

Paper 150-29

Bells and Whistles for One-Line Text Box to Create a Smarter Where Clause

Ted Guo, U.S. Food and Drug Administration, Rockville, MD

ABSTRACT

SAS/IntrNet-application developers are familiar with adding one or more one-line text boxes into the web pages of their design. When a SAS or HSQL program answers a call from the web page, it receives and processes the contents in the text box. Problems arise when the text box serves as a where clause: A simple typo is likely to result in an unintended and disappointing outcome. In addition, while filling in a text box as a where clause to construct a data subset, for instance, one might pause and wonder about the correct spelling of the name of a particular variable or the correct value of a selected variable. To provide the user a better control of the accuracy while working with the text box as part of a where clause, this paper introduces a practical solution to improve the quality and efficiency. Programming techniques in Application Dispatcher, htmSQL, and JavaScript are tied together to do the trick. Sample SAS and htmSQL programs are provided to the interested user allowing for testing and implementation. Major portions of these sample programs can be inserted into the user's SAS programs or htmSQL programs with only minor changes.

Key words: SAS/IntrNet, htmSQL, Application Dispatcher, JavaScript

OBJECTIVE

SAS/IntrNet-application developers (the developer) have been doing this many times: Adding one or more one-line text boxes or fields into a web page of their design. Sometimes they serve as part of a where clause. On the one hand, a text box, with its free-form nature, allows almost anything text string as its contents. Such flexibility, on the other hand, opens doors for all kinds of human errors. For example, the user may consider creating a subset condition including only three study centers by spelling out: CENTER in ('MD109','VA323','DC797'). Another example could be: SYMPTOM='ABDNA', where ABDNA is a code for an adverse-event symptom. Here is the third example of a subset condition: TREAT=3, indicating the inclusion of one treatment arm alone. In the first example, the user must enter the correct codes for the treatment centers – if the center codes are readily available in memory or in hand. In the second example, what if the user is not certain of the correct spelling for the adverse-event-symptom codes (which is often the case)? For the third example, if the treatment variable (TREAT) is formatted and appears as PLACEBO, ACTDRUG_A, ACTDRUG_B, ACTDRUG_C, and ACTDRUG_D in view, but the user really needs to know what the decoded values are while creating the subset condition. How easy it is to fill in the text field! And how hard to make the data entry error free! This paper shows you how to attach an auxiliary drill-down list to improve the accuracy for the where clause.

SOLUTION

My solution to easing the burden of filling in accurate information into a where clause (text box) is to generate a drill-down list coupled with a command button named Variable-Value Importer (VVI) in this paper. The web pages showing the VVI are created by either an HSQL or a SAS program, working under SAS/IntrNet (SAS Share and SAS/IntrNet Application Dispatcher). The drill-down list, in my design, has dual functions to show either the variable list or the values for a selected variable. The exact name of the variable or the values of a selected variable, while selected, are transferred to the text box by the work JavaScript functions. As a SAS/IntrNet application, the programming techniques in htmSQL, SAS program, and JavaScript are used to do the trick. For demonstration purposes, a fictitious data set named AE101SUB.sas7bdat is used to run the examples in this paper.

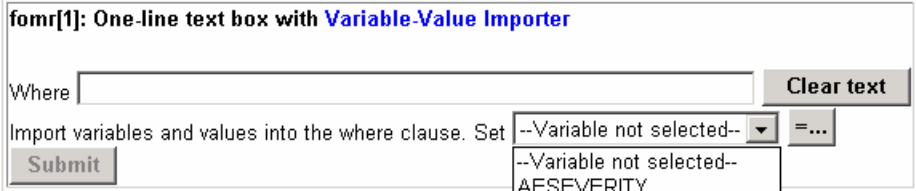
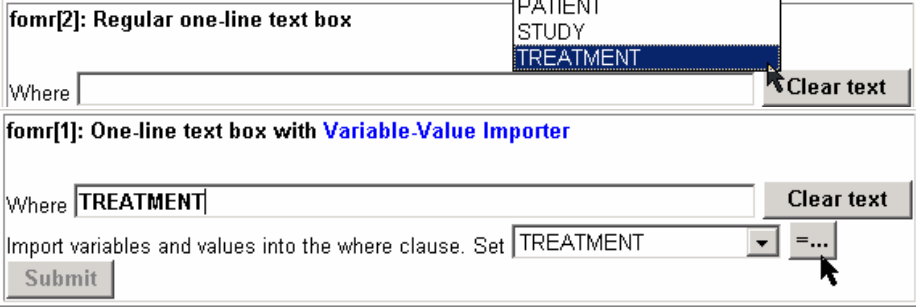
I break down the steps of VVI to illustrate how variable names and values are transferred into a text box to demonstrate to the reader how the error-causing typing work can be eliminated. Each *Step* is followed by *Results* of the action taken. Furthermore, the key portion of the program is shown under *Behind the scene* to explain why and how the job is accomplished.

