

Paper 118-2007

Using the SAS[®] Output Delivery System and PROC TEMPLATE to Create XHTML Files

Frederick E. Pratter, Eastern Oregon University, La Grande OR

ABSTRACT

SAS 8.2 introduced the ODS MARKUP statement, allowing users to export to a variety of markup languages, including HTML, XML and XHTML. The ODS MARKUP statement uses essentially the same syntax as the deprecated ODS HTML statement, except for the addition of TAGSET= option. The value of this option determines the type of output file to be created. In addition to the list of predefined tagsets available from SAS, it is also possible to create new ones, as well as customizing the SAS-supplied tagsets. The TEMPLATE procedure is used to review, create and customize tagsets.

This workshop will cover using an existing template to create XHTML from SAS output. In addition, we will explore modifying the existing template to change the look of the resulting output.

USING ODS STYLES

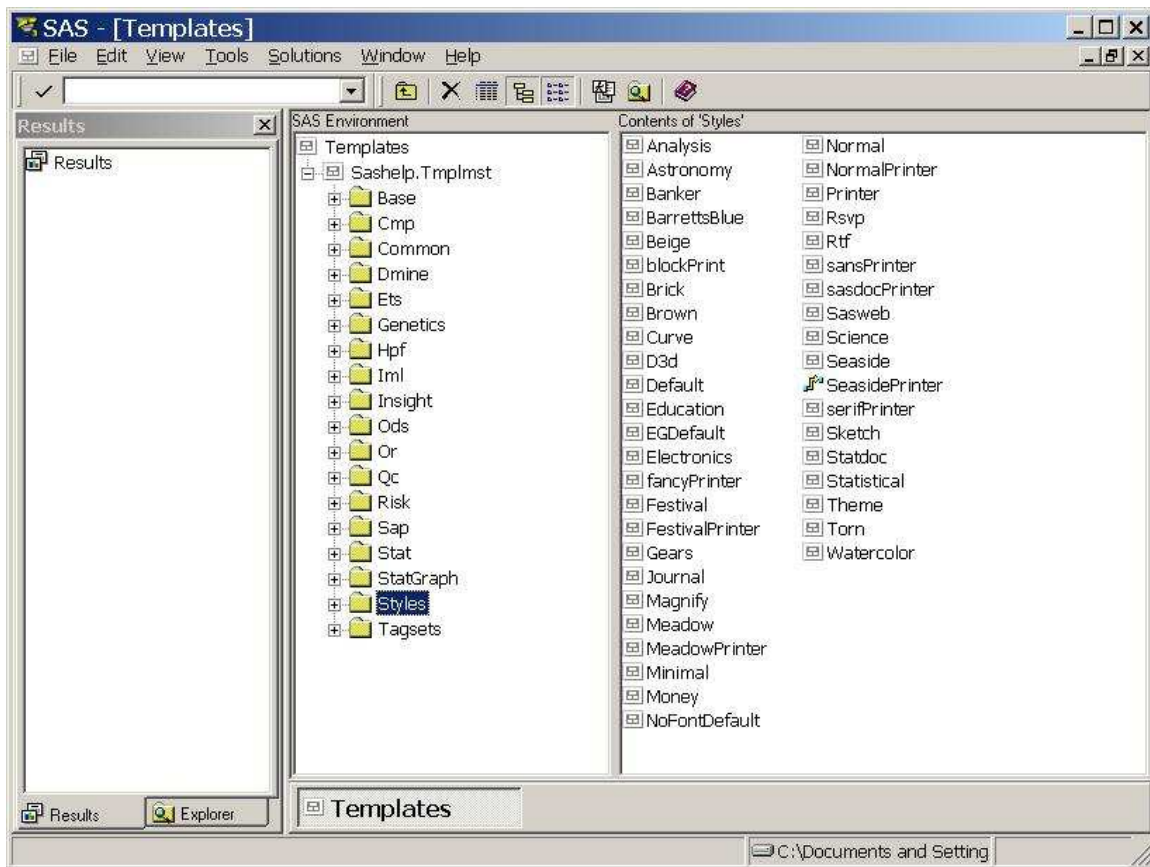
One initial ODS option needs to be discussed before the topic of custom tagsets can be introduced. Unfortunately, it is a huge one, and in the time available we can only nibble around the edges. This is the STYLE= option, used to specify a style template (also known as a style definition) for HTML.

