

Regulatory Compliance Reporting Using SAS® XML Mapper

Sarita Kannarath, Nishant Sharma, and Phil Hanna, SAS Institute Inc.
and Amitkumar Nakrani

ABSTRACT

As a part of regulatory compliance requirements, institutions are required to submit reports based on Microsoft® Excel, as per templates supplied by the regulators. This poses several challenges, including the high complexity of templates, the fact that implementation using ODS can be cumbersome, and the difficulty in keeping up with regulatory changes and supporting dynamic report content. At the same time, you need the flexibility to customize and schedule these reports as per your business requirements. This paper discusses an approach to building these reports using SAS® XML Mapper and the Excel XML spreadsheet format. This approach provides an easy-to-use framework that can accommodate template changes from the regulators without needing to modify the code. It is implemented using SAS® technologies, providing you the flexibility to customize to your needs. This approach also provides easy maintainability.

INTRODUCTION

The purpose of this paper is to introduce a technique to render Microsoft Excel reports with data by conforming to a predefined template, without putting any effort in re-creating the Microsoft Excel style information. The programmer can therefore only focus on getting the data right, from relevant data sources instead of the rendering aspects.

ABOUT THE TECHNIQUE

Various approaches were tried out before the suggested technique of report rendering was chosen. These approaches are listed below:

1. Apache POI: Apache POI is an open-source library that helps developers create Microsoft Office artifacts. They are available as APIs in Java. Due to this, the SAS program needs to depend on a Java execution environment for generation of the report. This would also require the exchange of a significant amount of data between SAS and a Java virtual machine.
2. SAS ODS: The SAS ODS library provides support for rendering Microsoft Excel documents from SAS. This works well for rendering a data set in a simple Microsoft Excel file. The regulatory templates are fairly complex formats and require a lot of formatting. Hence, the ODS code required to render these reports becomes bulky and difficult to maintain.
3. Token substitution in XML files: In this approach, the Microsoft Excel file is annotated with the keywords that would be replaced by data. Subsequently, the file is saved as XML. Later on, this XML is loaded into a data set as a set of strings and the tokens are substituted by actual values. This approach works well for fixed format reports. There are some reports that are dynamic in nature, that is, the number of data elements and sheets in the Microsoft Excel file are not known beforehand. Due to this limitation, this approach could not be pursued.

The last approach led to the evolution of the approach that will be described in this paper in the sections below.

HOW DOES IT WORK?

We must understand the following before we move ahead with understanding the complete rendering mechanism. The description of the following is only limited to the scope of describing the report rendering technique.

Understanding the Building Blocks

Microsoft Excel's XML Tag Sets

Microsoft Excel has provided XML tag sets for XML documents that can be opened in Microsoft Excel. The link, <http://msdn.microsoft.com/en-us/library/aa140066>, describes the various tag sets. If a Microsoft Excel document is saved as "XML Spreadsheet 2003 (*.xml)," then the file is saved as an XML document and appears just like any other XML document.

SAS XML LIBNAME Engine

The SAS XML LIBNAME translates the input XML document to the SAS proprietary file format. To import an XML document, the following LIBNAME statement for the XML engine should be executed in order to assign a libref to the physical location of an existing XML document. Then, SAS code to access the XML document as a SAS data set should be executed as shown in the following example where contents of an XML file are copied to the SAS WORK library as SAS data sets.

```
libname myxmlfil xml "C:\input_file.xml";  
proc copy in=myxmlfil out=work;  
run;
```

The XML engine imports only XML documents that conform to the markup types supported in the XMLTYPE= option. Attempting to import free-form XML documents that do not conform to the specifications required by the supported markup types will generate errors. To successfully import files that do not conform to the XMLTYPE= markup types, create a separate XML document, called an *XMLMap*.

If the XML document does not import successfully, then XMLMap can be used to indicate the rules for importing the XML. The XMLMap syntax tells the XML engine how to interpret the XML markup into a SAS data set or data sets, variables (columns), and observations (rows).

The following example illustrates how to import an XML document using the AUTOMAP= option to automatically generate an XMLMap file. By specifying the AUTOMAP= option in the LIBNAME statement, SAS analyzes the structure of the specified XML document and generates XMLMap syntax that describes how to interpret the XML markup into a SAS data set or data sets, variables (columns), and observations (rows).

Here is the XML document, NHL.XML, to be imported. Importing the document without an XMLMap will cause an error indicating that the data is not in a supported format.

```
<?xml version="1.0" encoding="iso-8859-1" ?>  
<NHL>  
  <CONFERENCE> Eastern  
    <DIVISION> Southeast  
      <TEAM name="Thrashers" abbrev="ATL" />  
      <TEAM name="Hurricanes" abbrev="CAR" />  
      <TEAM name="Panthers" abbrev="FLA" />  
      <TEAM name="Lightning" abbrev="TB" />  
      <TEAM name="Capitals" abbrev="WSH" />  
    </DIVISION>  
  </CONFERENCE>  
  <CONFERENCE> Western  
    <DIVISION> Pacific  
      <TEAM name="Stars" abbrev="DAL" />  
      <TEAM name="Kings" abbrev="LA" />  
      <TEAM name="Ducks" abbrev="ANA" />  
      <TEAM name="Coyotes" abbrev="PHX" />  
      <TEAM name="Sharks" abbrev="SJ" />  
    </DIVISION>  
  </CONFERENCE>
```

</NHL>

The following SAS statements import the XML document named NHL.XML.

```
filename NHL 'C:\My Documents\XML\NHL.xml';  
filename MAP 'C:\My Documents\XML\NHLGenerate.map';  
libname NHL xmlv2 automap=replace xmlmap=MAP;  
proc print data=NHL.TEAM;  
run;
```

The XML document can be created using a DATA step to write to a plain text external file. If the XML conforms to Microsoft Excel's tag sets, then the XML document can be opened in Microsoft Excel just like any other spreadsheet.

Putting Them Together to Form the Rendering Mechanism

The following diagram describes how the above mentioned aspects are combined to form a rendering mechanism for Microsoft Excel based XML reports.

