Paper 148-27

The Joy of Table-driven Information Delivery Systems

Jonathon Hagerman, Export Development Canada, Ottawa, ON

ABSTRACT

This paper will describe how we employed a data driven approach to generate the HTML pages in our information delivery system programmatically.

It was not only possible to create an information delivery system surfacing information to users via browser technology without a single-stored HyperText Mark-up Language (HTML) page; it was, in fact, desirable.

Programmatically creating these pages enabled us to get out of the HTML page maintenance business and also enabled the business users to deal with the most up-to-date information to make their decisions.

This paper will describe to an intermediate to advanced SAS programmer, the discoveries we made about information delivery systems and how we were able to create our information delivery system model using this information. Once this model was in place, with all of the required tables, programs, macros, and so forth, it was possible to create the next information delivery system in one week.

INTRODUCTION

This paper will describe the Information Delivery System that was developed using SAS/IntrNet® software coupled with a table-driven approach and the lessons learned during development.

The following topics will be addressed:

1. Describe the Information Delivery System
2. Discuss dynamically generating
   - Report Selection Menu
   - Selection Criteria Screens
   - Report Production
     - ❑ HTML
     - ❑ Printed version
     - ❑ Export data to a file if appropriate
3. Benefits to the developer
4. Benefits to the business user
5. Reuse of the Information Delivery System Model

1. THE INFORMATION DELIVERY SYSTEM

The project was to build an information delivery system to surface information out of an existing data mart to business users on our corporate intranet. The requirement was to develop an interface to sixteen reports to be produced using SAS software and then to deliver the reports via browser technology to business users on our corporate intranet. See the Technical Information section for information on the software and hardware used.

The following three figures illustrate a typical navigation through the system. **Figure 1** shows the Report Selection Menu with a drop down box of available reports. **Figure 2** shows the Selection Criteria Screen containing sub-setting, grouping and ordering selections presented to customize the report before it is produced. **Figure 3** shows the resulting report delivered in this example to a browser with the option to either print to a chosen printer or export the information to a csv file following the report.