The SAS installation procedures in Version 8 and later automatically create a library file called an *item store*. This file stores the standard ODS templates; default, it is called SASHELP.TMPLMST. The item store actually holds several different kinds of templates, the most important of which are *table* and *style*. The table templates are used by the SAS System to provide basic table layouts. The style templates, on the other hand, describe the presentation of the information, such as color, font, and alignment.

You can also create your own templates; by default these are stored in SASUSER.TEMPLAT. Be careful though; if SAS is reinstalled the SASUSER library is reinitialized and all your templates will go away. Keep a backup, just in case. (Note that you cannot move SASHELP.TMPLMST or SASUSER.TEMPLAT across platforms. You should keep the SAS code that created the template, and not just make a backup copy of the item store.)

If you are interested in the details of this process, select the *View>Results* from the main window in SAS Display Manager. From the *Results* window, again select *View*. The menu should have changed so that the first item is *Templates*. Click on *Templates* then expand SASHELP.TMPLMST by clicking on the + sign. . (You can also get this information by typing "odstemplate" in the task bar or command line.)

You should see something like the window shown in **Display 1**:



Display 1. SAS Template Library

Clicking on one of the styles brings up the SAS code for the corresponding TEMPLATE procedure. For example, the *Default* style starts out as shown in **Display 2** (and goes on for some ways):

```
proc template;
  define style Styles.Default;
    style fonts
      "Fonts used in the default style" /
      'TitleFont2' = ("Arial, Helvetica, sans-serif",4,Bold Italic)
      'TitleFont' = ("Arial, Helvetica, sans-serif",5,Bold Italic)
      'StrongFont' = ("Arial, Helvetica, sans-serif",4,Bold)
      'EmphasisFont' = ("Arial, Helvetica, sans-serif",3,Italic)
      'FixedEmphasisFont' = ("Courier New, Courier, monospace",2,Italic)
      'FixedStrongFont' = ("Courier New, Courier, monospace",2,Bold)
      'FixedHeadingFont' = ("Courier New, Courier, monospace",2)
      'BatchFixedFont' = ("SAS Monospace, Courier New, Courier, monospace",2)
      'FixedFont' = ("Courier",2)
      'headingEmphasisFont' = ("Arial, Helvetica, sans-serif",4,Bold Italic)

    [ ... many more statements omitted ... ]

  end;
```

Display 2. Default Style Template

Once you get into it, you can use the TEMPLATE procedure to modify existing styles and even create your own. To specify a style on the ODS HTML statement, just use the STYLE= option with one of the style templates. For example:

```

filename OUT  "example1.html";
title "Sales Total by Month 1991-1994";

ods listing close;
ods html body=OUT style=sasweb;

data _null_;
  set SASHELP.RETAIL;
  where (YEAR gt 1990);
  file print ods=(variables=(YEAR MONTH SALES));
  put _ods_;
run;

ods html close;
ods listing;

```

Example 1. ODS Sasweb Style Data Set Listing

The output resulting from running this program is shown in **Display 3** below:

Sales Total by Month 1991-1994		
YEAR	MONTH	Retail sales in millions of \$
1991	1	\$703
1991	4	\$709
1991	7	\$728
1991	10	\$807
1992	1	\$692
1992	4	\$797
1992	7	\$826
1992	10	\$889
1993	1	\$758
1993	4	\$909
1993	7	\$920
1993	10	\$991
1994	1	\$876
1994	4	\$998

Display 3. ODS HTML Sasweb Style

There are a great many other formatting options available with the ODS. Some of the most useful include:

- **STYLESHEET** option by itself (no = sign or file specification) inserts an *internal* CSS <STYLE> section into the created HTML page; see the preceding chapter for a discussion of Cascading Style Sheets and image files).

- `STYLESHEET=newcssfilename.css` creates an external CSS File and puts an HTML `<LINK>` tag in the created HTML page that points to the newly created CSS file.
- `STYLESHEET=(URL=existingcssfile.css)` puts an HTML `<LINK>` tag in the created HTML page that points to an already existing css file which should contain the HTML and CLASS selectors expected by ODS, as well as any corporate CLASS or other selectors that are required.

