Migrating M Databases to the WWW

by Patricia M. Riggie

Abstract

Let's face it, the world is changing. In the past couple of years, there has been an explosive growth on the Internet, and M is a part of this growth. M appears on the Internet in several ways: a listserv, a newsgroup, FTP sites, web sites, and home pages. While these mechanisms increase the visibility of M on the web, they do not really demonstrate the power of M to lurkers of the information super highway. Seeing is believing, and what better way to sell the power of M to the computer market than by providing M databases and applications on the World Wide Web (WWW). Some may think that it is impossible, but M can provide its resources via the WWW. West Virginia CONSULT has developed a solution for increasing the visibility and power of M on the Internet.

Introduction

West Virginia CONSULT (CONSULT) is a statewide, remote access information and communication network designed especially for physicians and allied health professionals in rural settings. The network is both a gateway to existing resources, such as the National Library of Medicine's MEDLARS databases and the World Wide Web (via LYNX), as well as the provider of several proprietary M-based applications. These applications include several databases: CE Link, a searchable continuing education calendar; FDA News, current agency press releases; New Drugs, authoritative drug monographs integrated with an interlibrary loan component; Perl, bibliographic patient education references approved by the American Academy of Family Physicians; and MailMan, an electronic mail system developed by the Department of Veterans Affairs (VA). Telecommunication access is primarily through an X.25 leased line, with toll-free dialup service available in 53 of West Virginia's 55 counties. Partial support for the Network comes from a grant from the National Library of Medicine — an outreach program of the Office of Rural Health at the Robert C. Byrd Health Sciences Center of West Virginia University.

Preliminary Analysis

As with most computer networks, CONSULT requires end-users to register via access/verify codes for login. One universal problem end-users face is the challenge of remembering all the access/verify codes or userid/passwords for their various network accounts. Are all these codes really necessary? For networks that contain sensitive data, the answer is yes. However, for networks that maintain more generic, publicly accessible data, the answer is no. While electronic mail, patient data, and other private information should require restricted login, more general information should not. In bypassing the need for userids and passwords, the database content becomes more accessible to a greater population.

One way to bypass user logins is to utilize existing tools for data retrieval. Many products on the market translate M globals into relational tables for SQL data retrieval. Other software packages, like Visual Basic and Delphi, provide a Graphical User Interface (GUI) applications development tool. Although these software tools provide a robust and effective method for transferring data between the M server and client, they are system dependent and could be costly to implement for very large networks. CONSULT could not afford to put commercial software on all end-user's computers nor could they force individual health professionals to purchase these products. A more practical and cost-effective solution was needed.

Using the Internet sounded promising, but raised an interesting question: Could M communicate with the WWW? No one had done it in the past, and migrating M databases to the WWW sounded like it might be an enormous task for a project that employed a single programmer. Therefore, CONSULT waited to see if other members of the M community would develop a solution. Unfortunately, when one became available, it was not compatible with the Network's platform (UNIX) or its M operating system (Micronetics - MSM). As a result, CONSULT decided to develop its own solution.

CONSULT wanted to make its data available to all users no matter what hardware or software they were using to access the WWW. To accomplish this, a Common Gateway Interface (CGI) script was needed to communicate between the Web server and the M operating system. CGI scripts allow the computer to generate web pages instantly. They also allow the Internet to offer the kind of interactive application users have come to expect. CGI scripts enable the Web server to pass user-specific data to other programs running on the server. They serve as a gateway to the M databases.

There are a wide range of CGI scripting languages. Some of the more popular ones include C/C++, Perl, Java, AppleScript, Tcl, and Visual Basic. CONSULT chose Perl as its CGI scripting lan-

guage for the following reasons:

- its ability to interface with other software
- it is very easy to learn
- it is highly portable
- it is readily available
- it is free

The CONSULT approach to migrate its M biomedical databases to the WWW was to build Perl scripts that allowed M and the WWW to communicate with one another. CE LINK, a continuing education calendar M-based application, will be used to demonstrate the steps in CONSULT's solution.

Migrating M Databases to the WWW

HyperText Markup Language (HTML) is the language of the WWW. It is not a programming language, but rather a formatting language. A client's web browser reads an HTML page from a Web server and displays the formatted page to the user. Details of the HTML language are not covered in this paper but may be found in many text books or by searching the Internet for tutorials on the subject.

Migrating M databases to the WWW was not as difficult as expected. Perl scripts provided a solution to this problem by allowing a two-way conversation between the HTML and M languages. Forms, a subset of HTML, provided a mechanism whereby the user of a web page could enter and return information to the Web server. The Web server processed this information by executing a CGI script and returned an HTML formatted page which matched the user's selection criteria.

Before writing a form or Perl script, programmers need to know what information is going to be extracted from the databases and how that data is going to be presented to the user. CONSULT wanted its WWW applications to look similar to those presented to end-users who sign on to the network using their access/verify codes. This meant that the programmer had to devise a way to perform queries on the databases based upon information submitted by the user. CONSULT also wanted to use the same Perl script and M routine for most of its data retrieval and processing. To use the same search script, certain parameters needed to be passed to the M routine. These included the following:

- M routine to execute
- current label in the Perl script
- current search criteria
- previous search criteria
- M output file
- hidden variables for HTML form processing
- next label in Perl script to be processed

Currently, CONSULT has four biomedical applications available on the WWW. These applications may be accessed by clicking on the *DATABASES* icon from CONSULT's Home Page (address: http://consult.hsc.wvu.edu). Figure 1 illustrates the databases available from the web page.

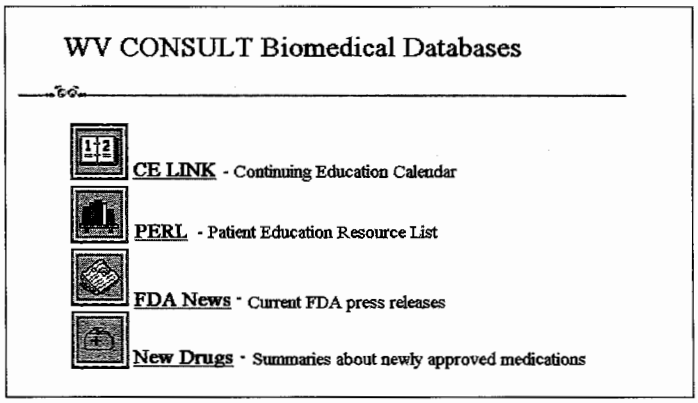

Fig. 1 Databases available from the Web page

Pressing the *CE LINK* icon, another web page appears giving a brief description of the application and a form for the user to choose the type of search to be performed on the database. Figure 2 illustrates CE LINK's search strategies available from the web page.

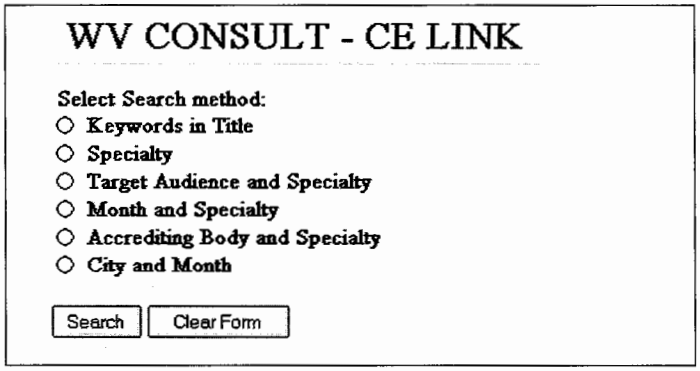

Fig. 2 CE LINK Search Strategies

To search the continuing education database by Target Audience and Specialty, the user would choose that strategy from the form and press the *Search* icon. Figure 3 outlines the portion of HTML code that processes the form.

```
<form method="POST"
action="http://consult.hsc.wvu.edu/cgi-bin/search.pl?start">
Select Search Method:<br>
<input type="radio" name="Search" value="VEWAS1"> Keywords
in Title
<br><input type="radio" name="Search" value="VEWAS2">
Specialty<br>
<input type="radio" name="Search" value="VEWAS3"> Target
Audience and Specialty<br>
<input type="radio" name="Search" value="VEWAS4"> Month and
Specialty<br>
<input type="radio" name="Search" value="VEWAS5"> Accrediting
Body and Specialty<br>
<input type="radio" name="Search" value="VEWAS6"> City and
Month<br>
<p><input type="submit" value="Search">
<input type="reset" value="Clear Form"></font></form>
```