The complete working HSQL and SAS programs and the JavaScript codes are included in this paper. To test and implement my solution, you must copy the highlighted portions and paste them into your own programs. Modify the sample programs wherever necessary to suit your application-development needs.

TRANSFER VARIABLES AND VALUES INTO THE TEXT BOX – THE IDEA

Explanation	Program
Simple Input: For a regular text box, the user must fill contents manually. VVI Input: Variables and values of a selected variable can be transferred with the help of a VVI How It Works: Take a glance of Selected program code behind the scene	fomr[2]: Regular one-line text box Where Clear text Submit fomr[1]: One-line text box with Variable-Value Importer Where Clear text Import variables and values into the where clause. Set --Variable not selected-- =... Submit <pre> { * Program: EvaluateTemplate.hsqli } { * Job: The drill-down list, generated by {SQL}, includes all variables that can be transferred into the where clauses by a JavaScript } ... { * Create a list of variables } {sql} select distinct name from dictionary.columns where libname=%upcase("&libref") and memname=%upcase("&memname"); {/sql} { * Text box holds the where clause } Where <input type="Text" name="wherecau" value="" onChange="ExistingText[1]=document.forms[1].wherecau.value"> { * Button clear the text } <input type="button" value="Clear text" onClick="document.forms[1].wherecau.value=''; ExistingText[1]=''; NewText=''; document.forms[1].wherecau.focus()"> { * Select a variable } Import variables and values into the where clause. Set <select name="varslst" onChange="updateWhereClauseText(1); document.forms[1].wherecau.focus()"> <option value="" selected>--Variable not selected-- </option> {eachrow} <option value="{&name}">{&name} {/eachrow} </select> <input type="button" value="=..." onClick="openNewWindow(1); document.forms[1].wherecau.focus()"> ... </pre> A similar program, EvaluateTemplate.sas is provided in this paper for interested users to test and implement.

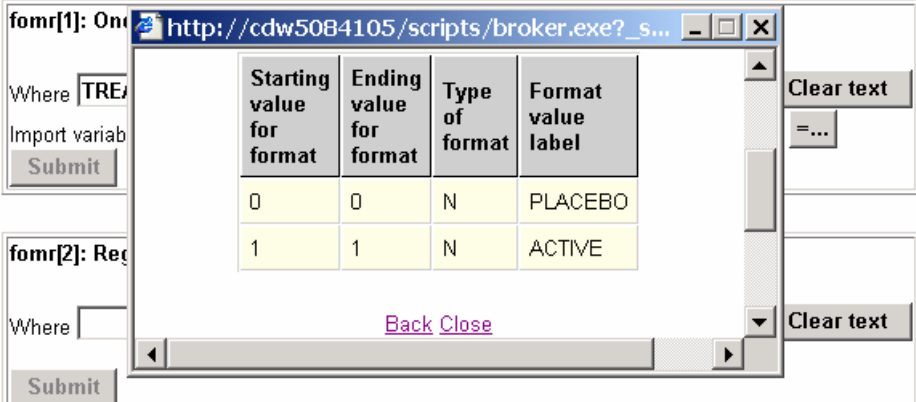
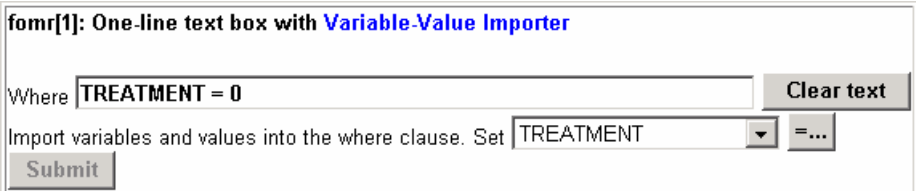
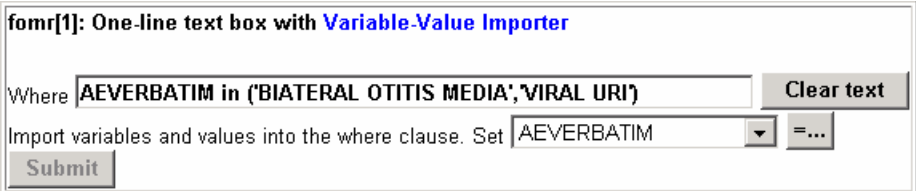
TRANSFER A VARIABLE INTO THE TEXT BOX – STEP ONE