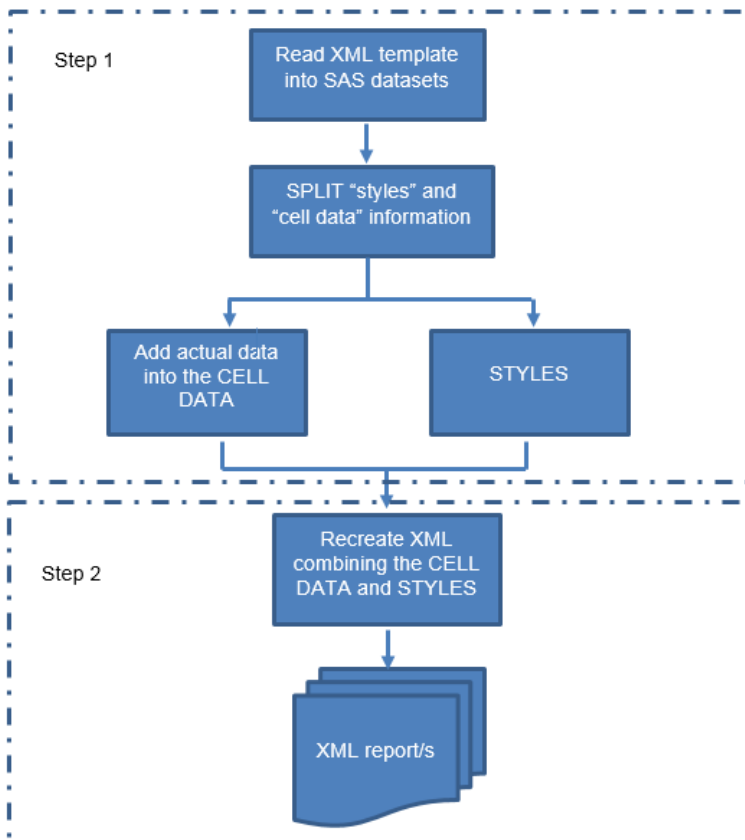


Figure 1. Diagram Depicting the Report Rendering Mechanism

Describing the Inputs to Create the Report

1. The XML Report Template

The report template is an XML document with the Microsoft Excel tag sets. The titles, styling, column headings, and aspects that describe the report (other than the data to be filled in the table cells) are all part of the report template. The final report will look exactly like the report template with the cells filled in with report data and with additional data driven worksheet insertions or row insertions.

The following figure depicts a sample report template. Let us call this template “report_template.xml.” This sample report template has two sheets, “Table 1” and “Table 2.” For more information about this template, see REPORT_TEMPLATE.XML in the Appendix section.

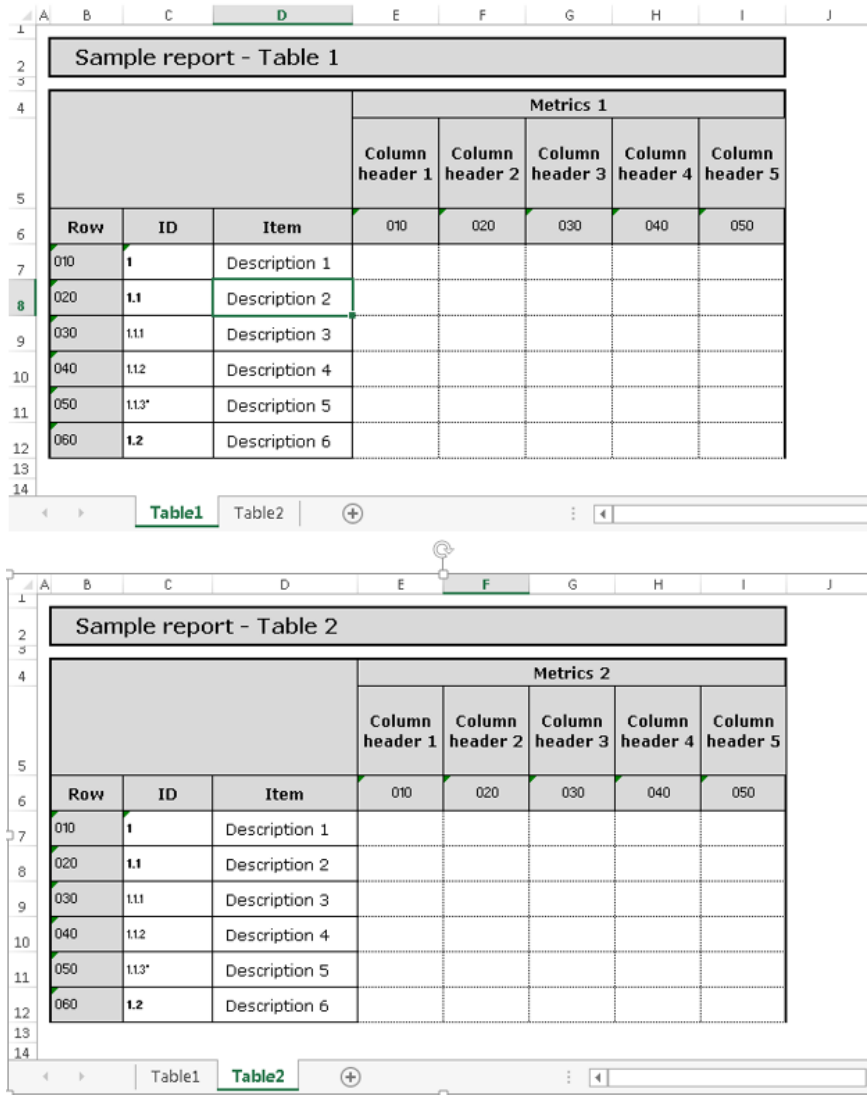


Figure 2 Report Template XML Document When Opened in Microsoft Excel

2. The Data to Be Reported

The data to be put in the cells of the report are to be placed in a data set with the following columns.

