODE="AF" GEOAREA="Juzjan Province" REFDATE="2004" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID=
   <XMLRECORD CCODE="AF" GEOAREA="Nangarhar Province" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCE
   <XMLRECORD CCODE="AF" GEOAREA="Kunduz Province" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID=
   <XMLRECORD CCODE="AF" GEOAREA="Quandahar Province" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCE
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2005" SUBPOP="General population" POPCODE="N" SEX="B" AGE="ALL" SOURCEID="M10
   <XMLRECORD CCODE="AF" GEOAREA="Hirat" REFDATE="2008(?)" SUBPOP="IVDU" POPCODE="I" SEX="B" AGE="ALL" SOURCEID="N0504" COMMENT
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="2007-2009" SUBPOP="IVDU" POPCODE="I" SEX="M" AGE="ALL" SOURCEID="N0575" COMME
   <XMLRECORD CCODE="AF" GEOAREA="Not specified" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="N
   <XMLRECORD CCODE="AF" GEOAREA="Badakshan Province" REFDATE="2004" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCE
   <XMLRECORD CCODE="AF" GEOAREA="Seven cities" REFDATE="2005-2006" SUBPOP="Adult TB pts." POPCODE="U" SEX="B" AGE="ALL" SOURCE
   <XMLRECORD CCODE="AF" GEOAREA="Parwan Province" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID=
   <XMLRECORD CCODE="AF" GEOAREA="Parwan Province" REFDATE="2004" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID=
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="1997" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="T0417" CO
   <XMLRECORD CCODE="AF" GEOAREA="Juzjan Province" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID=
   <XMLRECORD CCODE="AF" GEOAREA="Khost Province" REFDATE="2003" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="
   <XMLRECORD CCODE="AF" GEOAREA="Mazar-e-Sharef Province" REFDATE="2005" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" S
   <XMLRECORD CCODE="AF" GEOAREA="Nangarhar Province" REFDATE="2001" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCE
   <XMLRECORD CCODE="AF" GEOAREA="Kabul" REFDATE="1996" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="T0417" CO
   <XMLRECORD CCODE="AF" GEOAREA="Faryab Province" REFDATE="2003" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID=
   <XMLRECORD CCODE="AF" GEOAREA="Mazar-e-Sharef Province" REFDATE="2003" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" S
   <XMLRECORD CCODE="AF" GEOAREA="Farah Province" REFDATE="2003" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID="
   <XMLRECORD CCODE="AF" GEOAREA="Kunduz Province" REFDATE="2003" SUBPOP="Blood donors" POPCODE="B" SEX="B" AGE="ALL" SOURCEID=
```

Here is how this new approach compares to the XML Libname engine.

	SAS Data Set		XML92 Engine			Custom XML		
Table Name	#Obs	Size in MB	#Lines	Size in MB	Real time	#Lines	Size in MB	Real time
SGF2014.SAMPLE (23 Cols)	131,991	120	3,299,778	125	56.68 seconds	**131,994**	**78**	**27.06 seconds**
SGF2014.SAIPESD (5 Cols)	196,167		1,373,172	31.25		**196,170**	**15.2**	
SGF2014.SAIPESNC_COUNTY (3 Cols)	56,528		282,643	6.8	20.26 seconds	**56,531**	**3**	**12:07 seconds**

Now that I have got this issue addressed on the server I had to deal with large data on the client side.

[2] XML Element: "A logical document component either begins with a start-tag and ends with a matching end-tag or consists only of an empty-element tag". http://en.wikipedia.org/wiki/XML

All deployed Flex web applications

- Follows the same layout which comprised of two parts
  - Data Filters Part: Two or more data selectors widgets with data driven XML lookup files