Fig. 3 HTML code to process Search Method

Key elements for processing the form include: the action to be performed, search.pl?start; the name of the radio button, Search; and the values assigned to the radio buttons, VEWAS1, VEWAS2, VEWAS3, VEWAS4, VEWAS5, and VEWAS6. After clicking the Search icon, the Perl script search.pl will be executed. The first part of the script, displayed in Figure 4, pars-

es the form data and starts processing the search request. (See Appendix A for a complete listing of the Perl script search.pl).

```
$host = $ENV{'REMOTE_HOST'};
$data_path = "/var/tmp/";
$msm_path = "/msmmgr/msm";
$cgi_bin_path = "/cgi-bin/";
$label = $ENV{'QUERY_STRING'};
&parse_form_data (*FORM_DATA);
$routine = $FORM_DATA{'Search'};
if (defined($routine)) {
  if (defined($label)) {
    &process_search;
```

Fig. 4 Search pl. - Initialization of form data

In Figure 4, $label will contain the information after the question mark from the form action. This variable determines the steps that will be processed in the Perl script as well as the action to be performed in the M routine. $label can be one of three values: start, the start of a search criteria; getp1, get first search criteria; and getp2, get second search criteria. In the previous step, the form action was set to search.pl?start. The Perl script assigns "start" to the variable $label. The variable $routine contains the value of the Search radio button chosen from the previous step. When the user chose the search strategy Target Audience and Specialty, the value returned from the form data field "Search" was VEWAS3.

The Perl script then invokes subroutine "process_search" to process the search request. Figure 5 contains the section of code in process_search which is executed when $label equals "start."

The first step in processing the search request is to present the

```
sub process_search
{
   local ($args, $curr, $prev, $output, $hvar, $nlabel, $outfile);
   $hvar = "Search," . $routine;
   if ($label eq "start") {
      $output = $data_path . "s1.$host";
      system("rm $output");
      $nlabel = "search.pl?getp1";
      $args = " -r$routine -l$label -o$output -h$hvar -n$nlabel";
   } elsif ($label eq "getp1") {
            :
   }
   &exec_msm;
   &display_file;
```

Fig. 5 Subroutine process_search in Perl script search pl.

user with a list of target audiences from the database. In Figure 5, the subroutine initializes variables to be passed to MSM in order to accomplish this task. The radio button named "Search" and its value "VEWAS3" must be kept throughout the process of a search request. These items are stored in the variable $hvar and become a hidden field in M-generated HTML code. Hidden fields are a useful way of transferring information from one Perl script to another. It allows you to store information in a form without displaying that data to the user. The variable $output is set to a UNIX file that is used to store HTML code generated from the M routines. The last variable, $nlabel, is set to the next form action needed to process the search request.

After initializing the variables, the Perl script makes a system call to execute MSM, passing it the necessary parameters to retrieve the list of target audiences from the database. The following is an example of the parameters passed to MSM:

./msm VAH:VEWZWWW -rVEWAS3 -lstart -o/var/tmp/s1 .riggie.hsc.wvu.edu -hSearch,VEWAS3 -nsearch.pl?getp1

The routine VEWZWWW parses the arguments, stores the information in local variables, and then executes VEWAS3. (See Appendix B for a listing of this M routine). VEWAS3 creates a UNIX file, */var/tmp/s1.riggie.hsc.wvu.edu,* and writes HTML code which produces a web page of the events' target audiences to that file. (See Appendix D for a listing of the HTML code generated from the M routine). After MSM halts, execution returns to the next statement after the system call to MSM in the Perl script. It invokes subroutine display_file which reads and processes the HTML code generated from the M routine and sends this information to the user for display. Figure 6 illustrates the web page generated from the HTML code.

To search the database by Administrators & Managers, the

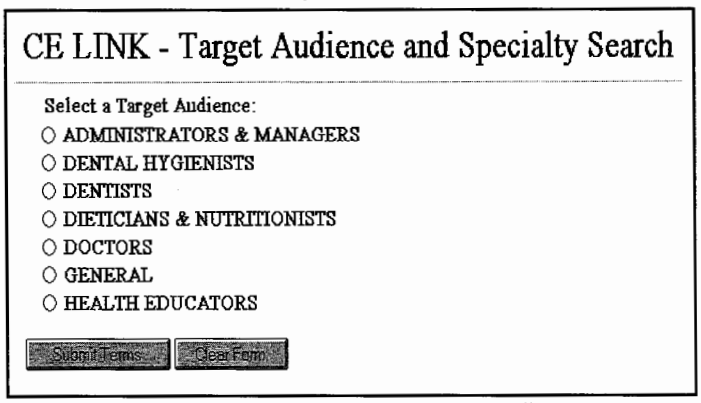

Fig. 6 CE LINK - Target Audience Selection

user chooses that target audience from the form and presses the *Submit Terms* icon. Figure 7 outlines the portion of HTML code that processes the form.

```
<FORM METHOD="POST"
ACTION="/cgi-bin/search.pl?getp1"> <B>
Select a TARGET AUDIENCE:</B><BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE=
"ADMINISTRATORS & MANAGERS">ADMINISTRATORS
& MANAGERS<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="DENTAL
HYGIENISTS">DENTAL HYGIENISTS<BR>
    :
<INPUT TYPE="radio" NAME="SRCH1" VALUE=
"TECHNOLOGISTS, TECHNICIANS & ASSISTANTS">
TECHNOLOGISTS, TECHNICIANS & ASSISTANTS<BR>
<INPUT TYPE="hidden" NAME="Search" VALUE=
"VEWAS3">
<P><INPUT TYPE="submit" VALUE="Submit Terms">
<INPUT TYPE="reset" VALUE="Clear Form"> </FORM>
```

Fig. 7 HTML code to process Target Audience Selection

In Figure 7, notice that the form action is different than the form action listed in Figure 3. Here, the action has changed to search?getp1 which was the "next label in Perl script" parameter passed to MSM. Also notice that the form contains the hidden field initialized in Figure 5 and passed as a parameter to MSM. The Perl script parses the form data and stores the chosen target audience in a UNIX file. The next step in processing the search request is to present the user with a list of specialties from the database. Only those specialties from the events that meet the first search criteria will be included in the list. After initializing the variables in subroutine process_search, the script makes a system call to execute MSM, passing it the necessary parameters to retrieve the specialties. The following is an example of the parameters passed to MSM:

./msm VAH:VEWZWWW -rVEWAS3 -lgetp1
-c/var/tmp/s1terms.riggie.hsc.wvu.edu -hSearch,VEWAS3
-o/var/tmp/s1results.riggie.hsc.wvu.edu -nsearch.pl?getp2

The routine VEWZWWW parses the arguments, stores the information in local variables, and then retrieves the user's selected target audience from the file */var/tmp/s1terms.riggie.hsc.wvu.edu.* It executes routine VEWAS3 searching the database for events targeted towards the user's selection (Administrators & Managers). It creates a UNIX file, */var/tmp/s1results.riggie.hsc.wvu.edu,* and writes HTML code which produces a web page of the specialties for the events that met the first search criteria. (See Appendix D for a listing of the HTML code generated from the M routine). After MSM halts, execution returns to the next statement after the system call to MSM in the Perl script. It invokes subroutine display_file which reads and processes the HTML code generated from the M routine and sends this information to the user for display. Figure 8 illustrates the web page containing the specialties.

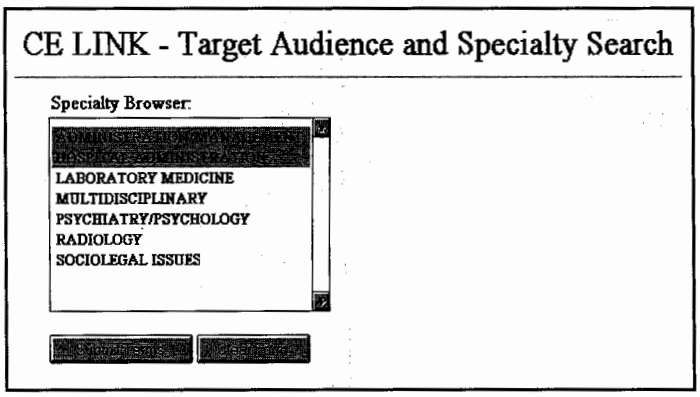

Fig. 8 CE LINK - Specialty Selection

The user chooses one or more specialties from the list and presses the *Submit Terms* icon. In Figure 8, the selected (highlighted) specialties are Administration/Management and Hospital Administration. Figure 9 outlines the portion of HTML code that processes the form.

