Session F1/Q1

Custom Forms and JavaScript In Innovative's WebPAC: Improved Functionality You Can Build Yourself

Custom Form Basics

Suggested Reading

Prior to attending this session, you should be familiar with the material contained on these pages:

CSDirect's Custom Forms FAQ
http://csdirect.iii.com/faq/custform.shtml

CSDirect's Examples of Custom Forms
http://csdirect.iii.com/faq/exampleforms.shtml

This material will be referred to during the program, and some of the basics will be covered, but we will be moving pretty quickly. You will find the discussion easier to follow if you know this material.

Program attendees should also be somewhat familiar with the JavaScript language. You need not be an expert, but some previous experience with JavaScript would be helpful.

Introduction

What are custom forms?

Whenever any command link (such as the /patroninfo screen) or command function (such as the Modify PIN screen) is requested by a user, the WebPAC dynamically generates an HTML document, based on a template that is hard-coded by III. Custom forms are simply a way for you to specify exactly how that template should appear, and in some cases, how that template should function.

How do they work?

Before processing any command link or command function, the WebPAC checks the /screens directory for an HTML document that corresponds to the requested command. If the WebPAC finds such a file, it uses this file as a template, instead of its internal template, to generate the requested screen.

```
<!--{token}-->
```

Tokens are the core of how custom forms work – they are the nuts and bolts out of which a custom form is constructed. According to CSDirect, “Tokens are placeholders in an HTML source document which output HTML.” For example, the `<!--{errmsg}-->` token could be replaced by the text “Sorry, cannot locate patron record.”

Tokens are also used to define how the custom form should react under certain conditions. Quoting CSDirect again:

> Tokens whose name begins with "if" are used for conditional processing of blocks of HTML code and tokens. The end of such a block is marked with an `<!--{xif}-->` token. For example:
>
> ```
> <!--{ifpinerrormessage}-->
> <!--{pinerrormessage}-->
> <!--{xif}-->
> ```

You may insert any additional HTML code you desire at any point within a custom form. It is this ability that enables you to match the design of your WebPAC’s system-generated pages to the design used throughout the rest of your catalog (i.e. on all srchhelp and opacmenu pages). It is also this ability that allows us to add new functionality to our WebPACs.

What custom forms are available, and what do they do?

The following forms are available for customization:

FORM	COMMAND	NEW HTML SCREEN
View Your Own Record patron verification screen	/patroninfo	pverify_web.html
Web Access Management custom patron verification screen	NONE	pverify_wam.html
Web Access Management new PIN patron verification screen	NONE	pverify_wam_newpin.html
Request verification form	NONE	pverify3_web.html
Suggestions form	/suggest	suggest_web.html
Items Library Should Acquire form	/acquire	acquire_web.html
Modify Your PIN form	NONE	newpin.html
Inter-Library Loan Request form (Book)	/illb	illbook.html
Inter-Library Loan Request form (Chapter)	/illc	illchapter.html
Inter-Library Loan Request form (Dissertation)	/illd	illdissert.html
Inter-Library Loan Request form (Technical Report)	/illt	illreport.html
Inter-Library Loan Request form (Government Documents)	/illg	illgov.html
Inter-Library Loan Request form (Conference Proceedings)	/illp	illconf.html
Inter-Library Loan Request form (Journal)	/illj	illjournal.html

Custom forms use tokens to provide dynamic functionality. This dynamic functionality falls into two categories: presenting system error messages, and maintaining form state.

During this program, we will discuss pverify_web.html and pverify3_web.html. As noted above, these forms correspond to the /patroninfo screen, and the item request screen.

Simple Changes

Change layout to better match catalog design

There are a number of good examples available on the CSDirect custom forms example page (http://csdirect.iii.com/faq/exampleforms.shtml). These provide basic starting points from which you can begin your work.

Add detailed instructions for patron login

A common, simple customization we add to our MOBIUS WebPACs is an additional informational table to help patrons to correctly format their patron ID. Sample code from our Quest cluster can be found in **Listing 1**, and can be seen in production at:
http://quest.missouri.edu/search/Yyo&SORT=D&searchscope=5/Yyo&SORT=D&searchscope=5/1,53,53,B/request&F=Yyo&1,1,&SORT=D

AVS Search Forms

Why can AVS forms be considered custom forms?

Alta Vista Search keyword search forms (srchhelp_X.html, srchhelp_Y.html, etc.) also use tokens to provide dynamic functionality. Just like the command custom forms, this dynamic functionality falls into two categories: presenting system error messages, and maintaining form state (i.e. if patrons wish to modify an advanced keyword search, tokens are responsible for remembering their previous search).

Of course, AVS forms have a much simpler job to do than the command custom forms. However, the concepts of adding JavaScript to portions of AVS forms are the same as adding JavaScript to command custom forms. And the results can be spectacular.

Adding JavaScript to the Mix: Basic Form Validation

Theory

If you would like to add JavaScript form validation to a custom form, you must first ask the form to use a validation function:

```
<FORM METHOD=POST onSubmit="return formCheck(this)">
```

Where do you put the JavaScript?

Now that the form is asking for a validation function (formCheck), you need to put the function somewhere. The best approach is to add the required validation functions directly below the token that provides the HTML element you would like to validate. This ensures that the element in question exists in your patron's browser before the validation code exists in their browser. For example:

```
<!--{ifneedNNA}-->
<P>
<!--{needbyprompt}-->
<!--{needby}-->
<!-- javascript validation code -->
<script language="JavaScript">
<!-- hide from older browsers

function formCheck(form) {

      // add JavaScript to validate the fields here
      // for a working example, see Listing 2
    return true;
}

//-->;
</script>

</p>

<!--{else}-->
<!-- passthrough javascript code, this is important, keep reading -->
<script language="JavaScript">
<!-- hide from older browsers
function formCheck(form) {
    return true;
}
//-->;
</script>
<!--{xif}-->
```

Pass-through JavaScript

In the example above, I've added an extra portion of code to the `<!--{ifneedNNA}-->` ... `<!--{xif}-->` block: an `<!--{else}-->` block. This block is necessary for cases when the `<!--{ifneedNNA}-->` block is not needed (say we decide that we'd no longer like to offer the option). In that case, the form validation function will still be called by the FORM tag, however, the function will always return true; having no reason not to continue, the request is submitted to the WebPAC.

What If my patron doesn't have JavaScript?

In this example, the patron will never process JavaScript code, because it is wrapped in HTML comments. This code will therefore not interfere with a patron's use of your WebPAC. Will this always be the case for any code you write? Yes, as long as you are simply validating patron input on a form. If, however, you are attempting to add new functionality to an AVS form (see page 5), you will have be more careful.

Examples

Not wanted after date input validation

See **Listing 2**.

Patron ID validation

See **Listing 3**.

Working With AVS Forms: Validation and New Functionality

Adding Form Validation to AVS Forms

The problem: Sorting by Relevance Can Misbehave If Fed Improper or Incomplete Input

We discovered in 2001 that if a patron fails to supply both a “Year After” and “Year Before” when limiting a keyword search by date, and they also request a sort by Relevance, an incorrect number of stars would display in a search results set.

The solution: Validate Year After and Year Before

A simple form validation script addresses this problem. The theory is the same as validating a command custom form: add an onSubmit call for a validation function to the FORM tag, and add the validation function to the code. See **Listing 4**.

Forms without tokens: completely control of the look of your form

The problem: some tokens contain markup you don’t want

Most tokens used by the AVS forms contain some sort of HTML markup. This markup can severely limit the design possibilities of your AVS form. You may also prefer to use a different sort of markup (such as radio buttons instead of a drop-down box).

The solution: replace tokens with hard-coded HTML, and JavaScript

If you have experimented with replacing a token with hard-coded HTML, you probably have discovered that the ‘modify search’ functionality no longer works, at least for the field you have replaced. That is to be expected: the WebPAC cannot work magic on flat HTML—it will only populate tokens with previous values when your patron wishes to modify their search.

We can get around this problem with JavaScript. A fairly simple collection of functions can parse the GET portion of the URL provided by the ‘modify search’ button. These values can then be written back into the form fields on the form.