  - Data Viewer Part: Data visualization widget, such as Data Grids, Charts and Maps
- Performed client side data selection validations to avoid Zero result sets returned.
  This proved to be troublesome and required some attention and thinking outside of the box!

I have always used Proc Summary/Proc Means to figure out the unique combinations of variables values, but when the final number of unique combinations exceeds the hundreds and starts to range in the thousands, tens of thousands, and hundreds of thousands, it starts to fail to load during the client application initialization, which in turn causes the application to crash!!

Here are few examples of the variable combinations I had to deal with

Variables Set	# Unique Combinations	Notes
Year, State, County	56,528	Counties vary from one year to another
Year, State, School District	196,167	School District vary from one year to another
Year, State, County Flag, Age Category, Race Category, Gender Category, Income Category	71,196	Certain category values vary across years

I had no chance loading such amounts of unique combinations, and even if I could, processing them at run time, would have resulted in a very unsatisfactory user experience!

I had to find an alternative approach to the traditional OLAP approach in order to reduce the size of the combinations without affecting the integrity of the data and the relationship amongst the values of the variables. This is where the power of the SAS language came to the rescue, and provided me with straight forward processing techniques allowed me to achieve my goal.

Having

- SAS supports long character strings (32,767 chars)
- SAS provides first. & last. processing
- All the variables I had to deal with have relatively short code values
- Custom developed data combinations class in Adobe ActionScript

Allowed me to transpose and collapse particular variable values into a single space delimited string, and find unique combinations based on the newly created string value.

The following screen shots illustrate the data transformation.

VIEWTABLE: Work_r.Saipesd

	State Fips Code	District ID	Estimate Year	_TYPE_	_FREQ_
1	01	0100001	1995	7	1
2	01	0100001	1997	7	1
3	01	0100001	1999	7	1
4	01	0100001	2000	7	1
5	01	0100001	2001	7	1
6	01	0100001	2002	7	1
7	01	0100001	2003	7	1
8	01	0100001	2004	7	1
9	01	0100001	2005	7	1
10	01	0100001	2006	7	1
11	01	0100001	2007	7	1
12	01	0100001	2008	7	1
13	01	0100001	2009	7	1
14	01	0100001	2010	7	1
15	01	0100003	1995	7	1
16	01	0100003	1997	7	1
17	01	0100003	1999	7	1
18	01	0100003	2000	7	1
19	01	0100003	2001	7	1
20	01	0100003	2002	7	1

Standard Proc Summary output data set with ***196,167*** observations.

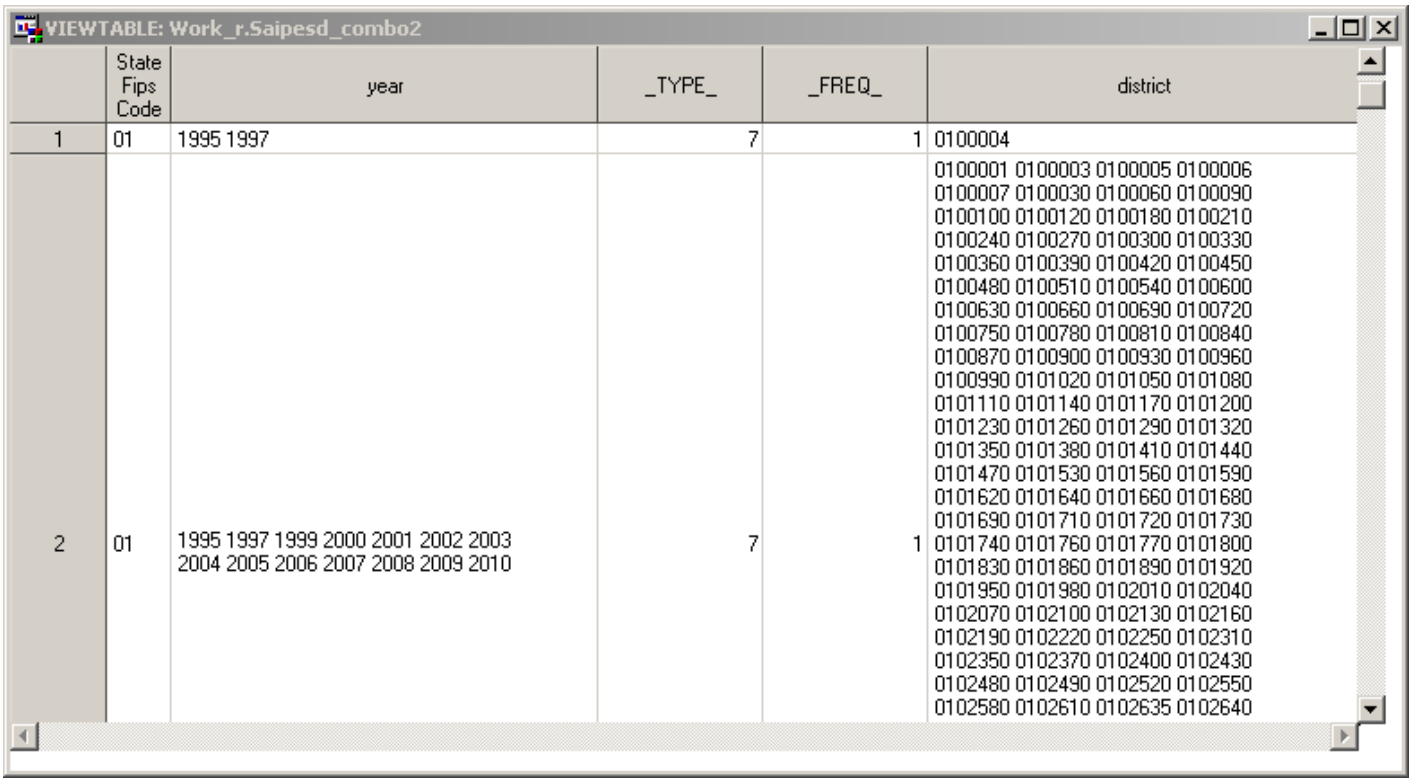
By transposing the Year values while maintaining the State and School District values, I got the following output

VIEWTABLE: Work_r.Saipesd_combo

	State Fips Code	District ID	_TYPE_	_FREQ_	year
1	01	0100001	7	1	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
2	01	0100003	7	1	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
3	01	0100004	7	1	1995 1997
4	01	0100005	7	1	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
5	01	0100006	7	1	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
6	01	0100007	7	1	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
7	01	0100008	7	1	1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
8	01	0100011	7	1	2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
9	01	0100012	7	1	2004 2005 2006 2007 2008 2009 2010
10	01	0100013	7	1	2005 2006 2007 2008 2009 2010