In Figure 9, notice that the form action is different than the form action listed in Figures 3 and 7. Here the action has changed to

```
<FORM METHOD="POST"
ACTION="/cgi-bin/search.pl?getp2">
<B>SPECIALTY Browser:</B><BR>
<SELECT NAME="SRCH2" SIZE=10 MULTIPLE>
<OPTION>ADMINISTRATION/MANAGEMENT
<OPTION>HOSPITAL ADMINISTRATION
<OPTION>LABORATORY MEDICINE
   :
<OPTION>SOCIOLEGAL ISSUES
</SELECT> <INPUT TYPE="hidden" NAME="Search"
VALUE="VEWAS3"> <P>
<INPUT TYPE="submit" VALUE="Submit Terms"> <INPUT
TYPE="reset" VALUE="Clear Form">
</FORM>
```

Fig. 9 HTML code to process Specialty Selection

search?getp2. Also notice that the form contains the hidden field which is needed throughout the process of the search request.

The Perl script parses the form data and stores the chosen specialties in a UNIX file. The final step in processing the search request is to search the database for the combination of terms submitted by the user. After initializing the variables in subroutine process_search, the script makes a system call to execute MSM, passing it the parameters needed to complete the search. The following is an example of the parameters passed to MSM:

./msm VAH:VEWZWWW -rVEWAS3 -lgetp2
-c/var/tmp/s2terms.riggie.hsc.wvu.edu -hSearch,VEWAS3
-p/var/tmp/s1terms.riggie.hsc.wvu.edu
-o/var/tmp/s2results.riggie.hsc.wvu.edu -n

The routine VEWZWWW parses the arguments, stores the information in local variables, and retrieves the user's selected target audience from the UNIX file */var/tmp/s1terms.riggie.hsc.wvu.edu* and the user's selected specialties from the UNIX file /var/tmp/s2terms.riggie.hsc.wvu.edu. It executes routine VEWAS3 searching the database for events that meet the search criteria. Then it creates a UNIX file, */var/tmp/s2results.riggie.hsc.wvu.edu,* and writes HTML code which produces a web page of the search results. Only those events whose target audience is Administrators & Managers and whose specialty is Administration/Management or Hospital Administration will be stored in the output file. (See Appendix D for a listing of the HTML code generated from the M routine). After MSM halts, execution returns to the next statement after the system call to MSM in the Perl script. It invokes subroutine display_file which reads and processes the HTML code generated from the M routine and sends this information to the user for display. Figure 10 illustrates the web page containing the search results (see next page).

The web page in Figure 10 provides the user with the number of matches found for the search criteria and a detailed record for each of the events that meet the search criteria.

CE LINK - Target Audience and Specialty Search

Search Results: 11 Matches Found.

Search Criteria: **Target Audience = ADMINISTRATORS & MANAGERS and Specialty = ADMINISTRATION/MANAGEMENT or HOSPITAL ADMINISTRATION**

ANNUAL MEETING OF THE ASSOCIATION OF MENTAL HEALTH ADMINISTRATORS

SEP 28, 1996 - OCT 1, 1996
New Orleans LA

Specialty : Psychiatry/Psychology; Administration/Management
Audience : Administrators & Managers; Doctors; Mental Health Therapists
Sponsor : Association of Mental health Administrators
Cost : To Be Announced
CE Credit : Application For Credits Pending
More Info : Contact Maria Helm 708-480-9626

Fig. 10 Search results screen

Changes to M Routines

CONSULT utilizes the VA's FileMan for most of its database design. When FileMan v. 21 became available in 1995, CONSULT modified its existing applications to incorporate the new Database Server (DBS) Application Programmer Interface (API) for accessing data in VA FileMan files. The principle function of the VA's API is to separate database access from user interaction. With FileMan DBS calls, programmers manage user interaction from within their own code by passing parameters to FileMan routines. Instead of communicating directly with the user by writing information to the screen, the DBS stores the data in arrays. It is the programmer's responsibility to determine how the data will be displayed to the end-user.

The VA's DBS API made it possible to start programming applications for alternative front-ends to FileMan databases. It provided a way to access data from applications running outside of M. CONSULT knew that it was just a matter of time before M would be available on the WWW. Hence, the programmer redesigned existing applications to allow for a Graphical User Interface (GUI). Entries in the VA's DIALOG File and DBS calls replaced read and write statements in all M routines.

When CONSULT decided to provide its applications on the Internet, existing routines required minimal changes. The programmer added logic to allow for both character and GUI interfaces. The type of access determined which entries in the DIALOG File the DBS calls returned to the calling routine. Appendix C includes the two main M routines used to search CE LINK by Target Audience and Specialty. The modified logic is in bold text.

Conclusions

The current trend in computer user interface design is GUI. Popup and pulldown windows, colorful graphics, and mouse interaction replace text-filled screens and keyboard interaction. Likewise, the current trend for finding non-sensitive information is on the electronic super highway - the Internet. The electronic marketplace is supplementing and, in some cases replacing, text books, journals and newspapers for gathering information. Now we can proudly say that M is a part of both of these trends.

CONSULT's solution increases the visibility and power of M on the Internet, providing the start of a true integration between M and the rest of the world. Non-M users can now see what M can do for them and how M can be a part of their solution to complex database problems.

For organizations committed to rewriting their current applications to allow for both character and GUI interfaces, CONSULT's solution offers an attractive, practical, and cost-effective method. Visit West Virginia CONSULT's Home Page, http://consult.hsc.wvu.edu, and see how existing technologies integrate M with the WWW.