Variable name	Description	SAS Variable Format	Data Type
WORKSHEET	The worksheet name as per the report template.	\$32.	CHARACTER

Variable name	Description	SAS Variable Format	Data Type
SUB_SHEET	The worksheet name as per the final report. In case the worksheet has to be replicated, this column has to be filled with the name for each new sheet against the cell coordinates and values for each sheet. This enables the capability of replicating worksheets if need be.	\$32.	CHARACTER
ROW	The row corresponding to the Excel row. In case a row has to be inserted, then this column takes the value of the last row before insertion plus 0.1.	8.	NUMERIC
COL	The column corresponding to the Excel column.	8.	NUMERIC
VALUE	The data to be reported in the cell. This can take any alpha-numeric value.	\$255.	CHARACTER

Table 1 Report Data Column Description

For example, the following table contains the data to be inserted in the sample report that will be generated using the template “report_template.xml.” As per this data, the final report will have sheets “T11” and “T12” that are based on sheet “Table1” and sheet “T21” that is based on sheet “Table2” of “report_template.xml.” Values will be inserted in the row and column combination of respective sheets as shown below. A row will be inserted after row 10 and it will follow the styling information of row 10. Let us call this data set, “report_data.” For more information, see “SAS Code to Generate the SAS Data Set “REPORT_DATA”” on page 15.

WORKSHEET	SUB_SHEET	ROW	COL	VALUE
Table1	T11	7	5	1234
Table1	T11	8	6	456
Table1	T12	9	6	4567
Table2	T21	10	5	8956
Table2	T21	10.1	8	789

Table 2 Sample Report Data

Creating the Report from the Inputs

To make use for the rendering technique, a wrapper SAS program needs to be written that will contain the following steps:

1. The macros to render the report should be included.

A set of macros will have to be written following the flow explained in Figure 1. Diagram Depicting the Report Rendering Mechanism. The main process of re-creating templates with

data will be handled by these macros. The macros can be included in the following manner, assuming that they are saved in "C:\ExcelXMLReporting."

Assuming that all the code saved in this location are only macro definitions and no macro calls, all macros in the current SAS session can be included safely using the following statement.

```
%include "C:\ExcelXMLReporting\macros\*.sas";
```

2. Define the parameters.

Assuming that the report template is saved in "C:\ExcelXMLReporting\template" and the report has to be generated in "C:\ExcelXMLReporting\report," the following parameters should be defined.

```
/*Path where the templates are stored*/  
%let template_path=C:\ExcelXMLReporting\template;  
/*Template file name*/  
%let template_sheet_name=report_template.xml;  
/*Path where the report needs to be generated*/  
%let report_path=C:\ExcelXMLReporting\report;  
/*Name of final report*/  
%let report_nm=test_report;  
/*Location where the report data can be found*/  
%let datamaplocation=C:\ExcelXMLReporting\datamap;  
/*The data to be reported*/  
%let datamap=report_data;
```

3. For more information, see "SAS Code to Generate the SAS Data Set "REPORT_DATA"" on page 15. Invoke the macro to create the report.

The report generation is triggered with the following macro call:

```
%recreate_wkbk;
```

The final report should be available in the location and name specified after a successful completion of this macro execution.

RECREATE_WKBK

RECREATE_WKBK is a SAS macro. This macro executes the steps described in Figure 1. Diagram Depicting the Report Rendering Mechanism.

1. Read the XML template into SAS data sets.
 - a. Connect to the report template using the XMLV2 engine.
 - b. Use the automap option to create XMLMaps dynamically for any given template and save it in a temporary file "inp_map" as shown in the code snippet below.
 - c. Copy the content of the template XML as a SAS data set to the SAS WORK library.

The following code creates an XMLV2 engine based SAS library called "input" that connects to the report template (that is, "&template_path.\&template_sheet_name") that resolves to "C:\ExcelXMLReporting\template\report_template.xml" via an automatically generated XMLMap saved in a temporary file referenced by fileref "inp_map."

```
filename inp_map temp;
```

```
libname input xmlv2 "&template_path.\&template_sheet_name"
automap=replace xmlmap=inp_map;
```

The contents of the "INPUT" library are copied to the "WORK" library using the "COPY" procedure:

```
proc copy in=input out=tmp1;
run;
```

The following is a snapshot of how the contents of the template XML (in this case the report_template.xml) will look like.

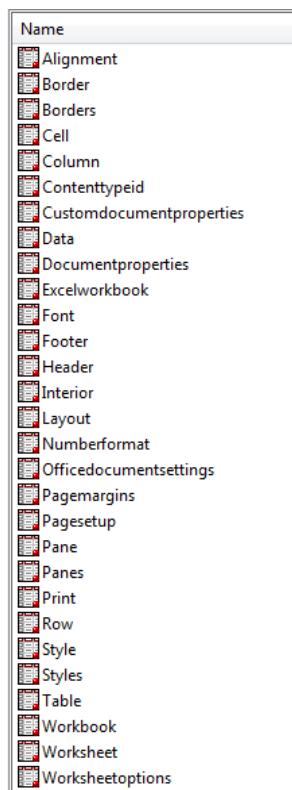


Figure 3 Contents of REPORT_TEMPLATE.XML

2. SPLIT "styles" and "cell data" information.

The content of the template XML is a set of SAS data sets. These data sets are copied to the SAS WORK library. The library can be organized into two distinct data sets: one containing the "STYLES" information and the other containing "CELL DATA" information. The "CELL DATA" information at this stage will contain the values that are present in the XML template cells and the association with the styles information. The values for the cells should be later replaced by the values that are to be reported.

The following is a snapshot of the columns present in the "STYLES" data.

COLUMN
id
style_ordinal
Parent
AlignmentHorizontal
AlignmentVertical
AlignmentWrapText
Borders
FontName
FontColor
FontBold
FontSize
NumberFormat
InteriorColor
Left
LeftLineStyle
LeftWeight
LeftColor
Right
RightLineStyle
RightWeight
RightColor
Top
TopLineStyle
TopWeight
TopColor
Bottom
BottomLineStyle
BottomWeight
BottomColor
AlignmentIndent
AlignmentReadingOrder
AlignmentRotate
AlignmentShrinkToFit
AlignmentVerticalText
FontItalic
FontOutline
FontShadow
FontStrikeThrough
FontUnderline
FontVerticalAlign
FontCharSet
FontFamily
InteriorPattern
InteriorPatternColor
ProtectionProtected
ProtectionHideFormula
Style_Name
Styles_ORDINAL
Font_Family

Figure 4 Columns in STYLE Data

The following is a snapshot of the STYLE data.