The first choice, the `STYLESHEET` option by itself, was necessary in SAS 8 to generate an inline `<STYLE>` section; in SAS 9 this is the default. In other words, in SAS 9 with ODS HTML you get an inline `<STYLE>` section without having to specify the option.

`PROC TEMPLATE` is available to modify existing styles or create your own from scratch; this procedure can also be used to create custom TABLE templates to be used with SAS procedures instead of the default TABLE templates. ODS can create a GIF file of SAS/Graph output and an HTML file with an `` tag to display the graph. Also, SAS now allows the use of SAS/Graph style format controls in the `TITLE`, `FOOTNOTE`, and `LABEL` statements.

All these topics (and more) await you when you begin to explore the resources SAS has provided for formatting ODS output; see <http://support.sas.com/rnd/base/index-ods-resources.html> for more information on the resources available for the SAS Output Delivery System. There is also a site for customer contributed tagsets (including XHTML) at <http://support.sas.com/rnd/base/topics/odsmarkup/customer.html>.

Using the ODS Markup Statement to Create XHTML

All of the discussion about creating Web pages thus far has applied mostly to HTML 4.01, but HTML is only one kind of markup language. As noted earlier, the existing specification has a major drawback, that is, that the HTML standard includes both data and the instructions for displaying that information. Mixing the data and the presentation in this way is considered a “no-no” in object oriented programming circles, because it makes it difficult to change one and not the other.

The solution (proposed in February 1998) is XML, the *Extensible Markup Language*. XML eliminates any formatting instructions, in favor of separate stylesheets. (See the SUGI papers referenced at the end of this chapter for more detail on how to create XML using SAS). XML immediately became the medium of choice in industry for cross-platform data transfer, but it has not replaced HTML for data presentation. This is probably due to two reasons: first, the large body of existing HTML code, and second, the additional complication of learning and maintaining the new formats.

In response to this singular lack of enthusiasm, in January 2000 the W3C issued a new recommendation for “XHTML™ 1.0: The eXtensible HyperText Markup Language, A Reformulation of HTML 4 in XML 1.0” (<http://www.w3.org/TR/xhtml1/>). This new standard was subsequently updated to XHTML 1.1 in May 2001 as “XHTML 1.1 - Module-based XHTML” (<http://www.w3.org/TR/2001/REC-xhtml11-20010531>). As the recommendation notes, XHTML is intended to replace HTML, which is now *deprecated* by the W3C. While HTML is still supported, it will go away at some point and be replaced by XHTML, which is an XML based Hypertext markup Language.

The folks at SAS have recognized that the application technology necessary to write HTML is pretty much the same as that needed to write XML, XSL, CSV (comma delimited), DTD (XML Document Type Definition), and CSS. The ODS MARKUP statement was introduced in SAS Release 8.2 in order to allow the creation of files in a variety of markup languages. As explained in the article “Using ODS to Export Markup Languages” (<http://support.sas.com/rnd/base/topics/odsmarkup/>), the syntax for ODS MARKUP is basically the same as ODS HTML, except that ODS MARKUP has the `TAGSET=` option. In future release of the SAS System, it is to be expected that ODS HTML will be deprecated and ODS MARKUP will be the method of choice.

The `TAGSET=` option is used to specify the type of output required. Currently in Release 8.2, over 20 different destinations are supported. The template window (see **Display 1**) can also be used to list the available tagsets.

As the **Example 2** below illustrates, if you know how to use the ODS HTML statement, you can use ODS MARKUP to create XHTML.

```
filename OUT "example2.html";

ods listing close;
ods markup body=OUT tagset=xhtml style=sasweb;
```

```

proc tabulate data=SASHELP.RETAIL;
  title "Retail Sales In Millions Of $";
  class YEAR/descending;
  var SALES;
  table YEAR="" all="Total", SALES="" *
    (sum="Total Sales"*f=dollar8.
     pctsum="Overall Percent"*f=8.2
     n="Number of Sales"*f=8.
     mean="Average Sale"*f=dollar8.2
     min="Smallest Sale"*f=dollar8.
     max="Largest Sale"*f=dollar8.)/
    box="Year" rts=8;

run;

ods markup close;
ods listing;

```

Example 2. ODS Markup: XHTML

Using the default XHTML template from the SAS 9.1 library it is possible to create the XHTML Web page shown in **Display 3**.