Appendix A. search.pl - PERL script

```
#!/usr/local/bin/Perl
#
#  Perl script:  search.pl
#  Copyright:  1996 WV CONSULT, West Virginia University
#
$webmaster = "Patricia M. Riggie (riggie.patricia\@consult.hsc.wvu.edu)";
$host = $ENV{'REMOTE_HOST'};
$data_path = "/var/tmp/";
$msm_path = "/msmmgr/msm";
$cgi_bin_path = "/cgi-bin/";

# invoke subroutine to parse form data
&parse_form_data (*FORM_DATA);

# Initialize M application/routine to be executed and the
# current step in the search process.  Then invoke subroutine
# to process search step.
$routine = $FORM_DATA{'Search'};
$label = $ENV{'QUERY_STRING'};
if (defined($routine)) {
   if (defined($label)) {
    &process_search;
   }else {
    &return_error(500, "Internal server error", " -Missing search label");
   }
}else {
   &do_nothing;
}
exit(0);

# Subroutine to process the form data.  Logic copied from CGI
# Programming on the World Wide Web by O'Reilly & Assoc.
sub parse_form_data
{
   local (*data) = @_;
    local ($request_method, $query_string, @key_value_pairs,$key_value, $key,
$value);

   $request_method = $ENV{'REQUEST_METHOD'};
   if ($request_method eq "GET") {
      $query_string = $ENV{'QUERY_STRING'};
   }elsif ($request_method eq "POST") {
```

```
    read (STDIN, $query_string, $ENV{'CONTENT_LENGTH'});
  }else {
    &return_error(405, "Server Error", "Method not allowed");
  }
  @key_value_pairs = split (/&/, $query_string);
  foreach $key_value (@key_value_pairs) {
    ($key, $value) = split (/=/, $key_value);
    $value =~ tr/+/ /;
    $value =~ s/%([\dA-Fa-f][\dA-Fa-f])/pack ("C", hex($1))/eg;
    $key =~ tr/+/ /;
    $key =~ s/%([\dA-Fa-f][\dA-Fa-f])/pack ("C", hex ($1))/eg;
    if (defined($data{$key})) {
      $data{$key}= join ("\n", $data{$key}, $value);
    }else {
      $data{$key}= $value;
    }
  }
}

# Subroutine to process the search request
sub process_search
{
  local ($args, $curr, $prev, $output, $hvar, $nlabel, $outfile);

# Initialize hidden field to be passed from one form to another.
# Based on $label, initialize variables to be passed to MSM
  $hvar = "Search," . $routine;

  if ($label eq "start") {
# Initialize output file for selection of 1st search criteria
# Initialize next label to be processed in Perl script
    $output = $data_path . "s1.$host";
    system("rm $output");
    $nlabel = "search.pl?getp1";
    $args = " -r$routine -l$label -o$output -h$hvar -n$nlabel";

  }elsif ($label eq "getp1") {
# Initialize output file to store M records that met 1st search
#   criteria and store this as the current file.
# Get user selection of 1st search criteria
# If search request only has one search criteria, initialize
#   output file to store M search results.  If search request
#   has 2 search criteria, initialize output file for selection
#   of 2nd search criteria and initialize next label to be
#   Processed in Perl script.
    $outfile = $data_path . "s1terms.$host";
    system("rm $outfile");
    &get_array("SRCH1");
    $curr = $outfile;
    if ($routine eq "VEWAS1" || $routine eq "VEWAS2" ||
      $routine eq "VEWCS1" || $routine eq "VEWCS2") {
      $output = $data_path . "s1results.$host";
    }else {
      $output = $data_path . "s2.$host";
      $nlabel = "search.pl?getp2";
    }
    system("rm $output");
    $args = " -r$routine -l$label -c$curr -o$output -h$hvar -n$nlabel";
  }elsif ($label eq "getp2") {
# Initialize output file to store M records that met 2nd search
#   criteria and store this as the current file.
# Get user selection of 2nd search criteria
# Initialize output file to store M search results that met both
#   Search criteria.
    $outfile = $data_path . "s2terms.$host";
    system("rm $outfile");
    &get_array("SRCH2");
    $curr = $outfile;
    $prev = $data_path . "s1terms.$host";
    $output = $data_path . "s2results.$host";
    system("rm $output");
    $args = " -r$routine -l$label -c$curr -p$prev -o$output -h$hvar -n$nlabel";
  }else {
    &return_error(500, "Internal Server Error", " -Invalid label [$label]");
  }

# Invoke subroutine to execute MSM.
# Invoke subroutine to read and display HTML code generated
# from M routine.
  &exec_msm;
  &display_file;
exit (0);
}

#  Subroutine that reads the form data and stores the information
# in an output file.
sub get_array
{
  local ($data)= @_;
  local ($term);
  $term = $FORM_DATA{$data};
  if (open (OUTFILE, "+>$outfile")) {
    print OUTFILE $term, "\n";
    close (OUTFILE);
  }else {
    &return_error(500, "Internal Server Error",
      "Cannot write to the form file [$outfile].");
  }
}

# Subroutine that executes MSM
sub exec_msm
{
  local ($msm);
  chdir $msm_path;
  $msm = "./msm VAH:VEWZWWW" . $args . " >/var/tmp/log";
  system($msm);
  chdir $cgi_bin_path;
}

# Subroutine that prints the html file returned from MSM
sub display_file
{
  print "Content-type: text/html", "\n\n";
  if (open (FILE, "<$output")) {
    sleep (2);
    while (<FILE>) {
```

```
      print $_;
   }
     close (FILE);
   }else {
     &return_error(500, "Internal Server Error",
        "Cannot read from the form file [$output].");
   }
exit (0);
}

# Subroutine that  prints an html header if an error occurs
# in the Perl script
sub print_head
{
   local ($title) = @_;
   print "Content-type: text/html", "\n\n";
   print "<html><head><title>", $title, "\n";
   print "</title></head>", "\n";
      print '<body bgcolor="#FFFFCF" text="#000000" link="0000FF"
vlink="#FF00FF" alink="#00475B">';
   print "<body><h2><center>", $title, "</center></h2>", "\n";
}

# Subroutine that  prints an html footer if an error occurs
# in the Perl script
sub print_end
{
   print "</body></html>", "\n";
}

# Subroutine that does nothing -no search criteria selected
sub do_nothing
{
   print "Content-type: text/plain\n";
   print "Status: 204 No Response\n\n";
   print "you should only see this message if your browser does ";
   print "not support the status code of 204\n";
}

# Subroutine that prints the actual error message when an
# error occurs in the Perl script.
sub return_error
{
   local ($status, $keyword, $message) = @_;
   &print_head("WV CONSULT Biomedical Databases");
   print "Status: ", $status, " ", $keyword, " ", $message,"<br>\n";
   print "Contact webmaster: ", $webmaster, "\n\n";
   &print_end();
   exit(1);
}
```