	id	style_ordinal	Parent	AlignmentHorizontal	AlignmentVertical	AlignmentWrapText	Borders	FontName	FontColor	FontBold	FontSize	NumberFormat	InteriorColor
1	Default	1			Bottom			Calibri	#000000		11		
2	s62	2			Bottom			Calibri	#000000		11		
3	m11467140	3		Left	Center		1	Verdana		1	14		#D9D9D9
4	m11467144	4		Center	Center		1	Verdana		1	11		#D9D9D9
5	m11467182	5		Left	Center		1	Verdana		1	14		#D9D9D9
6	m11467186	6		Center	Center		1	Verdana		1	11		#D9D9D9
7	s63	7			Center			Verdana			11		
8	s64	8	Center		Center			Verdana			11		
9	s72	9			Center			Verdana			11		FFFFFF
10	s73	10	Center		Center		1	Verdana		1	11		FFFFFF
11	s75	11	Center		Center		1	Verdana		1	11		#D9D9D9
12	s85	12			Center			Verdana			11		#D9D9D9
13	s86	13			Center			Verdana			11		#D9D9D9
14	s87	14	Center		Center		1	Verdana		1	11		#D9D9D9
15	s88	15	Center		Center		1	Verdana		1	11		#D9D9D9
16	s90	16	Center		Center			Verdana			9 @		#D9D9D9
17	s91	17	Center		Center		1	Verdana			9 @		#D9D9D9
18	s92	18	Left		Center			Verdana			9 @		#D9D9D9
19	s93	19	Left		Center			Verdana		1	8 @		
20	s94	20	Left		Center		1	Verdana			11		
21	s95	21			Center		1	Verdana			11		
22	s96	22			Center		1	Verdana			11		
23	s97	23			Center		1	Verdana			11		
24	s98	24	Left		Center			Verdana		1	8		
25	s99	25			Center		1	Verdana			11		
26	s100	26			Center		1	Verdana			11		
27	s101	27			Center		1	Verdana			11		
28	s102	28	Left		Center			Verdana			8		

Figure 5 Snapshot of STYLE Data

The following is a snapshot of the columns in CELL DATA.

COLUMN
r_index
r_hgt
row_autohgt
row_style
row_sp
column_autofitwidth
column_width
column_styleid
Worksheet_ORDINAL
Worksheet_Name
Table_ORDINAL
Column_ORDINAL
row
style
col
cell_index
type
value
MergeAcross
MergeDown
CellFormula
comment_author
Table_ExpandedColumnCount
Table_ExpandedRowCount
Table_FullColumns
Table_FullRows
Table_DefaultRowHeight
Table_StyleID
Table_DefaultColumnWidth
Column_Index
Column_Span
Cell_ORDINAL
Row_AutoFitHeight
Row_Height
Row_StyleID

Figure 6 Columns in CELL DATA Data

The following is a snapshot of the CELL DATA data.

Worksheet_ORDINAL	Worksheet_Name	Table_ORDINAL	row	col	Column_ORDINAL	cell_index	Cell_ORDINAL	value	Table_ExpandedColumnCount	Table_ExpandedRowCount	Table_FullColumns	Table
1	Table1	1	0		1				"g"	"12"	"1"	"1"
1	Table1	1	0		2				"g"	"12"	"1"	"1"
1	Table1	1	0		3				"g"	"12"	"1"	"1"
1	Table1	1	0		4				"g"	"12"	"1"	"1"
1			1						"g"	"12"	"1"	"1"
1			2	1			2	1 Sample report - Table 1	"g"	"12"	"1"	"1"
1			3	1			2	2	"g"	"12"	"1"	"1"
1			3	2			3	3	"g"	"12"	"1"	"1"
1			3	3			4	4	"g"	"12"	"1"	"1"
1			3	4			5	5	"g"	"12"	"1"	"1"
1			3	5			6	6	"g"	"12"	"1"	"1"
1			3	6			7	7	"g"	"12"	"1"	"1"
1			3	7			8	8	"g"	"12"	"1"	"1"
1			3	8			9	9	"g"	"12"	"1"	"1"
1			4	1			2	10	"g"	"12"	"1"	"1"
1			4	2				11 Metrics 1	"g"	"12"	"1"	"1"
1			5	1			2	12	"g"	"12"	"1"	"1"
1			5	2				13	"g"	"12"	"1"	"1"
1			5	3				14	"g"	"12"	"1"	"1"
1			5	4				15 Column header 1	"g"	"12"	"1"	"1"
1			5	5				16 Column header 2	"g"	"12"	"1"	"1"
1			5	6				17 Column header 3	"g"	"12"	"1"	"1"
1			5	7				18 Column header 4	"g"	"12"	"1"	"1"
1			5	8				19 Column header 5	"g"	"12"	"1"	"1"
1			6	1			2	20 Row	"g"	"12"	"1"	"1"
1			6	2				21 ID	"g"	"12"	"1"	"1"
1			6	3				22 Item	"g"	"12"	"1"	"1"
1			6	4				23 010	"g"	"12"	"1"	"1"
1			6	5				24 020	"g"	"12"	"1"	"1"
1			6	6				25 030	"g"	"12"	"1"	"1"
1			6	7				26 040	"g"	"12"	"1"	"1"
1			6	8				27 050	"g"	"12"	"1"	"1"
1			7	1			2	28 010	"g"	"12"	"1"	"1"
1			7	2				29 1	"g"	"12"	"1"	"1"
1			7	3				30 Description 1	"g"	"12"	"1"	"1"
1			7	4				31	"g"	"12"	"1"	"1"
1			7	5				32	"g"	"12"	"1"	"1"
1			7	6				33	"g"	"12"	"1"	"1"
1			7	7				34	"g"	"12"	"1"	"1"
1			7	8				35	"g"	"12"	"1"	"1"
1			8	1			2	36 020	"g"	"12"	"1"	"1"
1			8	2				37 1.1000000000000001	"g"	"12"	"1"	"1"
1			8	3				38 Description 2	"g"	"12"	"1"	"1"
1			8	4				39	"g"	"12"	"1"	"1"
1			8	5				40	"g"	"12"	"1"	"1"
1			8	6				41	"g"	"12"	"1"	"1"
1			8	7				42	"g"	"12"	"1"	"1"
1			8	8				43	"g"	"12"	"1"	"1"

Figure 7 Snapshot of CELL DATA Data

3. Add actual data into the CELL DATA.

The actual values that are to be reported in the final report override the values present in the "CELL DATA" data. The association with the corresponding styles in the "STYLES" data remain untouched.

