

# Comparing the SAS XML Libname Engine with ODS MARKUP in the Creation of XML Files

Kirk Paul Lafler, Software Intelligence Corporation

## Abstract

XML is quickly becoming the industry standard for the transmission and manipulation of data over the Web. With the availability of powerful tools and capabilities in the SAS System, SAS users are able to easily and quickly create XML files from SAS data sets and procedure output. This paper illustrates the SAS XML Libname Engine against the ODS MARKUP statement to create default XML output.

## Introduction

This paper compares the SAS XML Libname Engine with the ODS MARKUP destination in creating XML files. Users will learn how to use the XML Engine to format a SAS data set in order to move it across host environments. The ODS MARKUP destination will then be illustrated as an alternative approach in creating XML output.

## Table Used in Examples

The data used in each example consists of a selection of movies that I've viewed over the years. The MOVIES table consists of six columns: title, length, category, year, studio, and rating. Title, category, studio, and rating are defined as character columns with length and year being defined as numeric columns. The example data is illustrated below.

## MOVIES Table

	Title	Length	Category	Year	Studio	Rating
1	Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
2	Casablanca	103	Drama	1942	MGM / UA	PG
3	Christmas Vacation	97	Comedy	1989	Warner Brothers	PG-13
4	Coming to America	116	Comedy	1988	Paramount Pictures	R
5	Dracula	130	Horror	1993	Columbia TriStar	R
6	Dressed to Kill	105	Drama Mysteries	1980	Filmways Pictures	R
7	Forrest Gump	142	Drama	1994	Paramount Pictures	PG-13
8	Ghost	127	Drama Romance	1990	Paramount Pictures	PG-13
9	Jaws	125	Action Adventure	1975	Universal Studios	PG
10	Jurassic Park	127	Action	1993	Universal Pictures	PG-13
11	Lethal Weapon	110	Action Cops & Robber	1987	Warner Brothers	R
12	Michael	106	Drama	1997	Warner Brothers	PG-13
13	National Lampoon's Vacation	98	Comedy	1983	Warner Brothers	PG-13
14	Poltergeist	115	Horror	1982	MGM / UA	PG
15	Rocky	120	Action Adventure	1976	MGM / UA	PG
16	Scarface	170	Action Cops & Robber	1983	Universal Studios	R
17	Silence of the Lambs	118	Drama Suspense	1991	Orion	R
18	Star Wars	124	Action Sci-Fi	1977	Lucas Film Ltd	PG
19	The Hunt for Red October	135	Action Adventure	1989	Paramount Pictures	PG
20	The Terminator	108	Action Sci-Fi	1984	Live Entertainment	R
21	The Wizard of Oz	101	Adventure	1939	MGM / UA	G
22	Titanic	194	Drama Romance	1997	Paramount Pictures	PG-13

## The SAS XML Libname Engine

The SAS XML engine exports and imports XML documents in a sequential manner, starting at the beginning of a file and processing one record after another until reaching the end of the file. By using a Libname statement to associate a libref with an XML document, users are able to process XML documents. In the example illustrated below, an XML document is created from the SAS data set MOVIES by specifying the XML engine in the libname statement. The simple DATA step routine reads the data set WORK.MOVIES and produces the XML document using XML markup. A partial listing of the resulting XML document appears below.

## SAS Code

```
LIBNAME XMLCNVT XML 'C:\XML-DOCUMENT.XML';

DATA XMLCNVT.MOVIES;
  SET WORK.MOVIES;
RUN;
```

## XML Engine Output

```
<?xml version="1.0" encoding="windows-1252" ?>
= <TABLE>
= <MOVIES>
  <Title>Brave Heart</Title>
  <Length>177</Length>
  <Category>Action Adventure</Category>
  <Year>1995</Year>
  <Studio>Paramount Pictures</Studio>
  <Rating>R</Rating>
  </MOVIES>
= <MOVIES>
  <Title>Casablanca</Title>
  <Length>103</Length>
  <Category>Drama</Category>
  <Year>1942</Year>
  <Studio>MGM / UA</Studio>
  <Rating>PG</Rating>
  </MOVIES>
= <MOVIES>
  <Title>Christmas Vacation</Title>
  <Length>97</Length>
  <Category>Comedy</Category>
  <Year>1989</Year>
  <Studio>Warner Brothers</Studio>
  <Rating>PG-13</Rating>
  </MOVIES>
... .. Part of Listing Omitted ... ..
= <MOVIES>
  <Title>The Terminator</Title>
  <Length>108</Length>
  <Category>Action Sci-Fi</Category>
  <Year>1984</Year>
  <Studio>Live Entertainment</Studio>
  <Rating>R</Rating>
  </MOVIES>
= <MOVIES>
  <Title>The Wizard of Oz</Title>
  <Length>101</Length>
  <Category>Adventure</Category>
  <Year>1939</Year>
  <Studio>MGM / UA</Studio>
  <Rating>G</Rating>
  </MOVIES>
= <MOVIES>
  <Title>Titanic</Title>
  <Length>194</Length>
  <Category>Drama Romance</Category>
  <Year>1997</Year>
  <Studio>Paramount Pictures</Studio>
  <Rating>PG-13</Rating>
  </MOVIES>
</TABLE>
```