Figure 1 – Report Selection Screen

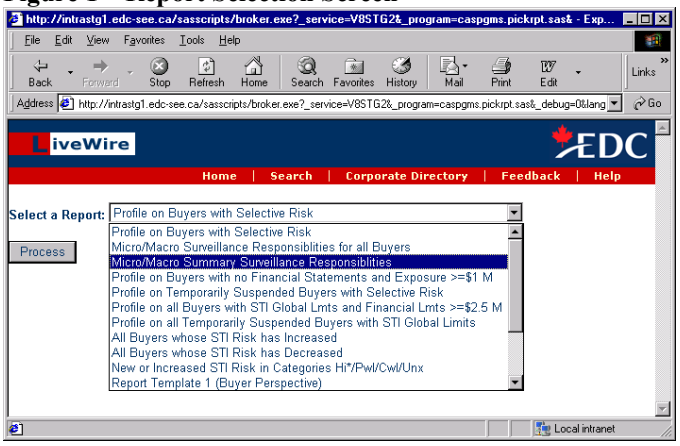


Figure 2 – Selection Criteria Screen

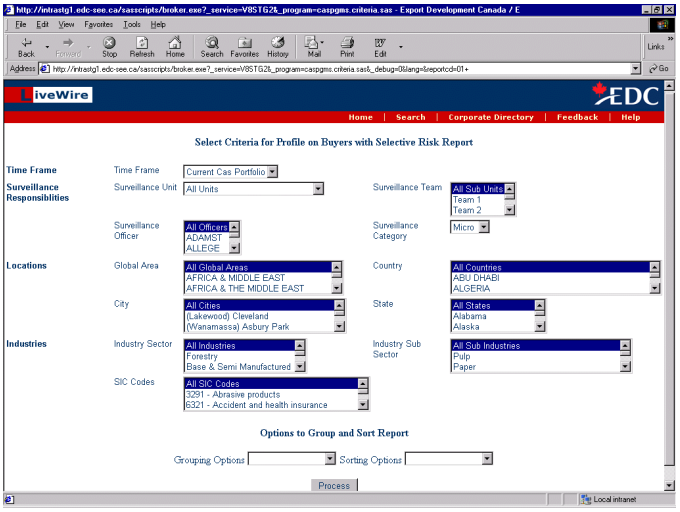
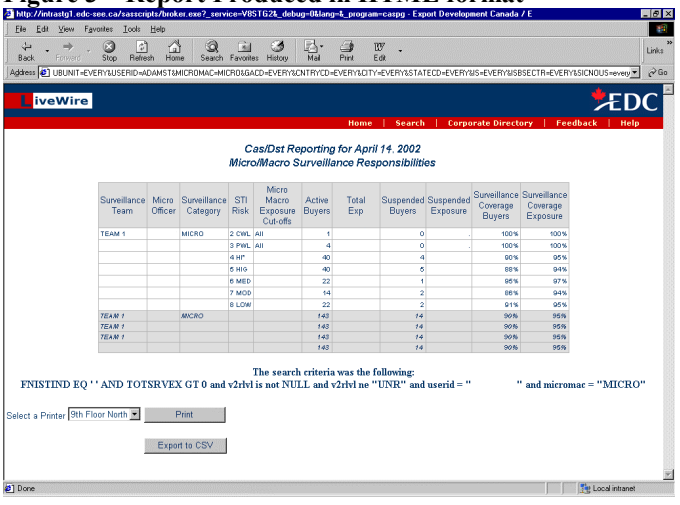


Figure 3 – Report Produced in HTML format



2. DYNAMIC GENERATION

BACKGROUND

One of the most important strategic directions that our group ever took was to minimize the number of HTML pages that were stored on the web server. This decision was based on the fact that we were SAS programmers and that we would rather develop programs that dynamically generated the required HTML and serve that to the browser rather than build static HTML pages.

The initial program was something like:

```
data _null_;
  file _webout;
  put "Content-type: text/html";
  put ;
  put 'HTML CODE REQUIRED TO GENERATE PAGE';
  put 'HTML CODE REQUIRED TO GENERATE PAGE';
  put 'HTML CODE REQUIRED TO GENERATE PAGE';
  put 'HTML CODE REQUIRED TO GENERATE PAGE';
  put 'HTML CODE REQUIRED TO GENERATE PAGE';
end;
```

Fortunately, by taking this approach we were positioned to move to a table-driven approach when it became apparent that it was possible. The first table in any of our systems contained the required name value pairs to build a drop down box. See how it has evolved in the SELECTION CRITERIA SCREEN section.

Granted, the creation of a drop down box from a table is straightforward and appears in many HTML books. By extending the concept to dynamically generating all of the content on the web page based on information contained in tables transformed our system. The entire Information Delivery System is dynamically generated using a series of SAS programs and tables containing information appropriate to the page being generated.

Repetition is the key to learning and by building several Information Delivery Systems the requisite components began to self identify.

The three components of Information Delivery Systems are as follows.

1. Report Selection Screen
2. Selection Criteria Screens (one per report)
3. Report.

REPORT SELECTION SCREEN

The Report Selection Screen can be broken down into four sections.

- the top of the page (**Source Code 1**)
- the opening of a HTML <form> (**Source Code 2**)
- the drop down box with the list of available reports (**Source Code 3**)
- the button to submit the form (**Source Code 4**)

The top of the page down to the navigation bar is the same for every page that we generate. The %livewire macro is used in all our applications to generate the top portion of the screen. The macro refers to the Cascading Style Sheet which controls the colors and fonts used on the generated page.