During this process of overriding the values, the data to be reported, might demand row insertions and sheet insertions.

- a. Row insertion: If there are data driven row insertions, then the "STYLE" data of the reference row is considered for the additional rows. For example, if a row is being inserted after row 10, then the extra row coordinate will be numbered "10.n" in the data to be reported, where n is an integer that represents the increment from row 10 (that is, 10.1 would mean the 1st row inserted after row 10 following the attributes of row 10). The association of the "STYLE" data with the rows in the "CELL DATA" data are modified accordingly.
- b. Worksheet insertion: If there are worksheet insertions following the template details of the current sheet, then the data is mapped accordingly. The data to be reported needs to have the SUB_SHEET column filled to indicate if the current sheet of the template has to be replicated for the data value and cell coordinate. If the current worksheet of the template does not need any replication, then the SUB_SHEET will take the same value as the WORKSHEET column. The association of the "STYLE" data with the WORKSHEETS in the "CELL DATA" data are modified accordingly.

4. Re-create XML by combining the CELL DATA data and STYLES data.

The data from the CELL DATA data and the associated STYLES data is written to a plain text file with an .XML extension using the Microsoft Excel XML tag sets for individual worksheets. Style information corresponding to the workbook and individual worksheets are also written to this file.

This step uses a set of DATA steps to build the output XML file one section at a time. Note that in all but the first DATA step, the "mod" option must be specified in the File statement, so that lines will be appended to the output XML file, rather than replacing it.

The following code snapshot demonstrates how the output XML file that is referenced as "outputxml" is being created in the first DATA step and then in the next DATA step lines related to the STYLES DATA are being added following the Microsoft Excel's XML tag sets.

```

data _NULL_;
  file outputxml;
  put '<?xml version="1.0" encoding="UTF-8"?>';
  /* This line causes Excel to recognize the XML as an Excel workbook */
  put '<?mso-application progid="Excel.Sheet"?>';
  /* Write <Workbook> element start tag */
  put '<ss:Workbook';
  /* Must declare these namespaces if we use the prefixes anywhere
     on the page. */
  put 'xmlns="urn:schemas-microsoft-com:office:spreadsheet"';
  put 'xmlns:x="urn:schemas-microsoft-com:office:excel"';
  put 'xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"';
  put 'xmlns:html="http://www.w3.org/TR/REC-html40"';
  /* Close the tag */
  put '>';
run;
data _NULL_;
  file outputxml mod;
  set STYLES DATA end=eof;
  /******
  * Before handling the first observation in the styles data set,
  * write <ss:Styles> element start tag
  ******/
  if (_N_ = 1) then do;
    put '<ss:Styles>';
  end;
  /******
  * Write the <ss:Style> element start tag, including the
  * required ss:ID attribute and possibly a ss:Parent attribute
  ******/
  put '<ss:Style' @;
  put ' ss:ID=' ''' id +(-1) ''' @;
  if (not missing(parent)) then do;
    put ' ss:Parent=' ''' parent +(-1) ''' @;
  end;
  put '>';
  /******
  * Write the <ss:Alignment> tag, if any alignment parameters
  * were specified
  ******/

  if ( not missing(AlignmentHorizontal) or
      not missing(AlignmentVertical) or
      not missing(AlignmentWrapText) or
      not missing(AlignmentIndent) or
      not missing(AlignmentReadingOrder) or
      not missing(AlignmentRotate) or
      not missing(AlignmentShrinkToFit) or
      not missing(AlignmentVerticalText)) then do;

    put '<ss:Alignment' @;
    if (not missing(AlignmentHorizontal)) then do;
      put ' ss:Horizontal=' ''' AlignmentHorizontal +(-1) ''' @;
    end;
    if (not missing(AlignmentVertical)) then do;
      put ' ss:Vertical=' ''' AlignmentVertical +(-1) ''' @;
    end;
    if (not missing(AlignmentWrapText)) then do;
      put ' ss:WrapText=' ''' AlignmentWrapText +(-1) ''' @;
    end;
  end;

```

```

end;
if (not missing(AlignmentReadingOrder)) then do;
  put ' ss:ReadingOrder=' ''' AlignmentReadingOrder +(-1) ''' @;
end;
if (not missing(AlignmentRotate)) then do;
  put ' ss:Rotate=' ''' AlignmentRotate +(-1) ''' @;
end;
if (not missing(AlignmentShrinkToFit)) then do;
  put ' ss:ShrinkToFit=' ''' AlignmentShrinkToFit +(-1) ''' @;
end;
if (not missing(AlignmentVerticalText)) then do;
  put ' ss:VerticalText=' ''' AlignmentVerticalText +(-1) ''' @;
end;
if (not missing(AlignmentIndent)) then do;
  put ' ss:Indent=' ''' AlignmentIndent +(-1) ''' @;
end;

put '/>';
end;

```

The CELL DATA is also added to the final output XML file following Microsoft Excel's XML tag sets in similar manner.

Once the output XML file is created in this manner, it can be opened in Microsoft Excel and it will look exactly same as the template XML, but the cells will have the actual values to be reported.

The following is a snapshot of the final report that was generated using the REPORT_TEMPLATE.XML and the input data described in Table 2 Sample Report Data.

Sample report - Table 1							
			Metrics 1				
			Column header 1	Column header 2	Column header 3	Column header 4	Column header 5
Row	ID	Item	010	020	030	040	050
010	1	Description 1	1234				
020	1.1	Description 2		456			
030	1.1.1	Description 3					
040	1.1.2	Description 4					
050	1.1.3	Description 5					
060	1.2	Description 6					

Sample report - Table 1							
			Metrics 1				
			Column header 1	Column header 2	Column header 3	Column header 4	Column header 5
Row	ID	Item	010	020	030	040	050
010	1	Description 1					
020	1.1	Description 2					
030	1.1.1	Description 3		4567			
040	1.1.2	Description 4					
050	1.1.3	Description 5					
060	1.2	Description 6					

Sample report - Table 2							
			Metrics 2				
			Column header 1	Column header 2	Column header 3	Column header 4	Column header 5
Row	ID	Item	010	020	030	040	050
010	1	Description 1					
020	1.1	Description 2					
030	1.1.1	Description 3					
040	1.1.2	Description 4	8956				
						789	
050	1.1.3	Description 5					
060	1.2	Description 6					

Figure 8 Final Report XML

CONCLUSION

The methodology described in this paper will definitely provide a sigh of relief to SAS programmers who otherwise would have to think of writing complex SAS macros to address individual reporting needs for Microsoft Excel based reports.