Explanation	Program
VVI at Work: <i>Step 1:</i> Select a variable (TREATMENT) from the drill-down list.	
<i>Result:</i> The name of the variable (TREATMENT) is transferred into the where clause. <i>Behind the scene:</i> On the right is part of the JavaScript programs written in an external JavaScript file. The selection of a variable triggers the function: updateWhereClausesText	 <pre> <!-- File: EvaluateTemplate.js: A JavaScript --> //Job: Transfer variable name from the drill-down list // to the text box ... function expandWhereClauseText(i){ if (NewText!='') { for (var j=0; j<NumberOfForms; j++) { if (j==i) { ExistingText[j]+=NewText if (document.forms[j].wherecau.value=='') { document.forms[j].wherecau.focus()} document.forms[j].wherecau.value =ExistingText[j]} } } } function updateWhereClauseText(i){ var list=document.forms[i].varslst //varlist is the // drill-down list VarNameIs[i]=list.options[list.selectedIndex].value NewText=VarNameIs[i] //NewText is the variable name expandWhereClauseText(i) //Add a variable or a value // to the where clause } ... </pre>

TRANSFER A VALUE INTO THE TEXT BOX – STEP TWO AND BEYOND

Explanation	Program
VVI at WORK: Step 2: Following Step 1, above, click on =... button to find a value for the selected variable. <i>Result:</i> A sub window, Get Values appears, showing the variable's type, format, and all the values.	fomr[1]: One-line text box with Variable-Value Importer Where TREATMENT Clear text Import variables and values into the where clause. Set TREATMENT Submit =... fomr[1]: One-line text box with Variable-Value Importer Where TREATMENT Clear text Import variables and values into the where clause. Set TREATMENT Submit =... fomr[2]: Register Get Values - EvaluateGetValueTemplate.hspl... Where TREATMENT Clear text Import variables and values into the where clause. Set TREATMENT Submit =... Get Values - EvaluateGetValueTemplate.hspl... Select values from variable: TREATMENT (Type: num., Fmt: TRTFMT.) Data set: ecorner.ae101sub --Value Not Selected-- Format definition: TRTFMT. Close
Step 3: Click on the drill-down list of the sub window to search a value to transfer. <i>Result:</i> This list shows all distinct values of the selected variable, similar to the effect of PROC FREQ. By selection, the value, 0 is transferred into the text box.	fomr[1]: One-line text box with Variable-Value Importer Where TREATMENT Clear text Import variables and values into the where clause. Set TREATMENT Submit =... fomr[2]: Register Get Values - EvaluateGetValueTemplate.hspl... Where TREATMENT Clear text Import variables and values into the where clause. Set TREATMENT Submit =... Get Values - EvaluateGetValueTemplate.hspl... Select values from variable: TREATMENT (Type: num., Fmt: TRTFMT.) Data set: ecorner.ae101sub --Value Not Selected-- Format definition: TRTFMT. Close 0 1
	fomr[1]: One-line text box with Variable-Value Importer Where TREATMENT 0 Clear text Import variables and values into the where clause. Set TREATMENT Submit =... fomr[2]: Register Get Values - EvaluateGetValueTemplate.hspl... Where TREATMENT Clear text Import variables and values into the where clause. Set TREATMENT Submit =... Get Values - EvaluateGetValueTemplate.hspl... Select values from variable: TREATMENT (Type: num., Fmt: TRTFMT.) Data set: ecorner.ae101sub --Value Not Selected-- Format definition: TRTFMT. Close 0 < Go back Please note the dual function of the drill-down list. Now it contains the values of the selected variable!