Appendix B. VEWZWW - New M Routine for Migrating M Databases to the WWW

```
VEWZWWW ;VEW/PMR -WWW Browser Entry Point  04/15/96 [5/15/96
3:42pm]
     ;;1.0
     ;Copyright 1996 West Virginia University, WV CONSULT
     ;
     ; % parameter passed from MSM
     ; -rROUTINE     -Routine to Call
     ; -lLABEL       -Current label in CGI Script
     ; -cCURR_SRCH CRITERIA -Current Search Criteria
     ; -pPREV_SRCH_CRITERIA -Previous Search Criteria
     ; -oOUTPUT_FILE  -Output File to pass to CGI Script
     ; -hHIDDEN_VARS  -Hidden Variable needed for html Forms
     ; -nNEXT_LABEL   -Next label in CGI Script to process
     ;
     N
RTN,LABEL,TERMS,OUTFILE,NEWLBL,HIDENAME,HIDEVAL,CURR
SRCH,PREVSRCH
     N QUIETIN,PKG,VEWHFS,BOLD,NORMAL,U,DTIME,DUZ
     S
X="EXIT^VEWZWWW",@^%ZOSF("TRAP"),QUIETIN=1,BOLD="
<B>",NORMAL="</B>"
     I '$G(DUZ) S U="^",DTIME=300,DUZ=0,DUZ(0)="@"
     D GETARGS
     S PKG=$E(RTN,1,4)
     KILL ^TMP(PKG,$J)
     D @RTN
     KILL ^TMP(PKG,$J)
     Q
GETARGS ; Parse % parameter passed from MSM
     N I,ARG,ARG2,FLAG,HIDEVARS,CFILE,PFILE
     S ^TEMP(1)=%
     F I=1:1 Q:$P($G(%)," ",I)']""  D
     . S ARG=$P(%," ",I),FLAG=$E(ARG,1,2)
     . I FLAG="-r" S RTN=$E(ARG,3,99),PKG=$E(ARG,3,6) Q
     . I FLAG="-l" S LABEL=$E(ARG,3,99) Q
     . I FLAG="-c" S CFILE=$E(ARG,3,99) Q
     . I FLAG="-p" S PFILE=$E(ARG,3,99) Q
     . I FLAG="-o" S OUTFILE=$E(ARG,3,99) Q
     . I FLAG="-n" S NEWLBL=$E(ARG,3,99) Q
     . Q:FLAG'="-h"
     . S
HIDEVARS=$E(ARG,3,999),HIDENAME=$P(HIDEVARS,",",1),HIDE-
VAL=$P(HIDEVARS,",",2) Q
     I $G(CFILE)]"" D GETVAL(CFILE,.CURRSRCH)
     I $G(PFILE)]"" D GETVAL(PFILE,.PREVSRCH)
     Q
GETVAL(FILE,SRCH)     ; Get Current/Previous Search Criteria
     S FILE=$G(FILE),SRCH=$G(SRCH)
     N REC,VEWIO,VEWHFS,CNT,IOP
     ;Get "HFS" Device
     S VEWHFS=0,CNT=0
     S
IOP="HFS",%ZIS("HFSNAME")=FILE,%ZIS("HFSMODE")="R"
     D ^%ZIS G:POP EXIT
     S VEWIO="O IO:"_IOPAR X VEWIO U IO
     F  R REC:DTIME Q:$$EOF  S CNT=CNT+1,SRCH(CNT)=REC
     S SRCH(0)=CNT
EXIT    ;Close HFS File
     D ^%ZISC
     Q
EOF()   ;
     I ^%ZOSF("OS")["MSM" D
     .S EOFCHK="S ZC=$ZC" X EOFCHK
```

```
    .I ZC<0 S VEWHFS=1
    E  I ^%ZOSF("OS")["VAX" D
    .S EOFCHK="S %ZE=$ZE" X EOFCHK
    .I %ZE<0 S VEWHFS=1
    Q VEWHFS
VEWAS1  D S1^VEWAS1 Q
VEWAS2  D S2^VEWAS2 Q
VEWAS3  D S3^VEWAS3 Q
VEWAS4  D S4^VEWAS4 Q
VEWAS5  D S5^VEWAS5 Q
VEWAS6  D S6^VEWAS6 Q
VEWBNAP D MAIN^VEWBNAP Q
VEWCS1  D S1^VEWCS1 Q
VEWCS2  D S2^VEWCS2 Q
VEWDNDD D MAIN^VEWDNDP(19181.12) Q
    Q

VEWZWW2 ;VEW/PMR -WWW Browser Utilities  04/15/96 [5/28/96 2:17pm]
    ;;1.0
    ;Copyright 1996 West Virginia University, WV CONSULT
    ;
HTMLHEAD(TITLE) ; Print html header, Search Summary
    ; Open output file to store HTML code and print HTML header from dia-
log file to output file
    ; Dialog Number          Dialog Text
    ;  19188.001             <HTML><HEAD><TITLE>
|1|</TITLE></HEAD><BODY BGCOLOR="#FFFFCF"
TEXT="#000000"
    ;                                  LINK="0000FF"
VLINK="#FF00FF" ALINK="#00475B"><BODY><H2><CEN-
TER>|1|</CENTER></H2><HR>
    S TITLE=$G(TITLE)
    D OPENW
    D
BLD^DIALOG(19188.001,TITLE),MSG^DIALOG("MW","",70,1),CLEA
N^DILF
    Q
    ;
FORMST(NEWLBL,TSRCH,SELNAME,TNUM)     ; Print form header
(Single Value Form)
    ; Build array for form variables to be passed to dialog file
    ;     NEWLBL - Form action field (-n parameter passed to MSM)
    ;     TSRCH - Type of search, e.g. specialty (initialized in routine call from
-r parameter)
    ;      SELNAME - Form SELECT NAME Field
    ;      TNUM - Dialog number (2 possible: One for single selecction, other
for multiple selection)
    ; Print HTML Form Header from dialog file to output file
    ; Dialog Number          Dialog Text
    ;  19188.014             <FORM METHOD="POST"
ACTION="/cgi-bin/|1|"><B>Select a |2|:</B><BR>
    ;  19188.004             <FORM METHOD="POST"
ACTION="/cgi-bin/|1|"><B>|2| Browser:</B><BR>
    ;                        <SELECT NAME="|3|" SIZE=10 MUL-
TIPLE>
    S NEWLBL=$G(NEWLBL),TSRCH=$G(TSRCH),SELNAME
=$G(SELNAME),TNUM=$G(TNUM)
    S ARRAY1(1)=NEWLBL,ARRAY1(2)=TSRCH
    S ARRAY1(3)=$$EZBLD^DIALOG(SELNAME)
    D BLD^DIALOG(TNUM,.ARRAY1),MSG^DIALOG
("MW","",70,1),CLEAN^DILF
    Q
    ;
FORMEND(MULT,HIDENAME,HIDEVAL) ; Print form ending
    ; Build array for form variables to be passed to dialog file
    ;      MULT - Flag to determine single or multiple value selection.  If
MULT, print end-of-selection HTML code
    ;      HIDENAME - Hidden Field in form
    ;      HIDEVAL - Value of Hidden Field in form
    ; Print HTML Form Ending from dialog file to output file
    ; Dialog Number    Dialog Text
    ;  19188.005       </SELECT>
    ;  19188.006       <INPUT TYPE="hidden" NAME="|1|"
VALUE="|2|">
    ;  19188.007       </FORM>
    ;  19188.008       <P><INPUT TYPE="submit" VALUE="Submit
Terms"><INPUT TYPE="reset" VALUE="Clear Form">
    S MULT=$G(MULT),HIDENAME=$G(HIDENAME),HIDE-
VAL=$G(HIDEVAL)
    I $G(MULT) D
BLD^DIALOG(19188.005),MSG^DIALOG("MW","",70,1),CLEAN^DIL
F
    S ARRAY2(1)=HIDENAME,ARRAY2(2)=HIDEVAL
    D
BLD^DIALOG(19188.006,.ARRAY2),MSG^DIALOG("MW","",70,1),CLE
AN^DILF
    D
BLD^DIALOG(19188.008),MSG^DIALOG("MW","",70,1),CLEAN^DILF
    D
BLD^DIALOG(19188.007),MSG^DIALOG("MW","",70,1),CLEAN^DILF
    Q
    ;
HTMLEND ; Print html ending
    ; Print HTML Ending from dialog file to output file and close file
    ; Dialog Number Dialog Text
    ;  19188.009          <P><IMG SRC="/images/eyesline.gif" alt=
"SEPARATOR LINE"><P><CENTER><A HREF=
    ; "http://consult.hsc.wvu.edu/celink.html"><img src= images/calendar.jpg"
    ; alt="GoTo " align=bottom> CE LINK</a><a href="http://consult.
    ; hsc.wvu.edu/perl.html"><img src="/images/books.gif" alt="GoTo "
    ; align=bottom> Perl</a><a href="http://consult.hsc.wvu.edu/
    ; fdanews.shtml"><img src= "/images/news.gif" alt="GoTo " align=bot
    ; tom>
    ;FDA NEWS</a><a href="http://consult.hsc.wvu.edu/newdrug.
    ; shtml"><img src="/images/docbag.jpg" alt="GoTo "align=bottom>
    ; New Drugs</a></center><P><HR><A HREF=
    ; "http://consult.hsc.wvu.edu">
    ; <IMG SRC="/images/home.gif" align=bottom alt="Back To">CON
    ; SULT Home Page</A><BR>Maintained by D. Alex Yohn, HREF=
    ; "mailto:yohn.alex@consult.hsc.wvu.edu">
    ; yohn.alex@consult.hsc.wvu.edu </A><BR> and Pat Riggie,<A
    ; HREF="mailto:riggie.patricia
    ; @consult.hsc.wvu.edu">riggie.patricia@consult.hsc.wvu.edu</A>
    ; <P><A HREF="http://www.hsc.wvu.edu"><IMG
    ; SRC="/images/goback.gif" align=center alt="Back To">Robert C. Byrd
    ; Health Sciences Center</A> of <A HREF="http://www.wvu.edu">
```

```
    ; West Virginia University</A>
    ;19188.002                    </BODY></HTML>
    D BLD^DIALOG(19188.009),MSG^DIALOG
("MW","",70,1),CLEAN^DILF
    D
BLD^DIALOG(19188.002),MSG^DIALOG("MW","",70,1),CLEAN^DILF
    W !
    D CLOSE
    Q
    ;
BLDSEL(SELECT,CURRSRCH) ; Build Search Criteria from html input
    ; Store array of selected terms from user into global. (Build search criteria
global)
    S SELECT=$G(SELECT),CURRSRCH=$G(CURRSRCH)
    N I
    S @(SELECT_"0)")=0
    F I=1:1:CURRSRCH(0) D
    . S @(SELECT_I_")")=CURRSRCH(I)
    . S @(SELECT_"0)")=@(SELECT_"0)")+1
    Q
    ;
OPENW  ;Trap any File Errors (EOF)
    ; Get "HFS" Device and open output file to write HTML code
    N VEWIO,IOP,%ZIS
    S IOP="HFS2",%ZIS("HFSNAME")=OUTFILE,%ZIS("HFS-
MODE")="W"
    D ^%ZIS G:POP CLOSE
    S VEWIO="O IO:"_IOPAR X VEWIO U IO
    Q
    ;
CLOSE  ;Close HFS File
    ; Close output file
    D ^%ZISC
    Q
```