The following program pickrpt.sas generates **Figure 1** above using a table with a row of data for each of the reports to be available in this Information Delivery System. Maintaining the table supports the addition of reports or the deletion of reports deemed no longer necessary.

```
*-- Pickrpt.sas ------------------------------*
| The following statement is used to include
| macros written and stored centrally to be made
| available to the entire Information Delivery
| System
*-----------------------------------------------*;
%inc caspgms(casmacs);

options mprint nodate nonumber center
fmtsearch=(casdata);
*-- Source Code 1 ----------------------------*
| Generate the corporate look and feel including
| the cascading style sheet maintained by
| marketing with the colors and font choices
*-----------------------------------------------*;
%livewire;
*-- Source Code 2 ----------------------------*
| Open the form and indicate which program will
| be executed on the server in this case
| criteria
*-----------------------------------------------*;
data _null_;
 file _webout;
   if _n_=1 then
   do;
     put '<form action="' "%superq(_url)" '">';
     put '<input type=hidden name="_service"
           value="'"%superq(_service)" '">';
     put '<input type=hidden name="_program"
           value="caspgms.criteria.sas">';
     put '<input type=hidden name="_debug"
           value="0">';
     put '<input type=hidden name="lang"
           value="'"%superq(lang)" '">';
     put '<p>';
     put '<h3><b>Select a Report:';
   end;
run;
*-- Source Code 3 ----------------------------*
| Generate the drop down box based on the
| information in the data set rptlist with the
| first report pre-selected
*-----------------------------------------------*;
data _null_;
 set caswdat.rptlist end=done;
   file _webout;

   if _n_=1 then
   do;
     put '<select name="reportcd" selected>';
   end;
   put '<option value="' reportcd '">' rpttitle;
   if done then
   do;
     put '</select>';
     put '<p>';
   end;
run;
*-- Source Code 4 ----------------------------*
| Complete the creation of the form including a
| button with the word Process at the bottom of
| the Report Selection Screen
*-----------------------------------------------*;
data _null_;
   file _webout;
   put '<p>';
   put '<input type=submit value="Process">';
   put '</form></b></h3>';
run;
```

SELECTION CRITERIA SCREENS

All Selection Criteria Screens are basically the same. After manually creating numerous Selection Criteria Screens they break down into components the same way the Information Delivery System broke down into components.

The components of the selection criteria screens are:

- the top of the page down to the navigation bar
- the opening of a HTML <form> including the specification of which SAS program is to execute
- the sub-setting selection criteria
- sorting or grouping selection criteria
- the button to submit the form

The first and last components are handled exactly as they were in the REPORT SELECTION SCREEN section above.

The second component regarding the opening of an HTML <form> is extended to dynamically insert the SAS program to be executed to generate the report from a table as follows.

```
data _null_;
 set work.dropdown end=done;
 file _webout;

 length b $30. ;
 if _n_=1 then
 do;
   put '<form  method=get action="'
       "%superq(_url)" '">';
   put '<input type=hidden name="_service"
       value="'"%superq(_service)" '">';
   put '<input type=hidden name="_debug"
       value="131">';
   put '<input type=hidden name="lang"
       value="'"%superq(lang)" '">';
   a = '<input type=hidden name="_program"
       value="';
   c = '">';
   prgline = trim(a)!!trim(sasprog)!!trim(c);
   put prgline;
   put '<b><center>Select Criteria for ';
 end;
run;
```

The %superq() is used to prevent characters that are fine on the web site from interfering with your SAS program.

The component of the Selection Criteria Screen for building the sub-setting selection criteria will be discussed in depth. All of the required selection criteria in the Information Delivery System can be categorized as follows:

1) drop down box (single or multiple selections)
2) textbox (free form text input)
3) textbox pairs (minimum and maximum)

The requirement for a textbox to accept user entered text should be challenged and replaced by a drop down box whenever possible. User input is unavoidable, however, and by applying rigor to the validation of the input you can ensure that the report will produce the desired results.

The creation of drop down boxes for either single selection or multiple selections can be accomplished with one macro. The following table illustrates how that first table with two fields used to create a drop down box has been extended to contain five fields. The following is the format of all tables created to hold information that will appear in a drop down box.

The fields in the table are:

Column	Example	Definition and use
DispText	City	The text to be displayed on the screen beside the box
VarName	CityCd	The variable name will be passed into the SAS program, the variable name will become the name portion of the name value pair and as the macro variable name
Code	01	The code will become the value side of the name value pair and the code is the behind the scene value associated with the displayed description
Desc	Ottawa	The text displayed in the drop down box
Mult	Y	Is this a multi selection box or not (Y/N)

Notice that the format of the table is defined and then one table is created for each collection of information that will be used to create a sub-setting drop down box. The number of tables required is driven by the reporting requirements of the Information Delivery System.