11	01	0100030	7	1	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
12	01	0100060	7	1	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
13	01	0100090	7	1	1995 1997 1999 2000 2001 2002 2003

*This resulted into reducing **196,167** observations down to **14,772** observations.*

Taking this one step further, by transposing School District values while maintaining the State and Year gave me this output



VIEWTABLE: Work_r.Saipesd_combo2

	State Fips Code	year	_TYPE_	_FREQ_	district
1	01	1995 1997	7	1	0100004
2	01	1995 1997 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010	7	1	0100001 0100003 0100005 0100006 0100007 0100030 0100060 0100090 0100100 0100120 0100180 0100210 0100240 0100270 0100300 0100330 0100360 0100390 0100420 0100450 0100480 0100510 0100540 0100600 0100630 0100660 0100690 0100720 0100750 0100780 0100810 0100840 0100870 0100900 0100930 0100960 0100990 0101020 0101050 0101080 0101110 0101140 0101170 0101200 0101230 0101260 0101290 0101320 0101350 0101380 0101410 0101440 0101470 0101530 0101560 0101590 0101620 0101640 0101660 0101680 0101690 0101710 0101720 0101730 0101740 0101760 0101770 0101800 0101830 0101860 0101890 0101920 0101950 0101980 0102010 0102040 0102070 0102100 0102130 0102160 0102190 0102220 0102250 0102310 0102350 0102370 0102400 0102430 0102480 0102490 0102520 0102550 0102580 0102610 0102635 0102640

*This resulted in reducing **14,772** observations down to **283** observations only. Bingo!*

Applying the same techniques against the other combination tables yielded the following results

Variables Set	# Unique Combinations	# Unique Combinations after Transposing Values	%Reduction
Year, State, County	56,528	**62** *[64 XML lines -25K]*	**99.89%**
Year, State, School District	196,167	**283** *[285 XML lines-134K]*	**99.86%**
Year, State, County Flag, Age Category, Race Category, Gender Category, Income Category	71,196	**328** *[330 XML lines-84K]*	**99.54%**

With combinations in such low numbers, we were able to maintain optimum application initialization and run time processing.

Conclusion

Working with large data sets often requires adoption of alternative techniques beyond compression and other standard out of the box functionalities provided by SAS.

I would strongly encourage the reader to think outside of the box and find ways to innovate. After all, developing custom solutions can sometime be frustrating and demanding, but when they work, they can be very rewarding.

References

Dan Jahn, 2006. "Service-Oriented Architectures – Going from Buzz to Business". Proceedings of the 31st Annual SAS Users Group International Conference. Cary, NC: SAS Institute Inc. Available at http://www2.sas.com/proceedings/sugi31/001-31.pdf

Klenz, B., and Jahn, D. 2008. "Creating Web Services Using SAS Analytics". Proceedings of the SAS Global Forum 2008 Conference. Cary, NC: SAS Institute Inc. Available at http://www2.sas.com/proceedings/forum2008/010-2008.pdf.

Jahn, D., and Klenz, B. 2009. "Tips and Techniques for Analytic Web Services". Proceedings of the SAS Global Forum 2009 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings09/004-2009.pdf

Joe Flynn, 2010. "Flex Your SAS® Muscle". Proceedings of the SAS Global Forum 2010 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings10/015-2010.pdf

Contact Information

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

Ahmed Al-Attar
AnA Data Warehousing Consulting, LLC.
McLean, VA 22101
Cell Phone: 703-477-7972
E-mail: ahmed.al-attar@anadwc.com
Web: www.anadwc.com

Trademarks

APPENDIX 1