See **Listing 5**.

Pros/Cons

The obvious pro is that, using this approach, you have complete freedom in designing your AVS forms. The cons are: patrons without JavaScript will be required to re-type all previous entries every time they select the ‘modify search’ button; and, any values that were previously provided by tokens will need to be manually re-coded every time they change. The former issue is probably an acceptable compromise; the latter may not be, depending on your staff workload, and the likelihood a change affecting of any options.

Real world: mix tokens and custom fields

In a real world application, you can mix and match tokens with hard-coded fields. This could make your JavaScript more complex: the example I give simply loops through all available fields on the AVS form, and populates with appropriate values obtained from the GET. You would need to account for any fields that do not populate correctly, as a result of being already populated by the token. In most cases, the form should work well as-is, with the example JavaScript from **Listing 5**, but, as always, you should test before releasing your code.

New functionality

A working example of the code referenced below can be found at:
http://bridges.missouri.edu/search/X

Radio buttons for material type

To be discussed in the presentation.

See **Listing 5**.

Single year entry

To be discussed in the presentation.

See **Listing 5**.

What If my patron doesn't have JavaScript?

If you add new functionality that requires JavaScript, the best way to ensure that you are not excluding patrons who do not have JavaScript-capable browsers is to write out the new element using JavaScript itself. For example, in Listing 5, the following code writes out the new single year entry box:

```
<!-- single year text box -->
<SCRIPT LANGUAGE="JavaScript">

<!--

// single year text box for JavaScript-capable broswers
document.write("Single Year: <input type=text size=4 maxlength=4 name=Sy value=''
onChange='resetDaAndDBIfNecessary();'>   OR   ");

// -->

</SCRIPT>
```

Since the element is only written out if the patron is running JavaScript, patrons without JavaScript will simply never see the element.

Applying JavaScript in this way ensures that JavaScript is only serving to enhance the experience of the patron, while at the same time ensuring that the WebPAC still serves all patrons, regardless of whether their browser supports JavaScript.

Coordinator/Presenter

Hardy Pottinger
MOBIUS Consortium
pottingerhj@umsystem.edu
http://mco.mobius.missouri.edu/