TRANSFER A VALUE INTO THE TEXT BOX – STEP TWO AND BEYOND (CONTINUED)

Explanation	Program
<i>Behind the scene:</i> Part of the program code behind the scene to transfer the selected value from the sub window to the main window. Step 4 (optional): Click on TRTFMT. to view the format definition.	<pre> <!-- EvaluateGetValueTemplate.hspl --> ... <script language=javascript> function getValues(form){ var list=form.values var itemSelected=list.options[list.selectedIndex].value self.opener.NewText=itemSelected self.opener.expandWhereClauseText({&i}) } </script> ... </pre> 
This function enables a quick view of code-label translation. This function can be useful for the user to look up the code-label correspondence in the case where the variable has a long list of values and is formatted. Common examples of formatted variables are: multiple-arm treatments, diseases, patient status, etc.	
<i>Behind the scene:</i> The SAS macro, ShowVarFmts decodes user-defined SAS formats. Step 5: Fix the where clause to obey the syntax of SAS. Example: This is a perfect example to demonstrate the advantage of employing VVI that makes such a tricky entry easier.	<pre> /* This is a compiled SAS macro function decoding the */ /* user-defined SAS format (to be provided on request)*/ %ShowVarFmts(libref=&libref,objsel=&objsel); /* &libref is the LIBREF to the data-format library; /* &objsel is the variable name with a user-defined format; /* Copy this compiled version of /* &libref.sasmacro.ShowVarFmts.macro /* to an appropriate library </pre>   The adverse events included in this where clause are transferred directly from the event list. The letter cases and spaces remain exactly the same as in the data set. Only the parentheses and quotes are typed by the user.

COMPLETE WORKING SOURCE CODE FOR THE USER

LIST OF PROGRAMS

Program	The program calls...	The program is called by...
EvaluateTemplate.htm (main)	EvaluateTemplate.hsql and .sas	
EvaluateTemplate.js	EvaluateTemplate.hsql	
EvaluateTemplate.hsql	EvaluateGetValueTemplate.hsql	
EvaluateTemplate.sas	EvaluateGetValueTemplate.hsql	
EvaluateGetValueTemplate.hsql	EvaluateVarfmts.sas	
EvaluateVarfmts.sas	%ShowVarFmts()	EvaluateGetValueTemplate.hsql

INDIVIDUAL PROGRAMS

EvaluateTemplate.htm

This is the key part of the main page that controls the demo.

```
<!-- Run hsql --><form action="/scripts/htmsql.exe/EvaluateTemplate.hsql">
<input type="hidden" name="libref" value="ecorner">
<input type="hidden" name="memname" value="ae101sub">
<input type="submit" value="EvaluateTemplate.hsql"></form>
<!-- Run SAS --><form action="/scripts/broker.exe">
<input type="hidden" name="_service" value="default">
<input type="hidden" name="_program" value="rootp.EvaluateTemplate.sas">
<input type="hidden" name="libref" value="ecorner">
<input type="hidden" name="memname" value="ae101sub">
<input type="submit" value="EvaluateTemplate.sas"></form>
```