Appendix C. VEWAS3 - CE LINK Search by Target Audience and Specialty Routines

```
VEWAS3 ;VEW/PMR -CE LINK -Search by Target Audience and Specialty
[5/15/96 1:41pm]
    ;;3.0;VEWA CE LINK; 05/01/96
    ;Copyright 1996 West Virginia University, WV CONSULT
S3   ; Entry Point for Target Audience and Specialty Search
    ; If QUIETIN then interface is QUI; otherwise interface is character based.
    I $G(QUIETIN) D S3WWW Q
    N ANS,RSP,MSG1,MSG2,BOLD,NORMAL,PFLAG,MNTH,WVS-
TATE,MAX1,MAX2
    N
PKG,TYPE,TSRCH1,TSRCH2,SFILE1,SFILE2,NODE1,NODE2,NODE3,X
,FILE,APP
    N LOOKUP1,LOOKUP2,SELECT1,SELECT2,FILENO
    S X="ERR^VEWAS3",@^%ZOSF("TRAP")
    ; Initialize variables, build array of keywords from Pointed-To File, print
summary for type of search and process the search.
    D INITS
    KILL ^TMP(PKG,$J)
    D BLDPKEY^VEWZS1(PKG,NODE1,TSRCH1,SFILE1)
    D PNTSUMM^VEWZUT3(19180.003,APP)
    D PENTER^VEWZUT3
    D SEARCH
EXIT  ;
    KILL ^TMP(PKG,$J)
    Q
    ;
SEARCH ;Process Search
    ; Character Interface
    ; Kill previous search globals
    ; Invoke Browser to present user with search criteria 1 terms and build
global of CE events that match 1st search criteria
    ; Invoke Browser to present user with search criteria 2 terms whose records
matched 1st criteria and invoke PROCESS to print matches
    F D Q:RSP=4
    . KILL ^TMP(PKG,$J,"SELECT1"),^TMP(PKG,$J,"SELECT2")
    . KILL ^TMP(PKG,$J,"MATCH1"),^TMP(PKG,$J,"MATCH2")
    . KILL ^TMP(PKG,$J,"MATCHES")
    . D BROWSER^VEWZS1(LOOKUP1,SELECT1,MAX1,TSRCH1)
    . I '$D(^TMP(PKG,$J,"SELECT1")) W @IOF,!!! S RSP=4 Q
    . D
PTROR^VEWZS3(PKG,NODE1,TSRCH1,"SELECT1","MATCH1")
    . D BLDMATCH^VEWZS1(PKG,NODE3,TSRCH2,SFILE2)
    . D BROWSER^VEWZS1(LOOKUP2,SELECT2,MAX2,TSRCH2)
    . I '$D(^TMP(PKG,$J,"SELECT2")) W @IOF,!!! Q
    . D
PTROR^VEWZS3(PKG,NODE2,TSRCH2,"SELECT2","MATCH2")
    . D DIFFAND^VEWZS3(PKG,"MATCH1","MATCH2","MATCHES")
    . I '$D(^TMP(PKG,$J,"MATCHES")) D Q
    . . D NOMATCH^VEWZS2(19187.026,19184.603,19184.601,19185.002)
    . D PROCESS
    Q
S3WWW  ; WWW Browser Start Point
    ; GUI Interface
    ; Invoke INIT to initialize variables
    ; Determine step in search process (variable LABEL is -l parameter
passed to MSM) and invoke S3P1, S3P2, or S3P3
    N MSG1,MSG2,LOOKUP1,LOOKUP2,SELECT1,
SELECT2,FILENO,ARRAY1,ARRAY2,TITLE
    N APP,TYPE,TSRCH1,TSRCH2,SFILE1,SFILE2,
NODE1,NODE2,NODE3,X,FILE,MULT
    S TITLE=$$EZBLD^DIALOG(19184.607)_$$EZBLD^DIA-
LOG(19187.507)
    S TITLE=TITLE_$$EZBLD^DIALOG(19184.003)
    D INITS
    I LABEL="start" D S3P1 Q
    I LABEL="getp1" D S3P2 Q
    I LABEL="getp2" D S3P3 Q
    Q
    ;
S3P1  ; Print Target Audience to Output file
    ; GUI Interface   -l paramerer passed to MSM was "start"
    ; Invoke HTMLHEAD to print HTML header to output file and invoke
FORMST to print form header to output file.
    ; Invoke BLDPKEY to write Target Audience keywords from Pointed-To
File to output file.
    ; Invoke FORMEND to print form ending to output file and HTMLEND
to print HTML ending to output file and close file.
    N SELNAME,SUMM,TNUM
    S
```

```
SELNAME=19187.515,SUMM=19180.015,TNUM=19188.014,MULT=0
    D HTMLHEAD^VEWZWW2(TITLE)
    D BLD^DIALOG(SUMM,APP),MSG^DIALOG
("MW","",70,1),CLEAN^DILF
    D FORMST^VEWZWW2(NEWLBL,TSRCH1,SELNAME,TNUM)
    D BLDPKEY^VEWZS1(PKG,NODE1,TSRCH1,SFILE1)
    D FORMEND^VEWZWW2(MULT,HIDENAME,HIDEVAL)
    D HTMLEND^VEWZWW2
    W!
    Q
    ;
S3P2  ; Match Target Audiences, Get specialties for those events
    ; GUI Interface   -l paramerer passed to MSM was "getp1"
    ; Invoke HTMLHEAD to print HTML header to output file.
    ; Invoke BLDSEL and PTROR to create global for selected Target
Audience
    ; Invoke FORMST to print form header to output file
    ; Invoke BLDMATCH to find specialties of CE Events that match selected
Target Audience
    ; Invoke FORMEND to print form ending to output file and HTMLEND
to print HTML ending to output file and close file.
    N SELNAME,TNUM
    S SELNAME=19187.516,TNUM=19188.004,MULT=0
    KILL ^TMP(PKG,$J)
    S QUIETIN=0 D BLDPKEY^VEWZS1(PKG,NODE1,TSRCH1,SFILE1)
S QUIETIN=1
    D HTMLHEAD^VEWZWW2(TITLE)
    D BLDSEL^VEWZWW2(SELECT1,.CURRSRCH)
    D PTROR^VEWZS3(PKG,NODE1,TSRCH1,
"SELECT1","MATCH1")
    D FORMST^VEWZWW2(NEWLBL,TSRCH2,SELNAME,TNUM)
    S MULT=1
    D BLDMATCH^VEWZS1(PKG,NODE3,TSRCH2,SFILE2)
    D FORMEND^VEWZWW2(MULT,HIDENAME,HIDEVAL)
    D HTMLEND^VEWZWW2
    W!
    Q
    ;
S3P3  ; Match Target Audience and Specialties selected
    ; GUI Interface   -l paramerer passed to MSM was "getp2"