Examples of sets of information that could be stored in these tables are the following: country, state, city, business team, and product. If x reports require a specific sub-setting drop down box you still only need one version of the table.

Re-creating these tables from your refreshed data mart will ensure that as data enters the system it is reflected in the Information Delivery System.

The macro is written to accept the name of the table (dsn) containing the set of information described above.

```
%macro dropdown(dsn);

data _null_;
 file _webout;
 set &dsn end=done;
   if _n_=1 then
   do;
     put Disptext;
     u = '<select name =';
     v = '>';
     w = trim(u)!!trim(varname)!!trim(v);
     put w;
     x = '<option value="';
     y = '">';
     z=trim(x)!!trim(Code)!!trim(y)!!trim(Desc);
     put z;
   end;
   if _n_ gt 1 then
   do;
     x = '<option value="';
     y = '">';
     z=trim(x)!!trim(Code)!!trim(y)!!trim(Desc);
     put z;
   end;
   if done then
   do;
     put '</select>';
     put '</center>';
   end;
run;

%mend dropdown;
```

Macros were created to:

- generate a textbox based on a table of appropriate information
- generate a textbox pair based on a table of appropriate information
- generate a drop down box for the grouping option
- generate a drop down box for the sorting option

Having these macros available allowed us to extend our use of tables again to generate the selection criteria screen required to solicit all sub-setting, grouping, and sorting criteria for all of the reports with one SAS program.

REPORT PRODUCTION

The SAS programs for the reports are written to accept all of the input from the Selection Criteria Screen and generate the report back to the user's browser in HTML format.

By using sessions, available in version 8.2, and creating a temporary data set of the information satisfying the sub-setting, grouping and sorting requirements of the user we were able to deliver the option to either print the report or export the data to a comma separated variable (csv) format at the bottom of the web page.

3. BENEFITS TO THE DEVELOPER

MAINTENANCE

One major benefit of this approach is that the maintenance of the Information Delivery System is reduced to the editing the tables. An example is that the addition of a report requires the addition of some information to a couple of tables and the creation of the SAS program to generate the report itself.

TOOLBOX

You end up with a toolbox of macros that can be utilized in other applications.

4. BENEFITS TO THE BUSINESS USER

There are several major benefits for the business user.

- Current information on the Selection Criteria Screen because the tables are refreshed when the data mart is refreshed.
- Virtually unlimited ability to slice and dice the data.
- Access to the data from all users' desktops rather than a select few analysts who know how to write programs to get their desired reports.

5. REUSE OF THE INFORMATION DELIVERY SYSTEM MODEL

This Information Delivery System model can be reused with relatively little effort. The Information Delivery System that was built using this model took about three days to put together the Report Selection Screen and the Selection Criteria Screens.

The actual reports take less time to develop as well because the routines to accept information from the selection criteria screen functions the same whether you are receiving a list of countries or a list of provinces.

TECHNICAL INFORMATION

The information delivery system is used to surface information out of SAS data sets residing on a Windows NT server running a SAS/IntrNet® version 8.2 Application Server session communicating with a Windows NT server running Internet Information Server with SAS/IntrNet® version 8.2 broker installed.

ACKNOWLEDGEMENTS

I would like to thank the team of programmers and business user who assisted in developing the components of the Information Delivery System.

Desmond Churchill, CAS Team, Export Development Canada
Melinda Philpott, CAS Team, Export Development Canada
Mark O'Brien, DW Team, Export Development Canada
Dion Hull, DW Team, Export Development Canada
Bill Walker, DW Team, Export Development Canada
Panteha Shafaie, Marketing, Export Development Canada
Pil Kim, SAS Institute
Pardeep Sekand, SAS Institute

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at:

Jonathon Hagerman
Export Development Canada
151 O'Connor Street
Ottawa, ON, K1A-3A4
Work Phone: (613) 598-3159
Fax: (613) 598-3142
Email: jhagerman@edc.ca
Web: www.edc.ca