The cell, worksheet, and workbook attributes are automatically taken care of by the macros used in this methodology. All a SAS programmer will have to do is get the report template prepared and have the report data and corresponding cell coordinates defined.

Data driven row insertions and sheet insertions will be taken care of automatically using this methodology.

The complete set of macros are provided in the Appendix. The way in which this methodology has to be used is already described in "Creating the Report from the Inputs." The "RECREATE_WKBK" macro call replaces the steps where a SAS programmer might have to call the ODS rendering steps.

REFERENCES

Stowe, Michael. 2001. XML Spreadsheet Reference. Available <http://msdn.microsoft.com/en-us/library/aa140066>. Accessed January 15, 2015.

SAS Institute Inc. 2013. SAS® 9.4 XML LIBNAME Engine User's Guide. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/documentation/cdl/en/engxml/64990/PDF/default/engxml.pdf>.

ACKNOWLEDGMENTS

Many thanks to Rajesh Khanwelkar, Amol Kokane, Aniruddha Kulkarni, Ram Mohan Dhulipalla, Priya Khirodkar and Taniya Raina for providing the valuable inputs and ideas while we came up with the technique described in this paper.

RECOMMENDED READING

SAS Institute Inc. 2013. *Base SAS® 9.4 Procedures Guide*. 3d ed. Cary, NC: SAS Institute Inc. Available <http://support.sas.com/documentation/cdl/en/proc/67327/PDF/default/proc.pdf>.

McDaniel, Stephen, and Hemedinger, Chris. 2010. *SAS® For Dummies®*. 2d ed. New York: John Wiley Sons Inc.

CONTACT INFORMATION

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

Sarita Kannarath
SAS Research and Development (India) Pvt. Ltd.
Level 3, Tower 5, Cybercity, Magarpatta City, Pune, India-411013
Sarita.kannarath@sas.com

Nishant Sharma
SAS Research and Development (India) Pvt. Ltd.
Level 3, Tower 5, Cybercity, Magarpatta City, Pune, India-411013
Nishant.Sharma@sas.com

Amitkumar Nakrani
mramitnakrani@gmail.com

Phil Hanna
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Phil.Hanna@sas.com
<http://www.sas.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.

APPENDIX

SAS Code to Generate the SAS Data Set "REPORT_DATA"

```
libname data_s "C:\ExcelXMLReporting\datamap";
data data_s.report_data;
  attrib worksheet length=$32.;
  attrib sub_sheet length=$32.;
  attrib row length=8.;
  attrib col length=8.;
  attrib value length=$255.;
  infile cards;
  input worksheet $ sub_sheet $ row col value $;
  cards;
Table1      T11  7    5    1234
Table1      T11  8    6    456
Table1      T12  9    6    4567
Table2      T21  10   5    8956
Table2      T21  10.1 8    789
;
run;
```

REPORT_TEMPLATE.XML

```
<?xml version="1.0"?>
<?mso-application progid="Excel.Sheet"?>
<Workbook xmlns="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:o="urn:schemas-microsoft-com:office:office"
  xmlns:x="urn:schemas-microsoft-com:office:excel"
  xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882"
  xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:html="http://www.w3.org/TR/REC-html40">
<DocumentProperties xmlns="urn:schemas-microsoft-com:office:office">
  <Author>malba</Author>
  <LastAuthor>Sarita Kannarath</LastAuthor>
  <LastPrinted>2013-07-25T13:30:58Z</LastPrinted>
  <Created>2013-06-25T14:25:32Z</Created>
  <LastSaved>2014-05-06T09:59:28Z</LastSaved>
  <Company>Microsoft</Company>
  <Version>15.00</Version>
</DocumentProperties>
<CustomDocumentProperties xmlns="urn:schemas-microsoft-com:office:office">
  <ContentTypeId dt:dt="string">0x010100B3F85B1C26C3834A923B0AE990DC5E8F</ContentTypeId>
</CustomDocumentProperties>
<OfficeDocumentSettings xmlns="urn:schemas-microsoft-com:office:office">
  <AllowPNG/>
</OfficeDocumentSettings>
<ExcelWorkbook xmlns="urn:schemas-microsoft-com:office:excel">
  <WindowHeight>11595</WindowHeight>
  <WindowWidth>19200</WindowWidth>
  <WindowTopX>0</WindowTopX>
  <WindowTopY>0</WindowTopY>
  <ActiveSheet>1</ActiveSheet>
  <ProtectStructure>False</ProtectStructure>
  <ProtectWindows>False</ProtectWindows>
</ExcelWorkbook>
<styles>
<Style ss:ID="Default" ss:Name="Normal">
  <Alignment ss:Vertical="Bottom"/>
  <Borders/>
  <Font ss:FontName="Calibri" x:Family="Swiss" ss:Size="11" ss:Color="#000000"/>
  <Interior/>
  <NumberFormat/>
  <Protection/>
</Style>
<Style ss:ID="s62" ss:Name="Normal_Assets_Final">
```

```

<Alignment ss:Vertical="Bottom"/>
<Borders/>
<Font ss:FontName="Calibri" x:Family="Swiss" ss:Size="11" ss:Color="#000000"/>
<Interior/>
<NumberFormat/>
<Protection/>
</Style>
<Style ss:ID="m114671408">
<Alignment ss:Horizontal="Left" ss:Vertical="Center" ss:Indent="2"
  ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="2"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="14" ss:Bold="1"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="m114671448">
<Alignment ss:Horizontal="Center" ss:Vertical="Center" ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="2"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11" ss:Bold="1"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="m114671824">
<Alignment ss:Horizontal="Left" ss:Vertical="Center" ss:Indent="2"
  ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="2"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="14" ss:Bold="1"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="m114671864">
<Alignment ss:Horizontal="Center" ss:Vertical="Center" ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="2"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11" ss:Bold="1"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s63">
<Alignment ss:Vertical="Center"/>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
</Style>
<Style ss:ID="s64">
<Alignment ss:Horizontal="Center" ss:Vertical="Center"/>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
</Style>
<Style ss:ID="s72">
<Alignment ss:Vertical="Center"/>
<Borders/>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
<Interior ss:Color="FFFFFF" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s73">
<Alignment ss:Horizontal="Center" ss:Vertical="Center" ss:WrapText="1"/>