EvaluateTemplate.js

```
<!-- File: EvaluateTemplate.js -=TedGuo=- 09/06/03 -->
```

```
function initText(){
    for (var i=0; i<NumberOfForms; i++) {
        ExistingText[i]=''
        VarNameIs[i]=''
    }
}
function expandWhereClauseText(i){
    if (NewText!='') {
        for (var j=0; j<NumberOfForms; j++) {
            if (j==i) {
                ExistingText[j]+=NewText
                if (document.forms[j].wherecau.value=='') {
                    document.forms[j].wherecau.focus()}
                document.forms[j].wherecau.value=ExistingText[j] }}}
}
function updateWhereClauseText(i){
    var list=document.forms[i].varslst
    VarNameIs[i]=list.options[list.selectedIndex].value
    NewText=VarNameIs[i]
    expandWhereClauseText(i)
}
function openNewWindow(i){
    var winStyle='height=200,width=400,top=295,left=270,scrollbars=yes,'
    winStyle+='resizable=no,status=no,toolbar=no,menubars=no,location=no'
    if (!NewWind[i] || NewWind[i].closed){
        var UrlAddress='http://'+TheservernameIs+
            '/scripts/htmsql.exe/EvaluateGetValueTemplate.hsql?libref='+
            ThelibrefIs+'&memname='+ThememnameIs+'&name='+VarNameIs[i]+'&i='+i
        NewWind[i]=window.open(UrlAddress,"",winStyle)
    }
    else {NewWind[i].focus()}
}
```

EvaluateTemplate.hsql

```

<!-- File: EvaluateTemplate.hsml --TedGuo-- 09/06/03 -->
<head>{query server="{&server_name}:5000"}
<Script Language="JavaScript">
    var NumberOfForms=2; // number of forms in the web page
    var NewWind      = new Array(NumberOfForms)
    var ExistingText = new Array(NumberOfForms)
    var VarNameIs    = new Array(NumberOfForms)
    var NewText
    var TheservernameIs="{&server_name}"
    var ThelibrefIs="{&libref}"
    var ThememnameIs="{&memname}"
</Script>
<Script Language="JavaScript" src="/EvaluateTemplate.js"></Script>
<Script Language="JavaScript">initText()</Script>
<title>Running EvaluateTemplate.hsml</title></head>
<body><h3>Web Page Created by EvaluateTemplate.hsml</h3>
<table><tr><td><form>
<h6>form[0]: One-line text box with <font color=blue>Variable-Value
Importer</font></h6>
{sql}
    select distinct name
    from dictionary.columns
    where libname=%upcase("{&libref}") and memname=%upcase("{&memname}");
{/sql}
Where <input type="Text" name="wherecau" value="" size="65" maxlength="65"
    onChange="ExistingText[0]=document.forms[0].wherecau.value">
<input type="button" value="Clear text"
    onClick="document.forms[0].wherecau.value=''; ExistingText[0]='';
    NewText=''; document.forms[0].wherecau.focus()">
Import variables and values into the where clause. Set
    <select name="varslst"
        onChange="updateWhereClauseText(0); document.forms[0].wherecau.focus()">
        <option value="" selected>--Variable not selected--</option>
        {eachrow}<option value="{&name}">{&name}{/eachrow}
    </select>
<input type="button" value="..."
    onClick="openNewWindow(0); document.forms[0].wherecau.focus()">
<input type="button" value="Submit"></td>
</form></tr></table>

<table><tr><td><form>
<h6>form[1]: One-line text box with <font color=blue>
Variable-Value Importer</font></h6>
{sql}
    select distinct name
    from dictionary.columns
    where libname=%upcase("{&libref}") and memname=%upcase("{&memname}"); {/sql}
Where <input type="Text" name="wherecau" value="" size="65" maxlength="65"
    onChange="ExistingText[1]=document.forms[1].wherecau.value">
<input type="button" value="Clear text"
    onClick="document.forms[1].wherecau.value=''; ExistingText[1]='';
    NewText=''; document.forms[1].wherecau.focus()">
Import variables and values into the where clause. Set
    <select name="varslst"
        onChange="updateWhereClauseText(1); document.forms[1].wherecau.focus()">
        <option value="" selected>--Variable not selected--</option>
        {eachrow}<option value="{&name}">{&name}{/eachrow}
    </select>