```
%MACRO shrinkCombos(
  p_inDsName=          /* Input Source Data Source */
, p_outDsName=         /* Output Data Set */
, p_byClause=          /* Classification Variables Set */
, p_classVar=          /* Class Variable to Transpose */
, p_prntClassVar=      /* Preceding Class Variable */
, p_aggrVarLen=10      /* Variable length of the new transposed values */
);

        /* Summarize & Sort */
        PROC SUMMARY DATA=&p_inDsName NWAY;
                CLASS   &p_byClause;
                OUTPUT OUT=&p_outDsName;
        RUN;

        /* Process the data and generate the transposed value */
        DATA &p_outDsName(DROP=&p_classVar RENAME=(aggr=&p_classVar));
                SET &p_inDsName;
                BY &p_byClause;

                LENGTH aggr $&p_aggrVarLen ;
                RETAIN aggr ;

                IF (FIRST.&p_prntClassVar) THEN
                        aggr='';

                aggr = catx(' ',aggr,&p_classVar);

                IF (LAST.&p_prntClassVar) THEN
                        OUTPUT;
        RUN;
%MEND shrinkCombos;

/* Usage Examples */
%shrinkCombos( p_inDsName=saipe.Saipeschldstrct, p_outDsName=work.saipesd_combo
, p_byClause=%str(state district year), p_classVar=year, p_prntClassVar=district, p_aggrVarLen=120);

/* Max District Count by State = 1045, --> 1045*8=8400 */
%shrinkCombos( p_inDsName=work.saipesd_combo, p_outDsName=work.saipesd_combo2
, p_byClause=%str(state year district), p_classVar=district, p_prntClassVar=year, p_aggrVarLen=8400);

filename saipesd "C:\Projects\SESUG14\saipesd.xml";

/* Export to XML*/
%util_reformatTable(p_inDsName=work.saipesd_combo2, p_newFormat=xml, p_outFileRef=saipesd
, p_dispNumObs_yn=N, p_numObs=, p_dispByteSize_yn=N, p_byteSize=);
```