```



```

<Borders/>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11" ss:Bold="1"/>
<Interior ss:Color="#FFFFFF" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s75">
<Alignment ss:Horizontal="Center" ss:Vertical="Center" ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="2"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11" ss:Bold="1"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s85">
<Alignment ss:Vertical="Center"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="2"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s86">
<Alignment ss:Vertical="Center"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s87">
<Alignment ss:Horizontal="Center" ss:Vertical="Center" ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11" ss:Bold="1"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s88">
<Alignment ss:Horizontal="Center" ss:Vertical="Center" ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="2"/>
  <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11" ss:Bold="1"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
</Style>
<Style ss:ID="s90" ss:Parent="s62">
<Alignment ss:Horizontal="Center" ss:Vertical="Center"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
</Borders>
<Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="9"/>
<Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
<NumberFormat ss:Format="@"/>
</Style>
<Style ss:ID="s91">
<Alignment ss:Horizontal="Center" ss:Vertical="Center" ss:WrapText="1"/>
<Borders>
  <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
  <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>

```

```

    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="2"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="9"/>
  <Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
  <NumberFormat ss:Format="@"/>
</Style>
<Style ss:ID="s92">
  <Alignment ss:Horizontal="Left" ss:Vertical="Center"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="2"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="9"/>
  <Interior ss:Color="#D9D9D9" ss:Pattern="Solid"/>
  <NumberFormat ss:Format="@"/>
</Style>
<Style ss:ID="s93">
  <Alignment ss:Horizontal="Left" ss:Vertical="Center"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="8" ss:Bold="1"/>
  <Interior/>
  <NumberFormat ss:Format="@"/>
</Style>
<Style ss:ID="s94">
  <Alignment ss:Horizontal="Left" ss:Vertical="Center" ss:Indent="1"
    ss:WrapText="1"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
  <Interior/>
</Style>
<Style ss:ID="s95">
  <Alignment ss:Vertical="Center" ss:WrapText="1"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
  <Interior/>
</Style>
<Style ss:ID="s96">
  <Alignment ss:Vertical="Center" ss:WrapText="1"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
  <Interior/>
</Style>
<Style ss:ID="s97">
  <Alignment ss:Vertical="Center" ss:WrapText="1"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="2"/>

```

```

    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
  <Interior/>
</Style>
<Style ss:ID="s98">
  <Alignment ss:Horizontal="Left" ss:Vertical="Center"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="8" ss:Bold="1"/>
  <Interior/>
</Style>
<Style ss:ID="s99">
  <Alignment ss:Vertical="Center" ss:WrapText="1"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
  <Interior/>
</Style>
<Style ss:ID="s100">
  <Alignment ss:Vertical="Center" ss:WrapText="1"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
  <Interior/>
</Style>
<Style ss:ID="s101">
  <Alignment ss:Vertical="Center" ss:WrapText="1"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="2"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="11"/>
  <Interior/>
</Style>
<Style ss:ID="s102">
  <Alignment ss:Horizontal="Left" ss:Vertical="Center"/>
  <Borders>
    <Border ss:Position="Bottom" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Left" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Right" ss:LineStyle="Continuous" ss:Weight="1"/>
    <Border ss:Position="Top" ss:LineStyle="Continuous" ss:Weight="1"/>
  </Borders>
  <Font ss:FontName="Verdana" x:Family="Swiss" ss:Size="8"/>
  <Interior/>
</Style>
</Styles>
<Worksheet ss:Name="Table1">
  <Table ss:ExpandedColumnCount="9" ss:ExpandedRowCount="12" x:FullColumns="1"
  x:FullRows="1" ss:StyleID="s63" ss:DefaultColumnWidth="60"
  ss:DefaultRowHeight="14.25">
    <Column ss:StyleID="s63" ss:AutoFitWidth="0" ss:Width="9.75"/>
    <Column ss:StyleID="s64" ss:AutoFitWidth="0" ss:Width="50.25"/>
    <Column ss:StyleID="s64" ss:AutoFitWidth="0" ss:Width="63"/>
    <Column ss:StyleID="s63" ss:AutoFitWidth="0" ss:Width="96"/>
  </Table>

```

```

<Row ss:AutoFitHeight="0" ss:Height="9.75"/>
<Row ss:AutoFitHeight="0" ss:Height="26.25">
  <Cell ss:Index="2" ss:MergeAcross="7" ss:StyleID="m114671408"><Data
    ss:Type="String">Sample report - Table 1</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="9.75" ss:StyleID="s72">
  <Cell ss:Index="2" ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="18.75">
  <Cell ss:Index="2" ss:MergeAcross="2" ss:StyleID="s75"/>
  <Cell ss:MergeAcross="4" ss:StyleID="m114671448"><Data ss:Type="String">Metrics 1</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="63">
  <Cell ss:Index="2" ss:StyleID="s85"/>
  <Cell ss:StyleID="s86"/>
  <Cell ss:StyleID="s86"/>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 1</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 2</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 3</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 4</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 5</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s88"><Data ss:Type="String">Row</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">ID</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Item</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">010</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">020</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">030</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">040</Data></Cell>
  <Cell ss:StyleID="s91"><Data ss:Type="String">050</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">010</Data></Cell>
  <Cell ss:StyleID="s93"><Data ss:Type="String">1</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 1</Data></Cell>
  <Cell ss:StyleID="s95"/>
  <Cell ss:StyleID="s96"/>
  <Cell ss:StyleID="s96"/>
  <Cell ss:StyleID="s96"/>
  <Cell ss:StyleID="s97"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">020</Data></Cell>
  <Cell ss:StyleID="s98"><Data ss:Type="Number">1.100000000000001</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 2</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">030</Data></Cell>
  <Cell ss:StyleID="s102"><Data ss:Type="String">1.1.1</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 3</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>