```

(Cont'd)

```

<input type="button" value="=..."
  onClick="openNewWindow(1); document.forms[1].wherecau.focus() ">
<input type="button" value="Submit"></td></form></tr></table>

<table><tr><td><form><h6>form[2]: Regular one-line text box</h6>
Where <input type="Text" name="wherecau" value="" size="65" maxlength="65">
<input type="button" value="Clear text"
  onClick="document.forms[2].wherecau.value='';
  document.forms[2].wherecau.focus() ">
<input type="button" value="Submit">
</td></form></tr></table>
<a href="JavaScript: history.back()">Go back</a>{/query} </body>

```

EvaluateTemplate.sas

```

*****;
*** Name: EvaluateTemplate.sas ***;
*** -=TedGuo=- 09/06/03 ***;
*****;
options nosymbolgen nomlogic mprint mstored sasstore=sasuser;
ods listing close;
ods path sasuser.templat(read) sashelp.tmplmst(read);
ods html body=_webout(dynamic nobot notop title="EvaluateTemplate.sas" )
  rs=none style=styles.custom2;
data _null_;
  file _webout;
  put '<head><Script language=JavaScript>';
  put "  var NumberOfForms=2";
  put "  var NewWind      = new Array(NumberOfForms)";
  put "  var ExistingText = new Array(NumberOfForms)";
  put "  var VarNameIs = new Array(NumberOfForms)";
  put "  var NewText";
  put "function initText(){";
  put "    for (var i=0; i<NumberOfForms; i++) {";
  put "      ExistingText[i]='';";
  put "      VarNameIs[i]='' } }";
  put "function expandWhereClauseText(i){";
  put "    if (NewText!='') {";
  put "      for (var j=0; j<NumberOfForms; j++) {";
  put "        if (j==i) {";
  put "          ExistingText[j]+=NewText";
  put "          if (document.forms[j].wherecau.value=='')";
  put "            {document.forms[j].wherecau.focus()}";
  put "          document.forms[j].wherecau.value=ExistingText[j] } }";
  put "    }";
  put "function updateWhereClauseText(i){";
  put "    var list=document.forms[i].varslst";
  put "    VarNameIs[i]=list.options[list.selectedIndex].value";
  put "    NewText=VarNameIs[i]";
  put "    expandWhereClauseText(i)";
  put "    }";
  put "function openNewWindow(i){";
  put "    var winStyle = 'height=200,width=400,top=295,left=270,'";
  put "    winStyle+='scrollbars=yes,'";
  put "    winStyle+='resizable=no,status=no,toolbar=no,'";
  put "    winStyle+='menubars=no,location=no'";
  put "    if (!NewWind[i] || NewWind[i].closed){";
  put "      UrlAddress = '/scripts/htmSQL.exe/'";
  put "      UrlAddress+='EvaluateGetValueTemplate.hspl?'"

```



```

(Cont'd)
put '          UrlAddress+="&libref="+'' "&libref" ''';
put '          UrlAddress+="&memname="+'' "&memname" ''';
put '          UrlAddress+="&name="+VarNameIs[i]';
put '          UrlAddress+="&i="+i';
put '          NewWind[i]=window.open(UrlAddress,"",winStyle)';
put '      } else {';
put '          NewWind[i].focus() }';
put '};';
put '</Script>';
put '<Script Language="JavaScript">initText()</Script>';
put '<title>Running EvaluateTemplate.sas</title></head>'; run;
proc sql noprint;
  create table namefile as
  select distinct name
  from dictionary.columns
  where libname=%upcase("&libref") and memname=%upcase("&memname"); quit;
data _null_;
  file _webout;
  put '<body style="font-family: arial; font-size: 8pt">';
  put '<h3>Web Page Created by EvaluateTemplate.sas</h3>'; run;
data _null_;
  file _webout;
  set namefile end=no_more;
  if _N_=1 then do;
    put '<table><tr><td><form>';
    put '<h6>form[0]: One-line text box with <font color=blue>'
      'Variable-Value Importer</font></h6>'
      'Where <input type="Text" name="wherecau" value="" '
      ' size="65" maxlength="65" ';
    put '  onChange="ExistingText[0]=document.forms[0].wherecau.value">';
    put '<input type="button" value="Clear text" ';
    put '  onClick="'
      " document.forms[0].wherecau.value=''; ExistingText[0]='';"
      " NewText=''; "
      " document.forms[0].wherecau.focus()" '>';
    put 'Import variables and values into the where clause: Set ';
    put '<select name="varslst" onChange="updateWhereClauseText(0);'
      ' document.forms[0].wherecau.focus()">';
    put '<option value="" selected>--Variable not selected--</option>';
    end;
    put '<option value="" name "">' name '</option>';
    if no_more then do;
      put '</select> <input type="button" value="..."';
      put '  onClick="openNewWindow(0);document.forms[0].wherecau.focus()" '
        ' style="font-family: arial; font-size: 8pt; font-weight: bold">';
      put '<input type="button" value="Submit" style="font-family: arial; '
        'font-size: 8pt; color: gray; font-weight: bold">';
      put '</td></form></tr></table>';
    end; run;
data _null_;
  file _webout;
  set namefile end=no_more;
  if _N_=1 then do;
    put '<table>';
    put '<tr><td><form>';
    put '<h6>form[1]: One-line text box with <font color=blue>'
      'Variable-Value Importer</font></h6>'
      'Where <input type="Text" name="wherecau" value="" size="65" ';