Retail Sales In Millions Of \$						
Year	Total Sales	Overall Percent	Number of Sales	Average Sale	Smallest Sale	Largest Sale
1994	\$1,874	5.97	2	\$937.00	\$876	\$998
1993	\$3,578	11.40	4	\$894.50	\$758	\$991
1992	\$3,204	10.21	4	\$801.00	\$692	\$889
1991	\$2,947	9.39	4	\$736.75	\$703	\$807
1990	\$2,734	8.71	4	\$683.50	\$606	\$749
1989	\$2,592	8.26	4	\$648.00	\$594	\$670
1988	\$2,412	7.69	4	\$603.00	\$546	\$643
1987	\$2,164	6.90	4	\$541.00	\$484	\$595
1986	\$1,922	6.13	4	\$480.50	\$419	\$541
1985	\$1,596	5.09	4	\$399.00	\$337	\$448
1984	\$1,528	4.87	4	\$382.00	\$342	\$413
1983	\$1,393	4.44	4	\$348.25	\$299	\$384
1982	\$1,252	3.99	4	\$313.00	\$284	\$343
1981	\$1,148	3.66	4	\$287.00	\$247	\$323
1980	\$1,030	3.28	4	\$257.50	\$220	\$295
Total	\$31,374	100.00	58	\$540.93	\$220	\$998

Display 3. ODS Markup: Default XHTML Output

PROC TEMPLATE: Not Just for Geeks Anymore

The following section explains how to create XHTML (or any other markup) using a custom tagset and ODS. As noted above, the *item store* contains templates for tables and styles. It also contains the tagsets used for the ODS markup statement. As noted above, SAS supplies a simple XHTML template, or you can create your own by running the following program:

```
proc template;

define tagset Tagsets.MyXhtml / store = SASUSER.TEMPLAT;

  define event cell_is_empty;
    put %nrstr("&nbsp;");
  end;

  define event doc;
    start:
      put "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" NL;
      put "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.1//EN\" \" \" NL;
      put "\"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd\">" NL;
      put "<html xmlns=\"http://www.w3.org/1999/xhtml\">" NL;
      ndent;
    finish:
      xdent;
      put "</html>" NL;
  end;

  define event doc_head;
    start:
      put "<head>" NL;
      ndent;
    finish:
      xdent;
      put "</head>" NL;
  end;

  define event doc_body;
    start:
      put "<body>" NL;
      put TITLE;
    finish:
      /* add W3C logo to page */
      put '<p>' NL;
      put '<a href="http://validator.w3.org/check/referer">' NL;
      put '</a>' NL;
      put '</p>' NL;
      put '</body>' NL;
  end;

  define event doc_title;
    put "<title>";
    put "SAS Output: " / if !exists(VALUE);
    put VALUE;
    put "</title>" NL;
    /* add user supplied stylesheet, or you can use
       the "stylesheet=" option on the ODS statement */
    put '<link rel="stylesheet" type="text/css" href=" styles.css">' NL;
  end;

  define event proc_title;
    put "<h2>" VALUE "</h2>" CR;
  end;
```

```

define event system_title;
    put "<h1>" VALUE "</h1>" CR;
end;

define event system_footer;
    put "<h1>" VALUE "</h1>" CR;
end;

define event byline;
    put "<h2>" VALUE "</h2>" CR;
end;

define event note;
    put "<h3>" VALUE "</h3>" CR;
end;

define event fatal;
    put "<h3>" VALUE "</h3>" CR;
end;

define event error;
    put "<h3>" VALUE "</h3>" CR;
end;

define event warning;
    put "<h3>" VALUE "</h3>" CR;
end;

define event table;
    start:
        put "<table border="1"
            cellspacing="0"
            cellpadding="4">" NL;
        ndent;
    finish:
        xdent;
        put "</table>" NL;
end;

define event row;
    start:
        put "<tr>" NL;
        ndent;
    finish:
        xdent;
        put "</tr>" NL;
end;

define event table_head;
    start:
        put "<thead>" NL;
        ndent;
    finish:
        xdent;
        put "</thead>" NL;
end;

define event table_body;
    start:
        put "<tbody>" NL;
        ndent;
    finish:
        xdent;
        put "</tbody>" NL;
end;