Presenter

John McCullough
Innovative Interfaces
jmccullough@iii.com
http://www.iii.com/

```
<!--{toplogo}-->
<FORM METHOD=POST>
<p>
<!--{iferrmsg}-->
<FONT COLOR=RED SIZE=+2><EM>
<!--{iferrmsgisPIN}-->
<!--{newpinmsg}-->
<!--{else}-->
<!--{errmsg}-->
<!--{xif}-->
</EM></FONT><BR><BR><BR>
<!--{xif}-->
<!--{enterinfo}--></p>
<table width="100%" border="0" cellspacing="0" cellpadding="2">
<tr>
<td valign="top" width="1%">

<TABLE>
<!--{ifneedpatronname}-->
<TR><TD WIDTH=1%></TD><TD class="blurb">
<!--{nameexample}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{nameprompt}-->
</TD><TD>
<!--{name}-->
</TD></TR>
<tr><td colspan="3"> </td></tr>
<!--{xif}-->
<!--{ifneedpatronverify}-->
<TR><TD width="1%"></TD><TD class="blurb">
<!--{barcodeexample}-->
</TD></TR>
<TR><TD class="blurb" width="1%" nowrap>
<!--{barcodeprompt}-->
</TD><TD>
<!--{barcode}-->
</TD></TR>
<!--{ifneedspin}-->
<!--{iferrmsgisPIN}-->
<TR><TD COLSPAN=2>
<!--{newpinmsg}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{pin1prompt}-->
</TD><TD>
<!--{pin1}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{pin2prompt}-->
</TD><TD>
<!--{pin2}-->
</TD></TR>
<!--{else}-->
<TR><TD></TD><TD class="blurb">
<!--{pininstructions}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{pinprompt}-->
</TD><TD>
<!--{pin}-->
</TD></TR>
<!--{xif}-->
<!--{xif}-->
<!--{xif}-->
<tr><td colspan="2">
<br>
<!--{submit}-->
</td></tr>
</TABLE>

</td>

<td valign="top" align="right">
<table width="200" border="0" cellspacing="0" cellpadding="2" bgcolor="#CDC8B1">
```

```
   <tr>
     <td>
       <p><b><font size="-1" face="Arial, Helvetica, sans-serif">Campus ID Help</font></b></p>
       <p class="blurb"><font size="-2" face="Arial, Helvetica, sans-serif">
         &middot; Central Methodist College - <b>CMC</b><br>
         &middot; Central Missouri State University - <b>CMSU</b><br>
         &middot; Missouri Valley College - <b>MVC</b><br>
         &middot; State Fair Community College - <b>SFCC</b></font></p>

       <p class="blurb"><font size="-2" face="Arial, Helvetica, sans-serif">
         Type your ID number (dashes are not necessary) plus your institution
     code.
       <br>    For example: <tt><font size="-1"
       color="#CC0000">999999999MVC</font></tt> would be a Campus
         ID for a student at Missouri Valley College.<br><br></font></p>
       </td>
   </tr>
</table>

</td>
</tr>
</table>

<P>

</FORM>
<TABLE WIDTH=100%>
<TR><TD>
<!--{startover}-->
</TD></TR>
</TABLE>

<!--{botlogo}-->

</BODY>
</HTML>
```

```
<!--{toplogo}-->
<FORM METHOD=POST onSubmit="return formCheck(this)">
<p>
<!--{iferrmsg}-->
<FONT COLOR=RED SIZE=+2><EM>
<!--{iferrmsgisPIN}-->
<!--{newpinmsg}-->
<!--{else}-->
<!--{errmsg}-->
<!--{xif}-->
</EM></FONT><BR><BR><BR>
<!--{xif}-->
<!--{ifrequestmoney}-->
<FONT SIZE=+1 COLOR=RED>
<!--{requestmoney}-->
</FONT><BR><BR>
<!--{xif}-->
<!--{enterinfo}-->
</p>

<table width="100%" border="0" cellspacing="0" cellpadding="2">
<tr>
<td valign="top" width="1%">

<TABLE>
<!--{ifneedpatronname}-->
<TR><TD WIDTH=1%></TD><TD class="blurb">
<!--{nameexample}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{nameprompt}-->
</TD><TD>
<!--{name}-->
</TD></TR>
<!--{xif}-->
<!--{ifneedpatronverify}-->
<tr><td colspan="3"> </td></tr>
<TR><TD width="1%"></TD><TD class="blurb">
<!--{barcodeexample}-->
</TD></TR>
<TR><TD class="blurb" width="1%" nowrap>
<!--{barcodeprompt}-->
</TD><TD>
<!--{barcode}-->
</TD></TR>
<!--{ifneedspin}-->
<!--{iferrmsgisPIN}-->
<TR><TD COLSPAN=2>
<!--{newpinmsg}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{pin1prompt}-->
</TD><TD>
<!--{pin1}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{pin2prompt}-->
</TD><TD>
<!--{pin2}-->
</TD></TR>
<!--{else}-->
<TR><TD></TD><TD class="blurb">
<!--{pininstructions}-->
</TD></TR><TR><TD class="blurb" width="1%" nowrap>
<!--{pinprompt}-->
</TD><TD>
<!--{pin}-->
</TD></TR>
<!--{xif}-->
<!--{xif}-->
<!--{xif}-->
</TABLE>
<P>
<!--{ifcampus}-->
```

```
<!--{campus}-->
<!--{else}-->
<!--{holdshelfmenu}-->
<!--{xif}-->
</p>

<!--{ifreqinst}-->
<P>
<!--{reqinst}-->
<BR>
<!--{inst}-->
<P>
<!--{xif}-->
<!--{ifillreq}-->
<P>
<!--{illreq}-->
</P>
<!--{xif}-->
<!--{ifneedNNA}-->
<P>
<!--{needbyprompt}-->
<!--{needby}-->
<!-- javascript validation code -->
<script language="JavaScript">
<!--

// utility functions for date calculations

function isLeapYear(intYear)
{
    if (intYear % 100 == 0)
    {
        if (intYear % 400 == 0) { return true; }
    }
    else
    {
        if ((intYear % 4) == 0) { return true; }
    }
    return false;
}

function isDayValidForThisMonthAndYear(intDay,intMonth,intYear)
{
    if ((intMonth == 4 || intMonth == 6 || intMonth == 9 || intMonth == 11) && intDay > 30 )
    {
        return false;
    }

    if (intMonth == 2)
    {
        if (isLeapYear(intYear))
        {
            if (intDay > 29)
                {
                return false;
                }
        }
        else
        {
            if (intDay > 28)
            {
                return false;
            }
        }
    }
    return true;
}

// form validation function

function formCheck(form)
```

```
{

////////////////////////////////////////////////////////////////////////////////////
// broswer sniffer
////////////////////////////////////////////////////////////////////////////////////

// Ultimate client-side JavaScript client sniff. Version 3.02
// (C) Netscape Communications 1999-2001.  Permission granted to reuse and distribute.
// Revised 17 May 99 to add is_nav5up and is_ie5up (see below).
// Revised 20 Dec 00 to add is_gecko and change is_nav5up to is_nav6up
//                       also added support for IE5.5 Opera4&5 HotJava3 AOLTV
// Revised 22 Feb 01 to correct Javascript Detection for IE 5.x, Opera 4,
//                       correct Opera 5 detection
//                       add support for winME and win2k
//                       synch with browser-type-oo.js
// Revised 26 Mar 01 to correct Opera detection

// Everything you always wanted to know about your JavaScript client
// but were afraid to ask. Creates "is_" variables indicating:
// (1) browser vendor:
//     is_nav, is_ie, is_opera, is_hotjava, is_webtv, is_TVNavigator, is_AOLTV
// (2) browser version number:
//     is_major (integer indicating major version number: 2, 3, 4 ...)
//     is_minor (float   indicating full  version number: 2.02, 3.01, 4.04 ...)
// (3) browser vendor AND major version number
//     is_nav2, is_nav3, is_nav4, is_nav4up, is_nav6, is_nav6up, is_gecko, is_ie3,
//     is_ie4, is_ie4up, is_ie5, is_ie5up, is_ie5_5, is_ie5_5up, is_hotjava3, is_hotjava3up,
//     is_opera2, is_opera3, is_opera4, is_opera5, is_opera5up
// (4) JavaScript version number:
//     is_js (float indicating full JavaScript version number: 1, 1.1, 1.2 ...)
// (5) OS platform and version:
//     is_win, is_win16, is_win32, is_win31, is_win95, is_winnt, is_win98, is_winme, is_win2k
//     is_os2
//     is_mac, is_mac68k, is_macppc
//     is_unix
//     is_sun, is_sun4, is_sun5, is_suni86
//     is_irix, is_irix5, is_irix6
//     is_hpux, is_hpux9, is_hpux10
//     is_aix, is_aix1, is_aix2, is_aix3, is_aix4
//     is_linux, is_sco, is_unixware, is_mpras, is_reliant
//     is_dec, is_sinix, is_freebsd, is_bsd
//     is_vms
//
// See http://www.it97.de/JavaScript/JS_tutorial/bstat/navobj.html and
// http://www.it97.de/JavaScript/JS_tutorial/bstat/Browseraol.html
// for detailed lists of userAgent strings.
//
// Note: you don't want your Nav4 or IE4 code to "turn off" or
// stop working when new versions of browsers are released, so
// in conditional code forks, use is_ie5up ("IE 5.0 or greater")
// is_opera5up ("Opera 5.0 or greater") instead of is_ie5 or is_opera5
// to check version in code which you want to work on future
// versions.

    // convert all characters to lowercase to simplify testing
    var agt=navigator.userAgent.toLowerCase();

    // *** BROWSER VERSION ***
    // Note: On IE5, these return 4, so use is_ie5up to detect IE5.
    var is_major = parseInt(navigator.appVersion);
    var is_minor = parseFloat(navigator.appVersion);

    // Note: Opera and WebTV spoof Navigator.  We do strict client detection.
    // If you want to allow spoofing, take out the tests for opera and webtv.
    var is_nav  = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('spoofer')==-1)
                && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
                && (agt.indexOf('webtv')==-1) && (agt.indexOf('hotjava')==-1));
    var is_nav2 = (is_nav && (is_major == 2));
    var is_nav3 = (is_nav && (is_major == 3));
    var is_nav4 = (is_nav && (is_major == 4));
    var is_nav4up = (is_nav && (is_major >= 4));
```

```
214     var is_navonly      = (is_nav && ((agt.indexOf(";nav") != -1) ||
215                           (agt.indexOf("; nav") != -1)) );
216     var is_nav6 = (is_nav && (is_major == 5));
217     var is_nav6up = (is_nav && (is_major >= 5));
218     var is_gecko = (agt.indexOf('gecko') != -1);
219
220
221     var is_ie     = ((agt.indexOf("msie") != -1) && (agt.indexOf("opera") == -1));
222     var is_ie3    = (is_ie && (is_major < 4));
223     var is_ie4    = (is_ie && (is_major == 4) && (agt.indexOf("msie 5")==-1) );
224     var is_ie4up  = (is_ie && (is_major >= 4));
225     var is_ie5    = (is_ie && (is_major == 4) && (agt.indexOf("msie 5.0")!=-1) );
226     var is_ie5_5  = (is_ie && (is_major == 4) && (agt.indexOf("msie 5.5") !=-1));
227     var is_ie5up  = (is_ie && !is_ie3 && !is_ie4);
228     var is_ie5_5up =(is_ie && !is_ie3 && !is_ie4 && !is_ie5);
229
230     // KNOWN BUG: On AOL4, returns false if IE3 is embedded browser
231     // or if this is the first browser window opened.  Thus the
232     // variables is_aol, is_aol3, and is_aol4 aren't 100% reliable.
233     var is_aol   = (agt.indexOf("aol") != -1);
234     var is_aol3  = (is_aol && is_ie3);
235     var is_aol4  = (is_aol && is_ie4);
236     var is_aol5  = (agt.indexOf("aol 5") != -1);
237     var is_aol6  = (agt.indexOf("aol 6") != -1);
238
239     var is_opera = (agt.indexOf("opera") != -1);
240     var is_opera2 = (agt.indexOf("opera 2") != -1 || agt.indexOf("opera/2") != -1);
241     var is_opera3 = (agt.indexOf("opera 3") != -1 || agt.indexOf("opera/3") != -1);
242     var is_opera4 = (agt.indexOf("opera 4") != -1 || agt.indexOf("opera/4") != -1);
243     var is_opera5 = (agt.indexOf("opera 5") != -1 || agt.indexOf("opera/5") != -1);
244     var is_opera5up = (is_opera && !is_opera2 && !is_opera3 && !is_opera4);
245
246     var is_webtv = (agt.indexOf("webtv") != -1);
247
248     var is_TVNavigator = ((agt.indexOf("navio") != -1) || (agt.indexOf("navio_aoltv") !=
    -1));
249     var is_AOLTV = is_TVNavigator;
250
251     var is_hotjava = (agt.indexOf("hotjava") != -1);
252     var is_hotjava3 = (is_hotjava && (is_major == 3));
253     var is_hotjava3up = (is_hotjava && (is_major >= 3));
254
255     // *** JAVASCRIPT VERSION CHECK ***
256     var is_js;
257     if (is_nav2 || is_ie3) is_js = 1.0;
258     else if (is_nav3) is_js = 1.1;
259     else if (is_opera5up) is_js = 1.3;
260     else if (is_opera) is_js = 1.1;
261     else if ((is_nav4 && (is_minor <= 4.05)) || is_ie4) is_js = 1.2;
262     else if ((is_nav4 && (is_minor > 4.05)) || is_ie5) is_js = 1.3;
263     else if (is_hotjava3up) is_js = 1.4;
264     else if (is_nav6 || is_gecko) is_js = 1.5;
265     // NOTE: In the future, update this code when newer versions of JS
266     // are released. For now, we try to provide some upward compatibility
267     // so that future versions of Nav and IE will show they are at
268     // *least* JS 1.x capable. Always check for JS version compatibility
269     // with > or >=.
270     else if (is_nav6up) is_js = 1.5;
271     // NOTE: ie5up on mac is 1.4
272     else if (is_ie5up) is_js = 1.3
273
274     // HACK: no idea for other browsers; always check for JS version with > or >=
275     else is_js = 0.0;
276
277     // *** PLATFORM ***
278     var is_win   = ( (agt.indexOf("win")!=-1) || (agt.indexOf("16bit")!=-1) );
279     // NOTE: On Opera 3.0, the userAgent string includes "Windows 95/NT4" on all
280     //        Win32, so you can't distinguish between Win95 and WinNT.
281     var is_win95 = ((agt.indexOf("win95")!=-1) || (agt.indexOf("windows 95")!=-1));
282
283     // is this a 16 bit compiled version?
```

```
284     var is_win16 = ((agt.indexOf("win16")!=-1) ||
285                (agt.indexOf("16bit")!=-1) || (agt.indexOf("windows 3.1")!=-1) ||
286                (agt.indexOf("windows 16-bit")!=-1) );
287
288     var is_win31 = ((agt.indexOf("windows 3.1")!=-1) || (agt.indexOf("win16")!=-1) ||
289                     (agt.indexOf("windows 16-bit")!=-1));
290
291     var is_winme = ((agt.indexOf("win 9x 4.90")!=-1));
292     var is_win2k = ((agt.indexOf("windows nt 5.0")!=-1));
293
294     // NOTE: Reliable detection of Win98 may not be possible. It appears that:
295     //       - On Nav 4.x and before you'll get plain "Windows" in userAgent.
296     //       - On Mercury client, the 32-bit version will return "Win98", but
297     //         the 16-bit version running on Win98 will still return "Win95".
298     var is_win98 = ((agt.indexOf("win98")!=-1) || (agt.indexOf("windows 98")!=-1));
299     var is_winnt = ((agt.indexOf("winnt")!=-1) || (agt.indexOf("windows nt")!=-1));
300     var is_win32 = (is_win95 || is_winnt || is_win98 ||
301                     ((is_major >= 4) && (navigator.platform == "Win32")) ||
302                     (agt.indexOf("win32")!=-1) || (agt.indexOf("32bit")!=-1));
303
304     var is_os2   = ((agt.indexOf("os/2")!=-1) ||
305                     (navigator.appVersion.indexOf("OS/2")!=-1) ||
306                     (agt.indexOf("ibm-webexplorer")!=-1));
307
308     var is_mac    = (agt.indexOf("mac")!=-1);
309     // hack ie5 js version for mac
310     if (is_mac && is_ie5up) is_js = 1.4;
311     var is_mac68k = (is_mac && ((agt.indexOf("68k")!=-1) ||
312                                 (agt.indexOf("68000")!=-1)));
313     var is_macppc = (is_mac && ((agt.indexOf("ppc")!=-1) ||
314                                  (agt.indexOf("powerpc")!=-1)));
315
316     var is_sun   = (agt.indexOf("sunos")!=-1);
317     var is_sun4  = (agt.indexOf("sunos 4")!=-1);
318     var is_sun5  = (agt.indexOf("sunos 5")!=-1);
319     var is_suni86= (is_sun && (agt.indexOf("i86")!=-1));
320     var is_irix  = (agt.indexOf("irix") !=-1);    // SGI
321     var is_irix5 = (agt.indexOf("irix 5") !=-1);
322     var is_irix6 = ((agt.indexOf("irix 6") !=-1) || (agt.indexOf("irix6") !=-1));
323     var is_hpux  = (agt.indexOf("hp-ux")!=-1);
324     var is_hpux9 = (is_hpux && (agt.indexOf("09.")!=-1));
325     var is_hpux10= (is_hpux && (agt.indexOf("10.")!=-1));
326     var is_aix   = (agt.indexOf("aix") !=-1);      // IBM
327     var is_aix1  = (agt.indexOf("aix 1") !=-1);
328     var is_aix2  = (agt.indexOf("aix 2") !=-1);
329     var is_aix3  = (agt.indexOf("aix 3") !=-1);
330     var is_aix4  = (agt.indexOf("aix 4") !=-1);
331     var is_linux = (agt.indexOf("inux")!=-1);
332     var is_sco   = (agt.indexOf("sco")!=-1) || (agt.indexOf("unix_sv")!=-1);
333     var is_unixware = (agt.indexOf("unix_system_v")!=-1);
334     var is_mpras    = (agt.indexOf("ncr")!=-1);
335     var is_reliant  = (agt.indexOf("reliantunix")!=-1);
336     var is_dec   = ((agt.indexOf("dec")!=-1) || (agt.indexOf("osf1")!=-1) ||
337            (agt.indexOf("dec_alpha")!=-1) || (agt.indexOf("alphaserver")!=-1) ||
338            (agt.indexOf("ultrix")!=-1) || (agt.indexOf("alphastation")!=-1));
339     var is_sinix = (agt.indexOf("sinix")!=-1);
340     var is_freebsd = (agt.indexOf("freebsd")!=-1);
341     var is_bsd = (agt.indexOf("bsd")!=-1);
342     var is_unix  = ((agt.indexOf("x11")!=-1) || is_sun || is_irix || is_hpux ||
343                  is_sco ||is_unixware || is_mpras || is_reliant ||
344                  is_dec || is_sinix || is_aix || is_linux || is_bsd || is_freebsd);
345
346     var is_vms   = ((agt.indexOf("vax")!=-1) || (agt.indexOf("openvms")!=-1));
347
348 ////////////////////////////////////////////////////////////////////////////
349 // begin form validation
350 ////////////////////////////////////////////////////////////////////////////
351
352     // if this client is running an unusable version of JavaScript, skip this validation
353     if (is_js < 1.1)
354     {
```

```
355         return true;
356     }
357
358     // read in values from the form
359     var strSuppliedYear = form.needby_Year.options[form.needby_Year.selectedIndex].text;
360     var strSuppliedMonth = form.needby_Month.options[form.needby_Month.selectedIndex].text;