## XML Engine Analysis

As you review the above XML document's contents, you will notice a series of markup tags. Each markup tag is enclosed by a "<" and ">" symbols. Each tag has a beginning tag and a matching ending tag, as illustrated with <Title> and </Title>, <Length> and </Length>, <Category> and </Category>, and so on. XML's main objective is to separate the data from the presentation. The XML document illustrated above displays just the actual data without any presentation or formatting attributes.

## ODS MARKUP

The ODS MARKUP statement can also be used to export XML documents. Users already familiar with Output Delivery System (ODS) will find the required syntax requirements to create an XML document consistent with other ODS destinations. In the next example, an XML document is created from the SAS data set MOVIES by specifying the ODS MARKUP statement along with the BODY= parameter to specify the location and physical name for the XML document being created and the "default" tagset parameter. The simple PROC PRINT routine routes output from the input data set WORK.MOVIES to the XML document using the external file referenced by the BODY= parameter. A partial listing of the resulting XML document appears below.

### SAS Code

```
ODS LISTING CLOSE;
ODS MARKUP BODY='C:\XML-DOCUMENT.XML' TAGSET=DEFAULT;

PROC PRINT DATA=WORK.MOVIES;
  TITLE;
RUN;

ODS MARKUP CLOSE;
ODS LISTING;
```

### ODS MARKUP Output

```
<?xml version="1.0" encoding="windows-1252" ?>
= <odsxml>
= <head>
= <meta operator="Software Intelligence Corporation" />
  </head>
= <body>
= <proc name="Print">
= <label name="IDX" />
= <branch name="Print" label="The Print Procedure" class="ContentProcName" toc-level="1">
= <leaf name="Print" label="Data Set WORK.MOVIES" class="ContentItem" toc-level="2">
= <output name="Print" label="Data Set WORK.MOVIES" clabel="Data Set WORK.MOVIES">
= <output-object type="table" class="Table">
= <style>
  <border spacing="1" padding="7" rules="groups" frame="box" />
  </style>
= <colspecs columns="7">
= <colgroup>
  <colspec name="1" width="2" align="right" type="int" />
  </colgroup>
= <colgroup>
  <colspec name="2" width="27" type="string" />
  <colspec name="3" width="3" align="decimal" type="double" />
  <colspec name="4" width="20" type="string" />
  <colspec name="5" width="4" align="decimal" type="double" />
  <colspec name="6" width="18" type="string" />
  <colspec name="7" width="5" type="string" />
  </colgroup>
  </colspecs>
= <output-head>
= <row>
= <header type="string" class="Header" row="1" column="1">
  <value>Obs</value>
  </header>
= <header type="string" class="Header" row="1" column="2">
  <value>Title</value>
  </header>
= <header type="string" class="Header" row="1" column="3">
  <value>Length</value>
  </header>
= <header type="string" class="Header" row="1" column="4">
  <value>Category</value>
  </header>
= <header type="string" class="Header" row="1" column="5">
```