```

```

define event table_foot;
  start:
    put "<tfoot>" NL;
    ndent;
  finish:
    xdent;
    put "</tfoot>" NL;
end;

define event rowcol;
  putq " rowspan=" ROWSPAN;
  putq " colspan=" COLSPAN;
end;

define event header;
  start:
    put "<th";
    trigger rowcol;
    put ">";
    put VALUE;
  finish:
    put "</th>";
end;

define event data;
  start:
    put "<th" / if cmp( section , "head" );
    put "<td" / if !cmp( section , "head" );
    trigger rowcol;
    put ">";
    put VALUE;
  finish:
    put "</th>" NL / if cmp( section , "head" );
    put "</td>" NL / if !cmp( section , "head" );
end;

mapsub = %nrstr("/&lt;/&gt;/&amp;/&quot;/");
map = %nrstr("<>&""");
split = "<br />";
output_type = "xml";
indent = 3;

end; /* define tagset */

```

Example 3. Create XHTML Template

PROC TEMPLATE listings can be intimidating, and many otherwise brave SAS users have avoided this procedure because of its reputation for complexity. No worries mate! It is just a question of putting together simple elements, as the above example illustrates.

There is a DEFINE statement for each event in the ODS output (shown in **bold**) that describes what to do in that case. Consider the following extract:

```

define event doc_head;
  start:
    put "<head>" NL;
    ndent;
  finish:
    xdent;
    put "</head>" NL;
end;

```


All this says is that when ODS gets to the top of the document, it should write out a header. At the start of the header, put the opening tag "<head>". After writing the initial tag, insert a line break and then indent. At the finish, stop indenting, put the closing tag "</head>" and then put in a line break. See, that's not so hard!

In any case, pasting this code into the SAS editor window and submitting it should result in the following log message:

NOTE: TAGSET 'Tagsets.MyXhtml1' has been saved to: SASUSER.TEMPLAT

You should also see a new tagset called XHTML in SASUSER.TEMPLAT in the template window (see **Display 1**). You can now modify **Example 2**, changing the name of the template to "MyXHTML" to get the result shown below in **Display 4**:

Retail Sales In Millions Of \$						
Year	Total Sales	Overall Percent	Number of Sales	Average Sale	Smallest Sale	Largest Sale
1994	\$1,874	5.97	2	\$937.00	\$876	\$998
1993	\$3,578	11.40	4	\$894.50	\$758	\$991
1992	\$3,204	10.21	4	\$801.00	\$692	\$889
1991	\$2,947	9.39	4	\$736.75	\$703	\$807
1990	\$2,734	8.71	4	\$683.50	\$606	\$749
1989	\$2,592	8.26	4	\$648.00	\$594	\$670
1988	\$2,412	7.69	4	\$603.00	\$546	\$643
1987	\$2,164	6.90	4	\$541.00	\$484	\$595
1986	\$1,922	6.13	4	\$480.50	\$419	\$541
1985	\$1,596	5.09	4	\$399.00	\$337	\$448
1984	\$1,528	4.87	4	\$382.00	\$342	\$413
1983	\$1,393	4.44	4	\$348.25	\$299	\$384
1982	\$1,252	3.99	4	\$313.00	\$284	\$343
1981	\$1,148	3.66	4	\$287.00	\$247	\$323
1980	\$1,030	3.28	4	\$257.50	\$220	\$295
Total	\$31,374	100.00	58	\$540.93	\$220	\$998



Display 4. Custom XHTML

The really interesting part of this example is the XHTML page source code shown in **Display 5**; this represents the resulting markup from the user-supplied template shown in **Example 3**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

  <head>
    <title>SAS Output: </title>
    <link rel="stylesheet" type="text/css"
      href="styles.css">
  </head>

  <body>
    <h1>Retail Sales In Millions Of $</h1>
```

```

<table border="1" cellspacing="0" cellpadding="4">
  <thead>
    <tr>
      <th>Year</th>
      <th>Total Sales</th>
      <th>Overall Percent</th>
      <th>Number of Sales</th>
      <th>Average Sale</th>
      <th>Smallest Sale</th>
      <th>Largest Sale</th></tr>
    </thead>
    <tbody>
      <tr>
        <th>1994</th><td> $1,874</td>
        <td> 5.97</td>
        <td> 2</td>
        <td> $937.00</td>
        <td> $876</td>
        <td> $998</td>
      </tr>