361     var strSuppliedDay = form.needby_Day.options[form.needby_Day.selectedIndex].text;
362
363     // convert supplied date strings to IETF date format
364     var strSuppliedDate = "" + strSuppliedDay + " " + strSuppliedMonth + " " +
        strSuppliedYear + " 00:00:00 CST";
365
366     // get today's date in GMTmilliseconds format
367     var Today = new Date();
368
369     // convert supplied date to GMTmilliseconds format
370     var SuppliedDate = new Date(strSuppliedDate);
371
372     // if the resulting date object is not a number, the user hasn't picked a date. skip
        validation.
373     if (isNaN(SuppliedDate))
374     {
375         return true;
376     }
377
378     // Compare Today to SuppliedDate
379     // If SuppliedDate is less than Today, alert the user, and return false
380     if (SuppliedDate <= Today)
381     {
382
383         alert("INVALID CANCEL BY DATE: please supply a 'cancel by' date that occurs in the
            future.");
384         return false;
385     }
386
387     // check to see if the supplied day value is valid for the supplied month and year
388     if (!isDayValidForThisMonthAndYear(parseInt(strSuppliedDay),
        form.needby_Month.selectedIndex, parseInt(strSuppliedYear)))
389     {
390         alert("INVALID CANCEL BY DATE: are you sure there are " + strSuppliedDay + " days in
            " + strSuppliedMonth + "?");
391         return false;
392     }
393
394     // if we made it this far, the date must be good
395     return true;
396
397 }
398
399 //-->;
400 </script>
401
402 </p>
403
404 <!--{else}-->
405 <!-- passthrough javascript code -->
406 <script language="JavaScript">
407 <!--
408 function formCheck(form) {
409     return true;
410 }
411 //-->;
412 </script>
413 <!--{xif}-->
414 <br>
415 <!--{submit}-->
416 </td>
417 <td valign="top" align="right">
418 <table width="200" border="0" cellspacing="0" cellpadding="2" bgcolor="#CDC8B1">
419   <tr>
420     <td>
```

```
421         <p><b><font size="-1" face="Arial, Helvetica, sans-serif">Campus ID Help</font></b></p>
422         <p class="blurb"><font size="-2" face="Arial, Helvetica, sans-serif">
423           &middot; Central Methodist College - <b>CMC</b><br>
424           &middot; Central Missouri State University - <b>CMSU</b><br>
425           &middot; Missouri Valley College - <b>MVC</b><br>
426           &middot; State Fair Community College - <b>SFCC</b></font></p>
427 
428         <p class="blurb"><font size="-2" face="Arial, Helvetica, sans-serif">
429           Type your ID number (dashes are not necessary) plus your institution
        code.
430         <br>    For example: <tt><font size="-1"
          color="#CC0000">999999999MVC</font></tt> would be a Campus
431           ID for a student at Missouri Valley College.<br><br></font></p>
432         </td>
433     </tr>
434 </table>
435 
436 </td>
437 </tr>
438 </table>
439 
440 <P>
441 
442 </FORM>
443 <TABLE WIDTH=100%>
444 <TR><TD>
445 <!--{startover}-->
446 </TD></TR>
447 </TABLE>
448 
449 <!--{botlogo}-->
450 
451 </BODY>
452 </HTML>
```

```
<!--{toplogo}-->
<FORM METHOD=POST onSubmit="return formCheck(this)">
<!--{iferrmsg}-->
<FONT COLOR=RED SIZE=+2><EM>
<!--{iferrmsgisPIN}-->
<!--{newpinmsg}-->
<!--{else}-->
<!--{errmsg}-->
<!--{xif}-->
</EM></FONT><BR><BR><BR>
<!--{xif}-->
<!--{enterinfo}-->
<table width="100%" border="0" cellspacing="0" cellpadding="2">

<tr>
<td valign="top">

<TABLE WIDTH=100%>
<!--{ifneedpatronname}-->
<TR><TD WIDTH=30%></TD><TD><FONT SIZE=-1>
<!--{nameexample}-->
</FONT></TD><TD></TD></TR><TR><TD align="right"><FONT SIZE=-1>
Your Name
</FONT></TD><TD>
<!--{name}-->
</TD><TD></TD></TR>
<tr><td colspan="3"><img src="/screens/shim.gif" width=1 height=10></td></tr>
<!--{xif}-->
<!--{ifneedpatronverify}-->
<TR valign="bottom"><TD></TD><TD><FONT SIZE=-1>
For example: &quot;123456789XX&quot;
</FONT></TD>
<TD rowspan=2>
<table border="0" cellspacing="0" cellpadding="2">
  <tr bgcolor="#99CCFF"><td nowrap><FONT SIZE=-1>EC = East Central College</FONT></TD><TR>
  <tr bgcolor="#99CCFF"><td nowrap><FONT SIZE=-1>JC = Jefferson College</FONT></TD><TR>
  <tr bgcolor="#99CCFF"><td nowrap><FONT SIZE=-1>SC = St. Charles Community
  College</FONT></TD><TR>
  <tr bgcolor="#99CCFF"><td nowrap><FONT SIZE=-1>ST = St. Louis Community
  College</FONT></TD><TR>
</table>
</TD>
</TR>
<TR valign="top"><TD align="right"><FONT SIZE=-1>
Your Social Security Number or equivalent plus your two letter College Code: EC, JC, SC, ST
</FONT></TD><TD>
<!--{barcode}-->
</TD></TR>
<tr><td colspan="3"><img src="/screens/shim.gif" width=1 height=10></td></tr>
<!--{ifneedspin}-->
<!--{iferrmsgisPIN}-->
<TR><TD COLSPAN=2>
<!-- newpinmsg token-->
</TD></TR><TR><TD align="right"><FONT SIZE=-1>
Enter Your New PIN
</FONT></TD><TD>
<!--{pin1}-->
</TD><TD></TD></TR>
<tr><td colspan="3"><img src="/screens/shim.gif" width=1 height=10></td></tr>
<TR><TD align="right"><FONT SIZE=-1>
Enter Your New PIN Again
</FONT></TD><TD>
<!--{pin2}-->
<script language="JavaScript">
<!--

////////////////////////////////////////////////////////////////////////////////
// begin form validation
////////////////////////////////////////////////////////////////////////////////

function formCheck(form)
```

```
{

    // read in value from the form
    var strPIN = form.pin1.value.toString();

    // if strPIN is less than four characters, it is invalid
    if (strPIN.length < 4)
    {
        alert("We're sorry, a PIN must be a minimum of 4 alphanumeric characters. Please
        input a longer PIN.");
        return false;
    }

    // if we made it this far, the pin must be good
    return true;

}

//-->;
</script>

</TD></TR>
<!--{else}-->
<TR><TD>

<!-- passthrough javascript code -->
<script language="JavaScript">
<!--
function formCheck(form) {
    return true;
}
//-->;
</script>

</TD><TD><FONT SIZE=-1>
<!-- pininstructions token -->
</FONT></TD></TR><TR><TD align="right"><FONT SIZE=-1>
<!--{pinprompt}-->
</FONT></TD><TD>
<!--{pin}-->
</TD></TR>
<!--{xif}-->
<!--{xif}-->
<!--{xif}-->
<tr><td colspan="2">
<br>
<br>
<!--{submit}-->
</td></tr>
</TABLE>

</td>
</tr>
</table>

<P>

</FORM>
<TABLE WIDTH=100%>
<TR><TD>
<!--{startover}-->
</TD></TR>
</TABLE>

<!--{botlogo}-->

</BODY>
</HTML>
```

```
<html>

<head>
<title>MERLIN Keyword Search</title>

<SCRIPT LANGUAGE="JavaScript">

<!--
////////////////////////////////////////////////////////////////////
// isEmpty function from JavaScript Bible

function isEmpty(inputStr) {
    if (inputStr == null || inputStr == "") {
        return true;
    }
    return false;
}

/*
MOBIUS srchhelp_x.html form usability enhancement functions
Author: Hardy Pottinger
Author Email: pottingerhj@umystem.edu
*/

function checkDates() {
    var field = document.frmAdvancedKeyword;
    if (isEmpty(field.Db.value) && ! isEmpty(field.Da.value)) {
        // add a before date, so relevancy ranking will work correctly
        field.Db.value = 9999;
    }
    if (isEmpty(field.Da.value) && ! isEmpty(field.Db.value)) {
        // add an after date, so relevancy ranking will work correctly
        field.Da.value = 1000;
    }
    return true;
}
// -->

</SCRIPT>

</head>

<body BGCOLOR="#FFFFF0" onLoad="document.forms[0].SEARCH.focus();
document.forms[0].SEARCH.select();">

<table border="0" width="100%" height="113">
  <tr>
    <td width="29%" height="107" valign="top" align="center"><a
    href="http://merlin.missouri.edu"><img
    src="/screens/merlinlogo_b.gif" alt="Merlin Web Gateway"
    WIDTH="180" HEIGHT="83 " border="0"> </a></td>
    <td width="71%" height="107" valign="middle" align="center"><b>Missouri Education and
    Research Libraries Information Network</b><br>
    <b>University of Missouri Campuses and Saint Louis University</b></td>
  </tr>
</table>

<center>

<h3>Keyword Search - All Collections</h3>


<!--{msg}-->

    <FORM NAME="frmSimpleKeyword" METHOD=GET>
        <TABLE BORDER=0 CELLPADDING=2 width="570">
            <TR>
                <TD align=top><B>Type the <b>WORD(S)</b> you want, then press
                    <b>&lt;Enter&gt;</b> or click <b>Submit Search</b>. For more
                    options, go to<b> <a href=#AVS>ADVANCED SEARCH</a></b>. </b></td>
            <tr align="center">
```

```
                <TD>

<!--{search}-->

                    <br>