```

(Cont'd)

```

    put '    onChange="ExistingText[1]=document.forms[1].wherecau.value">';
    put '<input type="button" value="Clear text" ';
    put '    onClick=""
        " document.forms[1].wherecau.value=''; ExistingText[1]=''; "
        " NewText=''; "
        " document.forms[1].wherecau.focus()"
        '" style="font-family: arial; font-size: 8pt; font-weight: bold">';
    put 'Import variables and values into the where clause: Set ';
    put '<select name="varslst" onChange="updateWhereClauseText(1);'
        '    document.forms[1].wherecau.focus()">';
    put '<option value="" selected>--Variable not selected--</option>';
end;
put '<option value="" name "">' name '</option>';
if no_more then do;
    put '</select> <input type="button" value="..."';
    put '    onClick="openNewWindow(1);document.forms[1].wherecau.focus()"
        '    style="font-family: arial; font-size: 8pt; font-weight: bold">';
    put '<input type="button" value="Submit" style="font-family: arial; '
        'font-size: 8pt; color: gray; font-weight: bold">';
    put '</td></form></tr></table>';
end; run;
data _null_;
    file _webout;
    put '<table><tr><td><form>';
    put '<h6>form[2]: Regular one-line text box</h6>';
    put 'Where <input type="Text" name="wherecau" value size="65" maxlength="65">';
    put '<input type="button" value="Clear text" ';
    put '    onClick="" "document.forms[2].wherecau.value='';
    put '    document.forms[2].wherecau.focus()"> ';
    put '<input type="button" value="Submit"></td></form>';
    put '</tr></table>';
    put '<a href="JavaScript: history.back()">Go back</a>'; run;
ods html close;
ods listing;

```

EvaluateGetValueTemplate.hsqli

```

<!-- EvaluateGetValueTemplate.hsqli -=TedGuo=- 09/06/03 -->
<!-- Open a sub window -->
<html><head>
<title>Get Values - EvaluateGetValueTemplate.hsqli</title>
{query server="{&server_name}:5000"}
{sql}
    select type, format as formatname,
        '<a href="http://'||"{&server_name}"||"/scripts/broker.exe?"||
        '_service=default&_program=rootp.EvaluateVarfmts.sas&libref='||
        '{&libref}'||'&objsel='||trim(left(format))||
        '&_debug=2'||'">'||trim(left(format))||'</a>' as format
    from dictionary.columns
    where libname=%upcase("{&libref}") and
        memname=%upcase("{&memname}") and
        name=%upcase("{&name}")
    order by varnum;
{/sql}
{sql}
    select distinct
    case
        when "{&name}"="" then "Data set: <font color=blue>{&memname}</font>"
        else "Select values from variable: <font color=blue>{&name}</font>"||