    ; Invoke BLDPKEY to build globals of target audiences and specialties.
    ; Invoke HTMLHEAD to print HTML header to output file.
    ; Invoke BLDSEL and PTROR to create global for selected Target
Audience and Specialties.
    ; Invoke DIFFAND to find CE Events that match target audience and spe-
cialties selected.
    ; Invoke PROCESS to write search results to output file and HTMLEND
to print HTML ending to output file and close output file.
    KILL ^TMP(PKG,$J)
    S QUIETIN=0
    D BLDPKEY^VEWZS1(PKG,NODE1,TSRCH1,SFILE1)
    D BLDPKEY^VEWZS1(PKG,NODE2,TSRCH2,SFILE2)
    S QUIETIN=1
    D HTMLHEAD^VEWZWW2(TITLE)
    D BLDSEL^VEWZWW2(SELECT1,.PREVSRCH)
    D BLDSEL^VEWZWW2(SELECT2,.CURRSRCH)
    D PTROR^VEWZS3(PKG,NODE1,
TSRCH1,"SELECT1","MATCH1")
    D PTROR^VEWZS3(PKG,NODE2,TSRCH2,
"SELECT2","MATCH2")
    D DIFFAND^VEWZS3(PKG,"MATCH1","MATCH2","MATCHES")
    D PROCESS
    D HTMLEND^VEWZWW2
    W!
    Q
    ;
INITS  ; Initialize Variables
    N TEMP
    S FILENO=19181.01
    D FILE^DID(FILENO,"","GLOBAL NAME","TEMP")
    S FILE=$G(TEMP("GLOBAL NAME")),NODE1=FILE_""
"ATARG"""
    S NODE2=FILE_""""ASPEC""""",NODE3=FILE_""""ADASPEC"""""
    S SFILE1="^VEW(19180.05,",SFILE2="^VEW(19180.04,"
    S APP=$$EZBLD^DIALOG(19184.607)
    S TSRCH1=$$EZBLD^DIALOG(19184.603),PKG="VEWA"
    S TSRCH2=$$EZBLD^DIALOG(19184.601),
TYPE=$$EZBLD^DIALOG(19187.025)
    S LOOKUP1="^TMP(PKG,$J,TSRCH1,",SELECT1
="^TMP(PKG,$J,""SELECT1"","
    S LOOKUP2="^TMP(PKG,$J,TSRCH2,",
SELECT2="^TMP(PKG,$J,""SELECT2"","
    I $G(QUIETIN) Q
    D INIT^VEWZUTL,STOREMO^VEWZUT3
(.MNTH),GETWV^VEWZUT3(.WVSTATE)
    S RSP=3,PFLAG=0,MAX1=1,MAX2=3
    Q
    ;
PROCESS ; Process Results
    N IDX
    S IDX=$O(^TMP(PKG,$J,"SELECT1",0))
    S MSG2=$$EZBLD^DIALOG(19184.403)_
$G(^TMP(PKG,$J,"SELECT1",IDX))
    S MSG2=MSG2_$$EZBLD^DIALOG(19184.404)
    D CMSG^VEWZS2(PKG,"SELECT2",.MSG1,.MSG2)
  D RESULTS^VEWAPS1(PKG,FILENO,
FILE,.ANS,.RSP,MSG1,MSG2,.PFLAG,.MNTH)
    I '$G(QUIETIN) W @IOF,!!!
    Q
    ;
ERR   ; Error Processing
    D ERROR^VEWZUT3
    G EXIT

VEWAPS1 ;VEW/PMR -CE LINK -Process Search Criteria Results [5/16/96
10:30am]
    ;;3.0;VEWA CE LINK;05/01/96
    ;Copyright 1996 West Virginia University, WV CONSULT
    ;
RESULTS(PKG,FILENO,FILE,ANS,RSP,MSG1,MSG2,PFLAG,MNTH)
;Process Search Results
    ; Initialize variables, iInvoke STORE to store results in ascending date
order
    ; If GUI interface, invoke PNTHTML to write results to output file;
Otherwise Character Interface, invoke PRINT to display result to screen.
    N SUB,TOTRCD,RCD,REPLY,PGM
    S
PKG=$G(PKG),FILENO=$G(FILENO),ANS=$G(ANS),RSP=$G(RSP),M
```

```
SG1=$G(MSG1)
    S MSG2=$G(MSG2),PFLAG=$G(PFLAG),SUB="",TOTRCD=0
    I FILENO=19181.01 S PGM="D ^VEWACP"
    E  S PGM="D ^VEWATP"
    D STORE
    I $G(QUIETIN) D PNTHTML Q
    I TOTRCD>15 D ^VEWAPS2 Q
    S SUB=$O(^TMP(PKG,$J,"EVENTS","")),RCD=1
    F  D PRINT Q:RSP>2
    KILL ^TMP(PKG,$J,"EVENTS")
    Q
STORE  ; Store Results in Ascending Date Order
    N NAME,BDATE,REC
    KILL ^TMP(PKG,$J,"EVENTS")
    F  S SUB=$O(^TMP(PKG,$J,"MATCHES",SUB)) Q:SUB=""  D
    . S NAME=$E($P(@(FILE_"SUB,0)"),U,2),1,40)
    . S BDATE=$P(@(FILE_"SUB,0)"),U,3),REC=BDATE_NAME_SUB
    . S ^TMP(PKG,$J,"EVENTS",REC)=SUB,TOTRCD=TOTRCD+1
    Q
PRINT  ; Display Events (Search Results < 16)
    ; Character Interface
    ; Invoke DETHEAD to print screen header, DETPNT to print detail
record, and PROMPT1 to print action line.
    ; Invoke DETANS and DETCHK to get user action and verify user selec-
tion
    F  D  Q:REPLY
    . S REPLY=1,RSP=1
    . D DETHEAD^VEWZS6(MSG1,MSG2,RCD,TOTR-
CD,1,"","","","",19187.504,"")
    . D DETPNT^VEWZS8(PKG,FILENO,"EVENTS",SUB,1)
    . D PROMPT1^VEWZS4(RCD,1,TOTR-
CD,19181.003,19181.004,19181.002,19181.001)
    . D DETANS^VEWZS5(.RCD,TOTRCD,.RSP,.ANS)
    . D
DETCHK^VEWZS7(PKG,PGM,.RCD,TOTRCD,"EVENTS",.SUB,.PFLA
G,.RSP,.REPLY,ANS)
    Q
PNTHTML ; Print Results for html search
    ; QUI Interface
    ; Print Search result header and invoke DETPNT to write detailed records
to output file.
    ; Dialog Number Dialog Text
    ;  19188.011     <H3>Search Results: |1| Matches Found.</H3>
    ;  19188.012     <PRE>
    ;  19188.013     </PRE>
    N SUB S SUB=""
    D
BLD^DIALOG(19188.011,TOTRCD),MSG^DIALOG("MW","",70,1),CLEA
N^DILF
    D
BLD^DIALOG(19188.012),MSG^DIALOG("MW","",70,1),CLEAN^DILF
W!
    D HEADER^VEWZS6
    F  S SUB=$O(^TMP(PKG,$J,"EVENTS",SUB)) Q:SUB=""  D
    . D DETPNT^VEWZS8(PKG,FILENO,"EVENTS",SUB,1)
    D
BLD^DIALOG(19188.013),MSG^DIALOG("MW","",70,1),CLEAN^DILF
    Q
```