**ODS MARKUP Output (continued)**

```
<value>Year</value>
</header>
= <header type="string" class="Header" row="1" column="6">
<value>Studio</value>
</header>
= <header type="string" class="Header" row="1" column="7">
<value>Rating</value>
</header>
</row>
</output-head>
= <output-body>
= <row>
= <header type="double" class="RowHeader" row="2" column="1">
<value>1</value>
</header>
= <data type="string" class="Data" row="2" column="2">
<value>Brave Heart</value>
</data>
= <data raw-value="QGYgAAAAAAA=" type="double" class="Data" row="2" column="3">
<value>177</value>
</data>
= <data type="string" class="Data" row="2" column="4">
<value>Action Adventure</value>
</data>
= <data raw-value="QJ8sAAAAAAA=" type="double" class="Data" row="2" column="5">
<value>1995</value>
</data>
= <data type="string" class="Data" row="2" column="6">
<value>Paramount Pictures</value>
</data>
= <data type="string" class="Data" row="2" column="7">
<value>R</value>
</data>
</row>
= <row>
= <header type="double" class="RowHeader" row="3" column="1">
<value>2</value>
</header>
= <data type="string" class="Data" row="3" column="2">
<value>Casablanca</value>
</data>
= <data raw-value="QFnAAAAAAA=" type="double" class="Data" row="3" column="3">
<value>103</value>
</data>
= <data type="string" class="Data" row="3" column="4">
<value>Drama</value>
</data>
= <data raw-value="QJ5YAAAAAAA=" type="double" class="Data" row="3" column="5">
<value>1942</value>
</data>
= <data type="string" class="Data" row="3" column="6">
<value>MGM / UA</value>
</data>
= <data type="string" class="Data" row="3" column="7">
<value>PG</value>
</data>
</row>
```

... .. Part of Listing Omitted ... ..

```
= <row>
= <header type="double" class="RowHeader" row="22" column="1">
<value>21</value>
</header>
= <data type="string" class="Data" row="22" column="2">
```

## ODS MARKUP Output (continued)

```
<value>The Wizard of Oz</value>
</data>
= <data raw-value="QFIAAAAAAAAA=" type="double" class="Data" row="22" column="3">
<value>101</value>
</data>
= <data type="string" class="Data" row="22" column="4">
<value>Adventure</value>
</data>
= <data raw-value="QJ5MAAAAAAAAA=" type="double" class="Data" row="22" column="5">
<value>1939</value>
</data>
= <data type="string" class="Data" row="22" column="6">
<value>MGM / UA</value>
</data>
= <data type="string" class="Data" row="22" column="7">
<value>G</value>
</data>
</row>
= <row>
= <header type="double" class="RowHeader" row="23" column="1">
<value>22</value>
</header>
= <data type="string" class="Data" row="23" column="2">
<value>Titanic</value>
</data>
= <data raw-value="QGhAAAAAAAAA=" type="double" class="Data" row="23" column="3">
<value>194</value>
</data>
= <data type="string" class="Data" row="23" column="4">
<value>Drama Romance</value>
</data>
= <data raw-value="QJ80AAAAAAAA=" type="double" class="Data" row="23" column="5">
<value>1997</value>
</data>
= <data type="string" class="Data" row="23" column="6">
<value>Paramount Pictures</value>
</data>
= <data type="string" class="Data" row="23" column="7">
<value>PG-13</value>
</data>
</row>
</output-body>
</output-object>
</output>
</leaf>
</branch>
</proc>
</body>
</odsxml>
```

## Conclusion

The XML Engine can be used by SAS users to process XML documents. XML documents can be created from a SAS data set by translating the proprietary format into XML markup. Once translated, the XML document can then be migrated to another host so the XML markup can be translated back into a SAS data set, as well as be used by products that process XML documents. XML documents can also be created using the ODS MARKUP statement. Users already familiar with ODS may find this method to be a useful extension to their skill set.

## References

SAS XML LIBNAME Engine: User's Guide, Getting Started with the XML Engine; and SAS Output Delivery System: User's Guide; SAS Institute Inc., Cary, NC, USA.

## Acknowledgments

I would like to thank Beth Newsom, WUSS Coders' Corner Section Chair, for accepting my abstract and paper, as well as the WUSS Leadership for their support of a great Conference.

## Author Bio

Kirk Paul Lafler, a SAS Certified Professional® and SAS Alliance Partner® (1996 - 2002) with more than 27 years of SAS software experience, provides consulting services and hands-on SAS training around the world. Kirk has written four books including PROC SQL: Beyond the Basics Using SAS by SAS Institute, Power SAS and Power AOL by Apress, and more than one hundred articles in professional journals and SAS User Group proceedings. His popular SAS Tips column, Kirk's Korner, appears regularly in the BASAS, HASUG, SANDS, SAS, SESUG, and WUSS Newsletters and websites. Kirk can be reached at:

Kirk Paul Lafler  
Software Intelligence Corporation  
P.O. Box 1390  
Spring Valley, California 91979-1390  
E-mail: KirkLafler@cs.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.