```

(Cont'd)

```

"(Type: <font color=blue>{&type}</font>, "||
"Fmt: <font color=blue>{&formatname}</font>"||
"Data set: <font color=blue>{&libref}</font>{&memname}</font>"
end as SETTITLE,
case
  when "{&name}"="" then "NAME"
  else "{&name}"
end as SETVALUE,
case
  when "{&name}"="" then "dictionary.columns where "||
    "libname='NO_SUCH_LIB' "||
    "and memname='NO_SUCH_MEMBER'"
  else "{&libref}</font>{&memname}"
end as SETNAME
from {&libref}</font>{&memname} (obs=1);
{/sql}
<script language=javascript>
function getValues(form){
  var list=form.values
  var itemSelected=list.options[list.selectedIndex].value
  self.opener.NewText=itemSelected
  self.opener.expandWhereClauseText({&i})
}
</script></head>
<body onLoad="document.forms[0].closebutton.focus()">
<h5>{&SETTITLE}</h5>
<form>
{sql}
  select distinct {&SETVALUE} as value
  from {&SETNAME};
{/sql}
{norows}
<font color=red face=arial><b>Variable was not selected!</b></font>
<input type="button" name="closebutton" value="Close" onClick="window.close()">
</form>
{/norows}
<select name="values" onChange="getValues(this.form); closebutton.focus()">
<option value="">--Value Not Selected--</option>
{eachrow}<option value="{&value}">{&value}</option>{/eachrow}
</select>Format definition: {&format}
<input type="button" name="closebutton" value="Close" onClick="window.close()">
</form>
{/query}</body></html>

```

EvaluateVarFmts.sas

```

* File: EvaluateVarFmts.sas -=TedGuo=- 09/06/03 *;
ods listing close;
ods html body=_webout(dynamic nobot notop title=" ") rs=none ;
libname rootp "c:\inetpub\wwwroot";
options mstored sasmstore=rootp;
%global libref objsel noshw;
data _null_;
  file _webout;
  put '<body>';
  put '<a href="javascript: history.back()">Back</a>&nbsp;';
  put '<a href="javascript: window.close()">Close</a>'; run;
%ShowVarFmts(libref=&libref,objsel=&objsel);

```

(Cont'd)

```

%macro IsVarFmtsShown;
%if %superq(noshow)=N %then %do;
  data _null_;
    file _webout;
    put '<a href="javascript: history.back()">Back</a>&nbsp;';
        '<a href="javascript: window.close()">Close</a></font>'; run;
%end;
%else %do;
  data _null_;
    file _webout;
    put 'Note: The selected format is not a user-defined format.'
        '<a href="javascript: history.back()">Back</a>'
        '&nbsp;<a href="javascript: window.close()">Close</a></font>'; run;
%end;
%mend;
%IsVarFmtsShown;
ods html close;
ods listing;

```

Macro function: %ShowVarFmts(libref=,objsel=)

The source code of the SAS macro, the function, %ShowVarFmts(), is not included in this paper. However, the compiled version, available upon request, can be installed in a library of your choosing. Please make changes to the LIBNAME statement in EvaluateVarFmts.sas, above.

REFERENCES

SAS® Web Tools: Advanced Dynamic Solutions Using SAS/IntrNet® Software - Course Notes 57696 (2001), SAS® Institute Inc.
 JavaScript™ Bible, 3rd Edition (1998), Danny Goodman, IDG Books Worldwide, Inc.

ACKNOWLEDGMENTS

The author thanks Ed Nevius, Ph.D., Director of Division of Biometrics II, Office of Biostatistics, Office of Pharmacoeconomics & Statistical Sciences, CDER, FDA, who provided helpful editing. Also, thanks go to John Senior, MD, of Office of Pharmacoeconomics & Statistical Sciences, CDER, FDA, for his advice in regulatory review-tool development. Active SAS/IntrNet users at CDER, FDA remain a source of motivation for researching, developing, and improving software tools.

CONTACT INFORMATION

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

Ted Guo, Ph.D.
 Division of Biometrics II
 Office of Biostatistics
 CDER, FDA
 HFD-715, Room 10B-06 Parklawn Building
 5600 Fishers Lane
 Rockville, MD 20857
 Work Phone: (301) 827-3109
 Fax: (301) 827-5875
 Email: GUOT@CDER.FDA.GOV

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.