Appendix D. M Generated HTML Code to Display Events' Target Audiences

```
HTML><HEAD><TITLE>CE LINK - Target Audience and Specialty
Search </TITLE> </HEAD>
<BODY BGCOLOR="#FFFFCF" TEXT="#000000" LINK="0000FF"
VLINK="#FF00FF" ALINK="#00475B">
<BODY><H2><CENTER>CE LINK - Target Audience and Specialty
Search</CENTER></H2><HR>
First select a Target Audience.  After choosing a Target Audience, a new form
will be displayed for you to select up to 3 terms from the SPECIALTY
Browser.<P>
 <PRE>To select more than 1 Specialty:
     IBM/Compatible Users:  HOLD DOWN the "Ctrl" KEY
     MAC Users:             HOLD DOWN the "Apple" KEY
 while using your mouse to select the next specialty.</PRE><P>
 If you select more than 1 specialty, CE LINK combines EACH specialty with
the specified Target Audience.<HR>
 <FORM METHOD="POST" ACTION="/cgi-bin/search.pl?getp1">
 <B>Select a TARGET AUDIENCE:</B><BR>
 <INPUT TYPE="radio" NAME="SRCH1" VALUE="ADMINISTRA-
TORS & MANAGERS">
ADMINISTRATORS & MANAGERS<BR>
 <INPUT TYPE="radio" NAME="SRCH1" VALUE="DENTAL
HYGIENISTS">DENTAL HYGIENISTS
<BR> <INPUT TYPE="radio" NAME="SRCH1" VALUE="DEN-
TISTS">DENTISTS<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="DIETICIANS &
NUTRITIONISTS">
DIETICIANS& NUTRITIONISTS<BR>
<INPUT TYPE="radio" NAME="SRCH1"
VALUE="DOCTORS">DOCTORS<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="GENERAL">GEN-
ERAL<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="HEALTH EDUCA-
TORS">HEALTH EDUCATORS <BR> <INPUT TYPE="radio"
NAME="SRCH1" VALUE="HUMAN SERVICES
PERSONNEL">HUMAN SERVICES PERSONNEL<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="MENTAL HEALTH
THERAPISTS">MENTAL HEALTH THERAPISTS<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="NURSES (RN,
LPN)">NURSES (RN,LPN)
<BR> <INPUT TYPE="radio" NAME="SRCH1" VALUE="PHARMA-
CISTS">PHARMACISTS<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="PHYSICIAN
ASSISTANTS">PHYSICIAN ASSISTANTS<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="REHABILITATION
THERAPISTS">
REHABILITATION THERAPISTS<BR>
<INPUT TYPE="radio" NAME="SRCH1" VALUE="TECHNOLO-
GISTS, TECHNICIANS & ASSISTANTS">TECHNOLOGISTS, TECHNI-
CIANS & ASSISTANTS<BR>
<INPUT TYPE="hidden" NAME="Search" VALUE="VEWAS3">
<P><INPUT TYPE="submit" VALUE="Submit Terms"> <INPUT
TYPE="reset" VALUE="Clear Form"> </FORM> <P><IMG
SRC="/images/eyesline.gif" alt="SEPARATOR LINE"><P> <CENTER>
<A HREF="http://consult.hsc.wvu.edu/celink.html"><img src="/images/cal-
endar.jpg" alt="GoTo " align=bottom> CE LINK</a>
```

```
<a href="http://consult.hsc.wvu.edu/perl.html"><img src="/images/books.gif"
alt="GoTo " align=bottom> Perl</a>
<a href="http://consult.hsc.wvu.edu/fdanews.shtml"><img
src="/images/news.gif" alt="GoTo " align=bottom> FDA NEWS</a>
<a href="http://consult.hsc.wvu.edu/newdrug.shtml"><img
src="/images/docbag.jpg" alt="GoTo " align=bottom> New Drugs</a>
</center> <P><HR>
<A HREF="http://consult.hsc.wvu.edu"><IMG SRC="/images/home.gif"
align=bottom alt="Back To">
CONSULT Home Page</A><BR>
Maintained by D. Alex Yohn, <A
HREF="mailto:yohn.alex@consult.hsc.wvu.edu">
yohn.alex@consult.hsc.wvu .edu </A><BR>
      and Pat Riggie, <A HREF=
"mailto:riggie.patricia@consult.hsc.wvu.edu">
riggie.patricia@consult.hsc.wvu.edu</A><P>
 <A HREF="http://www.hsc.wvu.edu"><IMG SRC="/images/goback.gif"
align=center alt="Back To">
Robert C. Byrd Health Sciences Center</A> of <A
HREF="http://www.wvu.edu">West Virginia University</A>
</BODY></HTML>
<HTML><HEAD><TITLE>CE LINK - Target Audience and Specialty
Search</TITLE></HEAD>
 <BODY BGCOLOR="#FFFFCF" TEXT="#000000" LINK="0000FF"
VLINK="#FF00FF"
ALINK="#00475B">
<BODY><H2><CENTER>CE LINK - Target Audience and Specialty
Search</CENTER></H2><HR>
<FORM METHOD="POST" ACTION="/cgi-bin/search.pl?getp2">
<B>SPECIALTY Browser:</B><BR>
<SELECT NAME="SRCH2" SIZE=10 MULTIPLE>
<OPTION>ADMINISTRATION/MANAGEMENT
<OPTION>HOSPITAL ADMINISTRATION
<OPTION>LABORATORY MEDICINE
<OPTION>MULTIDISCIPLINARY
<OPTION>PSYCHIATRY/PSYCHOLOGY
<OPTION>RADIOLOGY
<OPTION>SOCIOLEGAL ISSUES
</SELECT>
<INPUT TYPE="hidden" NAME="Search" VALUE="VEWAS3">
<P><INPUT TYPE="submit" VALUE="Submit Terms"> <INPUT
TYPE="reset" VALUE="Clear Form">
</FORM>
<P><IMG SRC="/images/eyesline.gif" alt="SEPARATOR LINE"><P>
<CENTER>
<A HREF="http://consult.hsc.wvu.edu/celink.html"><img src="/images/cal-
endar.jpg" alt="GoTo " align=bottom> CE LINK</a>
<a href="http://consult.hsc.wvu.edu/perl.html"><img src="/images/books.gif"
alt="GoTo " align=bottom> Perl</a>
<a href="http://consult.hsc.wvu.edu/fdanews.shtml"><img
src="/images/news.gif" alt="GoTo " align=bottom> FDA NEWS</a>
<a href="http://consult.hsc.wvu.edu/newdrug.shtml"><img
src="/images/docbag.jpg" alt="GoTo " align=bottom> New Drugs</a>
</center> <P><HR>
<A HREF="http://consult.hsc.wvu.edu"><IMG SRC="/images/home.gif"
align=bottom alt="Back To">
CONSULT Home Page</A><BR>
Maintained by D. Alex Yohn, <A
HREF="mailto:yohn.alex@consult.hsc.wvu.edu">
yohn.alex@consult.hsc.wvu .edu </A><BR>
      and Pat Riggie, <A HREF=
"mailto:riggie.patricia@consult.hsc.wvu.edu">
riggie.patricia@consult.hsc.wvu.edu</A><P>
<A HREF="http://www.hsc.wvu.edu"><IMG SRC="/images/goback.gif"
align=center alt="Back To">
 Robert C. Byrd Health Sciences Center</A> of <A
HREF="http://www.wvu.edu">West Virginia University</A>
</BODY></HTML>

<HTML><HEAD><TITLE>CE LINK - Target Audience and Specialty
Search</TITLE></HEAD>
<BODY BGCOLOR="#FFFFCF" TEXT="#000000" LINK="0000FF"
VLINK="#FF00FF"  ALINK="#00475B">
<BODY><H2><CENTER>CE LINK - Target Audience and Specialty
Search</CENTER></H2><HR>
<H3>Search Results: 8 Matches Found.</H3> <PRE>
<B>
Search Criteria:  Target Audience = ADMINISTRATORS & MANAGERS
and Specialty
          = ADMINISTRATION/MANAGEMENT</B><P>
<B>ANNUAL MEETING OF THE AMERICAN HOSPITAL ASSOCIA-
TION  </B>
AUG  5,1996 - AUG  9,1996
Philadelphia  PA

Specialty :  Administration/Management; Multidisciplinary
Audience  :  Administrators & Managers; General
Sponsor   :  American Hospital Association
Cost      :  To Be Announced
CE Credit :  Application For Credits Pending
More Info :  Contact 312-422-3000
             :
             :
<B>
RISKY BUSINESS:  MASTERING THE NEW BUSINESS OF HEALTH
 </B>
OCT 16,1996 - OCT 17,1996
Healthcare Forum Leadership Center
San Francisco  CA

Specialty :  Administration/Management; Multidisciplinary
Audience  :  Doctors; Administrators & Managers
Sponsor   :  Healthcare Forum
Cost      :  To be announced
CE Credit :  Application For Credits Pending
More Info :  Contact (415)356-4300

</PRE> <P><IMG SRC="/images/eyesline.gif" alt="SEPARATOR
LINE"><P><CENTER><A
HREF="http://consult.hsc.wvu.edu/celink.html">
<img src="/images/calendar.jpg" alt="GoTo " align=bottom> CE
LINK</a>
<a href="http://consult.hsc.wvu.edu/perl.html"><img src="/images/books.gif"
alt="GoTo " align=bottom> Perl</a>
<a href="http://consult.hsc.wvu.edu/fdanews.shtml"><img
src="/images/news.gif" alt="GoTo " align=bottom> FDA NEWS</a>
<a href="http://consult.hsc.wvu.edu/newdrug.shtml"><img
src="/images/docbag.jpg" alt="GoTo " align=bottom> New Drugs</a>
```

```
</center>
 <P><HR> <A HREF="http://consult.hsc.wvu.edu"><IMG
SRC="/images/home.gif" align=bottom alt="Back To">
 CONSULT Home Page</A><BR> Maintained by D. Alex Yohn, <A
HREF="mailto:yohn.alex@consult.hsc.wvu.edu">
yohn.alex@consult.hsc.wvu .edu </A><BR>
        and Pat Riggie, <A HREF=
"mailto:riggie.patricia@consult.hsc.wvu.edu">
riggie.patricia@consult.hsc.wvu.edu</A><P>
 <A HREF="http://www.hsc.wvu.edu"><IMG SRC="/images/goback.gif"
align=center alt="Back To"> Robert C. Byrd Health Sciences Center</A>
of <A HREF="http://www.wvu.edu">West Virginia University</A>
 </BODY></HTML>
```

M

References

[1] VA Fileman User's Manual, Version 21, Department of Veterans Affairs, 1995

[2] Randal L. Schwaretz, Learning Perl, O'Reilly & Associates, Inc., 1993

[3] Shishir Gundavaram, CGI Programming on the World Wide Web, O'Reilly & Associates, Inc. 1993

[4] West Virginia CONSULT, the Statewide Health Information Network, http://consult.hsc.wvu.edu, June 1996, Contact: Patricia Riggie, riggie.patricia@consult.hsc.wvu.edu

---

Patricia Riggie is the Lead Systems Programmer/Analyst for West Virginia CONSULT. Her email address is riggie.patricia@consult.hsc.wvu.edu.

All source code is displayed as presented by the author. Questions concerning the source code should be directed to the author.