```

```

<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">040</Data></Cell>
  <Cell ss:StyleID="s102"><Data ss:Type="String">1.1.2</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 4</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">050</Data></Cell>
  <Cell ss:StyleID="s102"><Data ss:Type="String">1.1.3</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 5</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">060</Data></Cell>
  <Cell ss:StyleID="s98"><Data ss:Type="Number">1.2</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 6</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
</Table>
<WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
  <PageSetup>
    <Layout x:Orientation="Landscape"/>
    <Header x:Margin="0.31496062992125984"/>
    <Footer x:Margin="0.31496062992125984"/>
    <PageMargins x:Bottom="0.74803149606299213" x:Left="0.70866141732283472"
      x:Right="0.70866141732283472" x:Top="0.74803149606299213"/>
  </PageSetup>
  <Unsynced/>
  <FitToPage/>
  <Print>
    <ValidPrinterInfo/>
    <PaperSizeIndex>9</PaperSizeIndex>
    <Scale>46</Scale>
    <HorizontalResolution>600</HorizontalResolution>
    <VerticalResolution>600</VerticalResolution>
  </Print>
  <Zoom>85</Zoom>
  <DoNotDisplayGridlines/>
  <Panes>
    <Pane>
      <Number>3</Number>
      <ActiveRow>7</ActiveRow>
      <ActiveCol>3</ActiveCol>
    </Pane>
  </Panes>
  <ProtectObjects>False</ProtectObjects>
  <ProtectScenarios>False</ProtectScenarios>
</WorksheetOptions>
</Worksheet>
<Worksheet ss:Name="Table2">
  <Table ss:ExpandedColumnCount="9" ss:ExpandedRowCount="12" x:FullColumns="1"
    x:FullRows="1" ss:StyleID="s63" ss:DefaultColumnWidth="60"
    ss:DefaultRowHeight="14.25">
    <Column ss:StyleID="s63" ss:AutoFitWidth="0" ss:Width="9.75"/>
    <Column ss:StyleID="s64" ss:AutoFitWidth="0" ss:Width="50.25"/>
    <Column ss:StyleID="s64" ss:AutoFitWidth="0" ss:Width="63"/>
    <Column ss:StyleID="s63" ss:AutoFitWidth="0" ss:Width="101.25"/>
  </Table>

```

```

<Row ss:AutoFitHeight="0" ss:Height="9.75"/>
<Row ss:AutoFitHeight="0" ss:Height="26.25">
  <Cell ss:Index="2" ss:MergeAcross="7" ss:StyleID="m114671824"><Data
    ss:Type="String">Sample report - Table 2</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="9.75" ss:StyleID="s72">
  <Cell ss:Index="2" ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
  <Cell ss:StyleID="s73"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="18.75">
  <Cell ss:Index="2" ss:MergeAcross="2" ss:StyleID="s75"/>
  <Cell ss:MergeAcross="4" ss:StyleID="m114671864"><Data ss:Type="String">Metrics 2</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="63">
  <Cell ss:Index="2" ss:StyleID="s85"/>
  <Cell ss:StyleID="s86"/>
  <Cell ss:StyleID="s86"/>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 1</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 2</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 3</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 4</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Column header 5</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s88"><Data ss:Type="String">Row</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">ID</Data></Cell>
  <Cell ss:StyleID="s87"><Data ss:Type="String">Item</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">010</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">020</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">030</Data></Cell>
  <Cell ss:StyleID="s90"><Data ss:Type="String">040</Data></Cell>
  <Cell ss:StyleID="s91"><Data ss:Type="String">050</Data></Cell>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">010</Data></Cell>
  <Cell ss:StyleID="s93"><Data ss:Type="String">1</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 1</Data></Cell>
  <Cell ss:StyleID="s95"/>
  <Cell ss:StyleID="s96"/>
  <Cell ss:StyleID="s96"/>
  <Cell ss:StyleID="s96"/>
  <Cell ss:StyleID="s97"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">020</Data></Cell>
  <Cell ss:StyleID="s98"><Data ss:Type="Number">1.100000000000001</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 2</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">030</Data></Cell>
  <Cell ss:StyleID="s102"><Data ss:Type="String">1.1.1</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 3</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>

```

```

<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">040</Data></Cell>
  <Cell ss:StyleID="s102"><Data ss:Type="String">1.1.2</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 4</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">050</Data></Cell>
  <Cell ss:StyleID="s102"><Data ss:Type="String">1.1.3*</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 5</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
<Row ss:AutoFitHeight="0" ss:Height="24.9375">
  <Cell ss:Index="2" ss:StyleID="s92"><Data ss:Type="String">060</Data></Cell>
  <Cell ss:StyleID="s98"><Data ss:Type="Number">1.2</Data></Cell>
  <Cell ss:StyleID="s94"><Data ss:Type="String">Description 6</Data></Cell>
  <Cell ss:StyleID="s99"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s100"/>
  <Cell ss:StyleID="s101"/>
</Row>
</Table>
<WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
  <PageSetup>
    <Layout x:Orientation="Landscape"/>
    <Header x:Margin="0.31496062992125984"/>
    <Footer x:Margin="0.31496062992125984"/>
    <PageMargins x:Bottom="0.74803149606299213" x:Left="0.70866141732283472"
      x:Right="0.70866141732283472" x:Top="0.74803149606299213"/>
  </PageSetup>
  <Unsynced/>
  <FitToPage/>
  <Print>
    <ValidPrinterInfo/>
    <PaperSizeIndex>9</PaperSizeIndex>
    <Scale>46</Scale>
    <HorizontalResolution>600</HorizontalResolution>
    <VerticalResolution>600</VerticalResolution>
  </Print>
  <Zoom>85</Zoom>
  <Selected/>
  <DoNotDisplayGridlines/>
  <Panes>
    <Pane>
      <Number>3</Number>
      <ActiveRow>16</ActiveRow>
      <ActiveCol>5</ActiveCol>
    </Pane>
  </Panes>
  <ProtectObjects>False</ProtectObjects>
  <ProtectScenarios>False</ProtectScenarios>
</WorksheetOptions>
</Worksheet>
</Workbook>

```