                    <INPUT TYPE=SUBMIT VALUE="SIMPLE SEARCH">
                </TD>
            </tr>
        </TABLE>
        <INPUT TYPE=HIDDEN NAME=SORT VALUE=D>
    </FORM>
    <p><a HREF="/search/">Return to Main Menu</a></p>
    <TABLE CELLPADDING=3 CELLSPACING=2 WIDTH="100%">
        <TR>
            <TH WIDTH=10%> </TH>
            <TH WIDTH=40%>Type in Words to search: </TH>
            <TH WIDTH=50%>EXAMPLES:</TH>
        </TR>
        <TR>
            <TD WIDTH=100><b>ADJACENCY</b></TD>
            <TD BGCOLOR="#CCFFCC"> The order of words matters. Multiple words
                are searched together as one phrase. For different results, use
                <i><b>and</b></i> between each word or phrase.</TD>
            <TD BGCOLOR="#99CCFF"> water resources<br>
                colorado<b> and</b> geology<br>
                electronic serial<b> and</b> physics</TD>
        <TR>
            <TD WIDTH=100><b>TRUNCATION</b></TD>
            <TD BGCOLOR="#CCFFCC"> Words may be internally or right-hand truncated
                using an asterisk *. Use double asterisks ** for more than 6 characters.
            </TD>
            <TD BGCOLOR="#99CCFF"> wom*n and librar*<br>
                math**</TD>
        <TR>
            <TD WIDTH=100><b>OPERATORS</b></TD>
            <TD BGCOLOR="#CCFFCC"> Use <i><b>and</b></i> or <i><b>or</b></i> to
                specify multiple words in any field, any order. Use <i><b>and
                not</b></i> to exclude words. Use parentheses to group words when
                using multiple operators. </TD>
            <TD BGCOLOR="#99CCFF"> cat <b>or</b> feline <br>
                (alaska <b>or</b> canada)<b> and</b> (recreation<b> and not</b>
                hiking) </TD>
        <TR>
            <TD WIDTH=100><b>PROXIMITY</b></TD>
            <TD BGCOLOR="#CCFFCC">Use <i><b>near</b></i> to specify words close
                to each other, in any order. Use <i><b>within #</b></i> to specify
                words within a set range of each other. Symbol # may stand for
                any number. </TD>
            <TD BGCOLOR="#99CCFF"> California <b>near</b> university<br>
                colorado<b> within 12 </b>wyoming</TD>
        <TR>
            <TD WIDTH=100><b>FIELDS</b></TD>
            <TD BGCOLOR="#CCFFCC"> Specify fields to search, using field abbreviation.
                Fields available for this database are <b>a:</b> (author),<b>
                t:</b> (title),<b> s:</b> (subject), <b> c:</b> (table of contents), and<b>
                n:</b> (additional
                keywords.) Subjects are NOT cross-referenced. </TD>
            <TD BGCOLOR="#99CCFF"><b> a:</b>twain <b>and t:</b>huck<br>
                (<b>a:</b>frost <b> or a:</b>whitman) <b>and s:</b>poet* <br>
                <b>t:</b>medicine <b>and (n:</b>research<b> or s:</b>information)</TD>
        </TR>
    </TABLE>

<!-- begin advanced keyword search form -->

<FORM NAME="frmAdvancedKeyword" METHOD=GET ONSUBMIT="return checkDates();">
        <TABLE BORDER=0 CELLPADDING=3 CELLSPACING=0 ALIGN="center" BGCOLOR="#99CCFF">
            <TR>
    <TD ALIGN=center colspan=4> <B><a name=AVS></a>For a more Advanced
        Search, type WORD(S), select OPTIONS <br> and click on ADVANCED
        SEARCH: </B> <br>
```

```
141
142 <!--{msg}-->
143
144 <br>
145
146 <!--{search}-->
147
148     </TD>
149 </TR>
150
151 <TR>
152     <TD colspan=3 align=right>
153 <!--{lang}-->
154     </TD>
155     <TD align=center>
156 <!--{sort}-->
157     </TD>
158 </TR>
159
160 <TR>
161     <TD align=right>
162 <!--{serbk}-->
163     </TD>
164     <TD colspan=2 align=center>
165         or
166     </TD>
167     <TD>
168 <!--{mattype}-->
169     </TD>
170 </TR>
171
172 <TR>
173     <TD align="center" colspan=2>
174 Scope:
175 <!--{scope}-->
176     </TD>
177     <TD colspan=2>
178 <!--{branch}-->
179     </TD>
180 </TR>
181
182 <TR>
183     <TD ALIGN=center colspan=2>
184 <!--{publish}-->
185     </TD>
186     <TD ALIGN=center colspan=2>
187 <!--{genre}-->
188     </TD>
189 </TR>
190
191 <TR>
192     <TD ALIGN=center colspan=4>
193 <!--{DOCDATE}-->
194     </TD>
195 </TR>
196
197 <TR>
198     <TD ALIGN=center colspan=4>
199         <INPUT TYPE=SUBMIT VALUE="ADVANCED SEARCH">
200     </TD>
201 </TR>
202
203 </TABLE>
204 </FORM>
205
206 <!-- end advanced keyword search form -->
207
208 <p><a HREF="/search/">Return to Main Menu</a></p>
209
210 </center>
211 </body>
```

```
212  </html>
```

```
<HTML>
<HEAD>
    <TITLE>Bridges/ Bridges Catalog</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta http-equiv="Refresh" CONTENT="300;URL=http://bridges.missouri.edu/search----->">
</HEAD>
<BODY bgcolor="#ffffff" link="#0000FF" alink="#0000FF" vlink="#0000FF">
    <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr valign="top">
        <td rowspan="2" valign="bottom"><a href="http://bridges.missouri.edu/"><IMG
        SRC="http://207.160.138.30/~hardy/screens/bridges_navbar.gif" HEIGHT="45" WIDTH="150"
        ALT=" Bridges home " BORDER="0" ></a></td>
        <td align="right" width="90%"> </td>
        <td align="right" nowrap>

            <font color="#000000" face="Georgia, Times New Roman, Times, serif"><b>Bridges
            Library Catalog</b></font>

        </td>
      </tr>
      <tr>
        <td align="right" valign="bottom"> </td>
        <td align="right" valign="middle" nowrap><font color="#000000" face="Verdana, Arial,
        Helvetica, sans-serif" size="-2"><b>
          <a href="http://mobius.missouri.edu/search/">Search MOBIUS</a> |
          <a href="/search~/">Search Bridges</a> |
          <a href="/">Bridges Home</a> |
          <a href="/screens/help.html">Help</a>
          </b></font></td>
      </tr>
      <tr>
        <td colspan="3" height="1" bgcolor="#DEDDC9"><img src="/screens/shim.gif" width="1"
        height="2" border="0"></td>
      </tr>
    </table>
<br>
<!-- Begin Search Help Text -->
<!--{patron}-->
<center>
<h3 align="center"><b>Keyword Search</b></h3>
<FONT COLOR=RED SIZE=+2><b>
<!--{msg}-->
</b></FONT>
<form NAME="frmAdvancedKeyword" METHOD=GET ONSUBMIT="return setDaAndDb();">
<TABLE BORDER=0 CELLPADDING=4 CELLSPACING=2>
<TR>
<TD BGCOLOR="#99CCFF" colspan=2>
<CENTER><b>Search: </b>
<!--{search}-->
<INPUT TYPE=submit VALUE=Search>
<BR>
<center>Example: <tt>civil rights and (congress or supreme court)</tt><br>
See <a href="#tips">Search tips</a> below for further instructions.
</center>
</TR>
<TR BGCOLOR="#99CCFF">
<TD BGCOLOR="#99CCFF">
<B>Sort Results</B>
</TD>
<TD>
<input type=radio name=SORT VALUE=A CHECKED>By Title  
<input type=radio name=SORT VALUE=D>By Date  
<input type=radio name=SORT VALUE=R>By Relevance  
</TD>
</TR>
<TR BGCOLOR="#99CCFF">
<TD BGCOLOR="#99CCFF">
<b>Language of Work</b>
</TD>
<TD>
<select name=l>
```

```
<option value="">ANY
<option value="eng">English
<option value="chi">Chinese
<option value="dut">Dutch
<option value="fre">French
<option value="ger">German
<option value="gre">Greek
<option value="heb">Hebrew
<option value="hun">Hungarian
<option value="ita">Italian
<option value="jpn">Japanese
<option value="kor">Korean
<option value="lat">Latin
<option value="pol">Polish
<option value="por">Portuguese
<option value="rus">Russian
<option value="spa">Spanish
</select>
</TD>
</TR>
<TR>
<TD BGCOLOR="#99CCFF"><b>Year of Publication</b></TD>
<TD BGCOLOR="#99CCFF">
<SCRIPT LANGUAGE="JavaScript">
<!--
// single year text box for JavaScript-capable broswers
document.write("Single Year: <input type=text size=4 maxlength=4 name=Sy value=''
onChange='resetDaAndDBIfNecessary();'>   OR   ");
// -->
</SCRIPT>
Specify Range: after <input type=text size=4 name=Da maxlength=4 value=""
onChange="resetSyIfNecessary();">
before <input type=text size=4 maxlength=4 name=Db value="" onChange="resetSyIfNecessary();">
</TD>
</TR>
<TR>
<TD BGCOLOR="#99CCFF"><b>Material format</b></TD>
<td BGCOLOR="#99CCFF">
<input type=radio name=s VALUE="m">Book   
<input type=radio name=s VALUE="s">Serial  
<input type=radio name=s VALUE="">
Other:
<select name=m onchange="checkOtherIfNecessary();">
<option value="">ANY
<option value="t">THESIS/MANUSCRIPT
<option value="2">E-BOOK
<option value="m">ELECTRONIC RESOURCE
<option value="g">VIDEO
<option value="i">AUDIO, SPOKEN
<option value="j">AUDIO, MUSIC
<option value="c">MUSIC SCORE