      [ ... many table rows omitted ... ]

      <tr>
        <th>1980</th><td> $1,030</td>
        <td> 3.28</td>
        <td> 4</td>
        <td> $257.50</td>
        <td> $220</td>
        <td> $295</td>
      </tr>
      <tr>
        <th>Total</th><td> $31,374</td>
        <td> 100.00</td>
        <td> 58</td>
        <td> $540.93</td>
        <td> $220</td>
        <td> $998</td>
      </tr>
    </tbody>
  </table>
  <p>
    <a href="http://validator.w3.org/check/referer">
      </a>
    </p>
</body>
</html>

```

Display 5. Generated XHTML Source

CONCLUSION

XHTML is an alternative to HTML for data display, when used with custom stylesheets to create a formatted document in the client's browser. Using ODS, it is possible to customize your output for almost any conceivable application.

REFERENCES

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 registered trademarks or trademarks of their respective companies.

From SAS Press

- Gupta, Sunil K. 2003. *Quick Results with the Output Delivery System (Art Carpenter's SAS Software)*. Cary, NC: SAS Institute Inc.
- Haworth, Lauren E. 2001. *Output Delivery System: The Basics*. Cary, NC: SAS Institute Inc.
- Johnson, Bernadette. *Instant ODS: Style Templates for the SAS Output Delivery System*. SAS Publishing. 2003.
- Pratter, Frederick E. 2006. *Web Development with SAS® by Example, 2nd edition*. Cary, NC: SAS Institute Inc.

Links

- Extensible Markup Language. <http://www.w3.org/TR/REC-xml.html>
- XHTML. <http://www.w3.org/TR/2001/REC-xhtml11-20010531>
- SAS XML FAQ. <http://www.sas.com/rnd/base/topics/templateFAQ/xmlfaq.htm>
- HTML Formatting Tools. <http://www.sas.com/rnd/web/intrnet/format/>
- HTML Output Formatter Syntax Reference - <http://www.sas.com/rnd/web/intrnet/format/out/outsyn.html>
- HTML Tabulate Formatter Syntax Reference. www.sas.com/rnd/web/intrnet/format/tab/tabsyn.html
- HTML Data Set Formatter Syntax Reference - <http://www.sas.com/rnd/web/intrnet/format/ds/dssyn.html>
- Output Delivery System. <http://www.sas.com/rnd/base/index-ods-resources.html>
- Using ODS to Export Output in a Markup Language. <http://www.sas.com/rnd/base/topics/odsmarkup/>
- Creating Customized Tagsets to Use with ODS and XML LIBNAME Engine. <http://www.sas.com/rnd/base/topics/odstagsets/>
- PROC TEMPLATE FAQ - <http://www.sas.com/rnd/base/topics/templateFAQ/Template.html>.
- Using Styles and Templates to Customize SAS ODS Output. <http://whitepapers.techrepublic.com.com/whitepaper.aspx?docid=157122>
- SAS SUGI proceedings: ODS. http://www.lexjansen.com/cgi-bin/sugi.php?x=tods&s=sugi_t

SAS Training

- *The Complete Guide to the SAS Output Delivery System*- <http://support.sas.com/rnd/base/early-access/odsdoc2/sashtml/tw5195/index.htm>
- *Getting Started with the Output Delivery System Topics*. <http://support.sas.com/training/us/crs/odsqs.html>
- *Advanced Output Delivery System Topics*. <http://support.sas.com/training/us/crs/odsadv9.html>
- *Using the Output Delivery System to Create XML Files*. <http://support.sas.com/training/us/crs/odsxml.html>
- *Creating Markup Language Files Using ODS Markup, SAS XML Libname Engine, and the TEMPLATE Procedure*. <http://support.sas.com/training/us/crs/lwxml.html>

CONTACT

Frederick Pratter
Computer Science/Multimedia Studies Program
Eastern Oregon University
One University Boulevard
La Grande OR 97850
fpratter@eou.edu