<option value="e">MAP OR ATLAS
<option value="k">CHART/PICTURE
<option value="b">ARCHIVAL MATL
<option value="o">KIT
<option value="r">3D OBJECT/GAME
</select>
</td>
</TR>
<TR>
<TD BGCOLOR="#99CCFF"><b>Publisher Name Keywords</b></TD>
<td BGCOLOR="#99CCFF"><input type=text size=40 name=p value=>
</TD>
</TR>
<TR>
<TD BGCOLOR="#99CCFF"><b>Specific Branch</b>
<br>
OR<br>
<b>Specific Library Location</b></TD>
<td BGCOLOR="#99CCFF">
<select name=searchscope >
```

```
136             <option value="10" selected>Bridges Catalog</option>           <option value="1"
                >Covenant Seminary</option>           <option value="2" >Fontbonne
                University</option>           <option value="3" >Harris-Stowe State
                College</option>           <option value="4" >Kenrick-Glennon Seminary</option>
                          <option value="5" >Lindenwood University</option>           <option
                value="6" >Logan College of Chiropractic</option>           <option value="7"
                >Maryville University</option>           <option value="8" >Missouri Baptist
                College</option>           <option value="9" >Webster Univ./Eden
                Seminary</option>
137 </select>
138 <br><br>
139 <select name=b>
140 <option value="" selected>All Bridges Locations
141 <option value="cbb">Covenant Seminary Library
142 <option value="web">Eden-Webster Library
143 <option value="fcb">Fontbonne College Library
144 <option value="hsb">Harris-Stowe State College Library
145 <option value="kgb">Kenrick-Glennon Seminary Library
146 <option value="lbb">Lindenwood University Library
147 <option value="ojb">Logan College of Chiropractic Library
148 <option value="mub">Maryville University Library
149 <option value="bjb">Missouri Baptist College Library
150 <option value="bsb">Missouri Baptist College St. Clair
151 <option value="btb">Missouri Baptist College Troy/Wentzville
152 </select>
153 </td>
154 </TR>
155 <TR>
156 <TH colspan=2 bgcolor="#99CCFF">
157 <INPUT TYPE=submit VALUE=Search>
158 <INPUT TYPE=reset VALUE=Clear>
159 </TH>
160 </TR>
161 </TABLE>
162 </FORM>
163 <SCRIPT LANGUAGE="JavaScript">
164 <!--
165 ////////////////////////////////////////////////////////////////////
166 // Utility Functions from JavaScript Bible
167 function isPosInteger(inputVal) {
168     var inputStr = inputVal.toString();
169     for (i = 0; i < inputStr.length; i++) {
170         var oneChar = inputStr.charAt(i);
171         if (oneChar < "0" || oneChar > "9") {
172             return false;
173         }
174     }
175     return true;
176 }
177 function isEmpty(inputStr) {
178     if (inputStr == null || inputStr == "") {
179         return true;
180     }
181     return false;
182 }
183 /*
184 MOBIUS srchhelp_x.html form usability enhancement functions
185 Author: Hardy Pottinger
186 Author Email: pottingerhj@umystem.edu
187 */
188 function checkOtherIfNecessary() {
189     var field = document.forms[0];
190     if (field.m.selectedIndex != 0) {
191         field.s[2].checked = true;
192     }
193 }
194 function setDaAndDb() {
195     var field = document.forms[0];
196     if (! isEmpty(field.Sy.value)) {
197         if (isPosInteger(field.Sy.value)) {
198             field.Da.value = field.Sy.value - 1;
```

```
            field.Db.value = field.Sy.value - 1 + 2;
        } else {
            alert("Sorry, '" + field.Sy.value + "' is not a valid entry for a single year.");
            field.Sy.focus();
            field.Sy.select();
            return false;
        }
    }
    if (isEmpty(field.Db.value) && ! isEmpty(field.Da.value)) {
        // add a before date, so relevancy ranking will work correctly
        field.Db.value = 9999;
    }
    if (isEmpty(field.Da.value) && ! isEmpty(field.Db.value)) {
        // add an after date, so relevancy ranking will work correctly
        field.Da.value = 1000;
    }
return true;
}
function resetSyIfNecessary() {
    var field = document.forms[0];
    if (! isEmpty(field.Sy.value)) {
        if (! isEmpty(field.Da.value) || ! isEmpty(field.Db.value)) {
            field.Sy.value = "";
        }
    }
}
function resetDaAndDBIfNecessary() {
    var field = document.forms[0];
    if (! isEmpty(field.Da.value) || ! isEmpty(field.Db.value)) {
        if (! isEmpty(field.Sy.value)) {
            field.Da.value = "";
            field.Db.value = "";
        }
    }
}
function getFORMDATA(strFieldName) {
    if (FORM_DATA[strFieldName]){
        strResult = "" + FORM_DATA[strFieldName];
    } else {
        strResult = "";
    }
    return strResult;
}
////////////////////////////////////////////////////////////////////////
/*
Webmonkey GET Parsing Module
Language: JavaScript 1.0
The parsing of GET queries is fundamental
to the basic functionality of HTTP/1.0.
This module parses GET with JavaScript 1.0.
Source: Webmonkey Code Library
(http://www.hotwired.com/webmonkey/javascript/code_library/)
Author: Patrick Corcoran
Author Email: patrick@taylor.org
*/
function createRequestObject() {

  FORM_DATA = new Object();
    // The Object ("Array") where our data will be stored.

  separator = ',';
    // The token used to separate data from multi-select inputs

  myURL = '' + this.location;
    // Get the current URL so we can parse out the data.
    // Adding a null-string '' forces an implicit type cast
    // from property to string, for NS2 compatibility.

  query = myURL.substring((myURL.indexOf('?')) + 1);
    // Keep everything after the question mark '?'.
  if (query.length < 1 || query.length == myURL.length) { return false; }
```

```
    // Perhaps we got some bad data?

  keypairs = new Object();
  numKP = 1;
    // Local vars used to store and keep track of name/value pairs
    // as we parse them back into a usable form.

  while (query.indexOf('&') > -1) {
    keypairs[numKP] = query.substring(0,query.indexOf('&'));
    query = query.substring((query.indexOf('&')) + 1);
    numKP++;
      // Split the query string at each '&', storing the left-hand side
      // of the split in a new keypairs[] holder, and chopping the query
      // so that it gets the value of the right-hand string.
  }
  keypairs[numKP] = query;
    // Store what's left in the query string as the final keypairs[] data.

  for (i in keypairs) {
    keyName = keypairs[i].substring(0,keypairs[i].indexOf('='));
      // Left of '=' is name.
    keyValue = keypairs[i].substring((keypairs[i].indexOf('=')) + 1);
      // Right of '=' is value.
    while (keyValue.indexOf('+') > -1) {
      keyValue = keyValue.substring(0,keyValue.indexOf('+')) + ' ' +
      keyValue.substring(keyValue.indexOf('+') + 1);
        // Replace each '+' in data string with a space.
    }

    keyValue = unescape(keyValue);
      // Unescape non-alphanumerics

    if (FORM_DATA[keyName]) {
      FORM_DATA[keyName] = FORM_DATA[keyName] + separator + keyValue;
        // Object already exists, it is probably a multi-select input,
        // and we need to generate a separator-delimited string
        // by appending to what we already have stored.
    } else {
      FORM_DATA[keyName] = keyValue;
        // Normal case: name gets value.
    }
  }
  return FORM_DATA;
}
FORM_DATA = createRequestObject();
  // This is the array/object containing the GET data.
  // Retrieve information with 'FORM_DATA [ key ] = value'.
////////////////////////////////////////////////////////////////////////
/*
MOBIUS: auto fill form fields with data provided in a URL request
Language: JavaScript 1.0
This script is useful for providing an alternative method for modifying
an advanced keyword search on an III Millenium WebPAC. With this script,
you are no longer required to use III's tokens to provide this key
functionality. Which gives you a great deal of freedom to design a
user-friendly advanced keyword search form.
Author: Hardy Pottinger
Author Email: pottingerhj@umsystem.edu
*/
// only execute this script if there are any URL parameters AND there is a form on this page
if (FORM_DATA && document.forms.length > 0) {
    var field = document.forms[0];
    // handle the search field first
    /* try using search token instead
    if (FORM_DATA["NOSRCH"]){
        field.elements[0].value = FORM_DATA["NOSRCH"];
    } else {
        field.elements[0].value = FORM_DATA["SEARCH"];
    }
    */

```

```
340      // loop through the rest of the form fields
341      for (i = 1; i < field.length; i++) {
342
343          // using nested if/else statements instead of switch
344          // in order to maintain JavaScript 1.0 compatibility
345          // text area
346          if ((field.elements[i].type == "text") || (field.elements[i].type == "textarea")){
347              field.elements[i].value = getFORMDATA(field.elements[i].name);
348          } else {
349          // select box
350              if (field.elements[i].type.toString().charAt(0) == "s") {
351                  for (j = 0; j < field.elements[i].length; j++) {
352                      if ((field.elements[i].options[j].value ==
                        getFORMDATA(field.elements[i].name)) ||
353                          (field.elements[i].options[j].text ==
                            getFORMDATA(field.elements[i].name))) {
354                              field.elements[i].selectedIndex = j;
355                      }
356                  }
357              } else {
358          // radio button
359                  if ((field.elements[i].type == "radio")) {
360                      if (field.elements[i].value == getFORMDATA(field.elements[i].name)) {
361                          field.elements[i].checked = true;
362                      }
363                  } else {
364          // checkbox
365                      if (field.elements[i].type == "checkbox") {
366                          if
                            (getFORMDATA(field.elements[i].name).indexOf(field.elements[i].value)
                            <0) {
367                              field.elements[i].checked = false;
368                          } else {
369                              field.elements[i].checked = true;
370                          }
371                      }
372                  }
373              }
374          }
375      }
376      // handle special cases
377      // if there is an entry in the single year field, suppress entries in the range fields Da
         and Db
378      if (FORM_DATA["Sy"]) {
379          if (! isEmpty(FORM_DATA["Sy"])) {
380              field.Da.value = "";
381              field.Db.value = "";
382          }
383      }
384  }
385  ////////////////////////////////////////////////////////////////////
386  // set focus to the search box
387  document.frmAdvancedKeyword.SEARCH.focus();
388  // -->
389  </SCRIPT>
390  <P>
391  <table width="100%" border="0" cellspacing="0" cellpadding="0">
392    <tr>
393      <td height="1" bgcolor="#DEDDC9"><img src="/screens/shim.gif" width="1" height="2"
       border="0"></td>
394    </tr>
395  </table>
396  <a name="tips"><h3 align="center"><b>Search tips and examples</b></h3></a>
397  <TABLE CELLPADDING=4 CELLSPACING=2 WIDTH="100%">
398  <TR>
399  <TH>Search Type</TH>
400  <TH>Description</TH>
401  <TH>Examples</TH></TR>
402  <TR>
403  <TD WIDTH=100 BGCOLOR="#99CCFF" valign="top"><b>Phrase</b></TD>
404  <TD BGCOLOR="#99CCFF" valign="top">
```

```
Multiple words are searched together as an exact phrase.</TD>
<TD BGCOLOR="#99CCFF" valign="top"><TT>united states supreme court</TT></TD>
<TR>
<TD WIDTH=100 BGCOLOR="#99CCFF" valign="top"><b>Truncation</b></TD>
<TD BGCOLOR="#99CCFF" valign="top">Words may be truncated using an asterisk at the end.<br>
Use a single asterisk * to truncate from 1-5 characters.
Use a double asterisk ** for open-ended truncation.</TD>
<TD BGCOLOR="#99CCFF" valign="top"><TT>polic*
   <br>environm**</TT>
   </TD>
<TR>
<TD WIDTH=100 BGCOLOR="#99CCFF" valign="top"><b>Connectors</b></TD>
<TD BGCOLOR="#99CCFF" valign="top">Use &quot;and&quot; or &quot;or&quot; to specify multiple
words in any field.  Use &quot;and not&quot; to exclude words. </TD>
<TD BGCOLOR="#99CCFF" valign="top"><TT>dispute resolution and (united states or canada)
and not resident*</TT></TD>
<TR>
<TD WIDTH=100 BGCOLOR="#99CCFF" valign="top"><b>Proximity</b></TD>
<TD BGCOLOR="#99CCFF" valign="top">
Use &quot;within&quot and a number to specify proximity of words to one another, in any
order. </TD>
<TD BGCOLOR="#99CCFF" valign="top"><TT>employment within 3 discrimination</TT></TD>
<TR>
<TD WIDTH=100 BGCOLOR="#99CCFF" valign="top"><b>Field Search</b></TD>
<TD BGCOLOR="#99CCFF" valign="top">
Specify fields to search, using field abbreviation.  Fields available for
this database are<br> a: (author)<br>
t: (title)<br>
s: (LC subject)<br>
j: (Medical Subject)<br>
m: (Children's Subject)
</TD>
<TD BGCOLOR="#99CCFF" valign="top"><TT>a:shakespeare and t:hamlet</TT><br><br>
<TT>education and a:missouri and (s:handicapped or s:disabled)</TT></TD></TR>
</TABLE>
<p>
</center>
<!-- End Search Help Text -->
<p align="center"><A HREF="/search~S0"><IMG SRC="/screens/b_newsrch.gif" BORDER=0 ALT="[START
OVER]"></A></p>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td colspan="3" height="1" bgcolor="#DEDDC9"><IMG SRC="/screens/shim.gif" HEIGHT="2"
    WIDTH="1" BORDER="0" ALT=""></td>
  </tr>
  <tr>
    <td align="right" valign="bottom"> </td>
    <td align="right" valign="middle" nowrap><font face="Verdana, Arial, Helvetica,
    sans-serif" size="-2"><b>
      <a href="http://mobius.missouri.edu/search/">Search MOBIUS</a> |
      <a href="/search~">Search Bridges</a> |
      <a href="/">Bridges Home</a> |
      <a href="/screens/help.html">Help</a>
      </b></font></td>
  </tr>
</table>
</BODY>
</HTML>
```