
OPAC: Add Links to the Action Box

Created By: Laura Guy
Created on: 5/14/2019

[This article was copied from the Voyager Wiki.]

Introduction

This relates to the "New" Tomcat WebVoyage that is included in Voyager 7.0.1 and forward.

We wanted to add four links to the Record Display "Action Box" (see the screenshot below). Two of these were static links to help pages, two are search redirects allowing patrons to do a simple ISBN search against our consortium's unified catalog and against WorldCat. These were capabilities provided in our 'classic' interface and we wished to continue their availability in the new interface.

Create xsl file

Each of the four links is a separate xsl file. These files live here: /m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/your_SKIN/xsl/pageFacets

The following example is the xsl file for creating the "Item Status Help" link you see in the following screenshot (filename: infoStatus.xsl)

Code Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
**          Note: Creates a link to Location Information
**          Version : 1.0
**          Created : 15-MAY-08
**          Created By : LG
**          Description: infoStatus.xsl creates static link to status info
-->
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:page="http://www.exlibrisgroup.com/voyager/webvoyage/page"
```

```

xmlns:fo="http://www.w3.org/1999/XSL/Format">

<!-- ##### -->
<xsl:template name="infoStatus">
<a target="_blank">
  <xsl:attribute name="href">http://www.mines.edu/library/instruc...st-status.html</xsl:attribute>Item
  Status Help</a>
</xsl:template>
</xsl:stylesheet>

```

Create a different xsl file

The next example is the xsl file for creating the "Search WorldCat for Item" link you see in the following screenshot (filename: isbnwcSearch.xsl). It looks at the first 020 field, trims any unwanted data, and uses it to perform an isbn search against WorldCat

Code Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
**          Note: Creates a link to WorldCat using the ISBN
**          Version : 1.0
**          Created : 15-APR-08
**          Created By : ASP
**          Modified By : Laura Guy 5-20-08
**          Modified By : Laura Guy 12-08-08
-->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:page="http://www.exlibrisgroup.com/voyager/webvoyage/page"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
<!-- ##### -->
<xsl:template name="recordIsbnSearchwc">

```

```

<xsl:variable name="isbn">

<xsl:call-template name="trimData">

  <xsl:with-param name="sData">

<xsl:call-template name="BMDProcessMarcTags">

  <xsl:with-param name="field" select="'020'"/>

  <xsl:with-param name="indicator1" select="'X'"/>

  <xsl:with-param name="indicator2" select="'X'"/>

  <xsl:with-param name="subfield" select="'a'"/>

  <xsl:with-param name="mfhdID" select="$bibID"/>

  <xsl:with-param name="recordType" select="'bib'"/>

  </xsl:call-template>

  </xsl:with-param>

</xsl:call-template>

</xsl:variable>

<xsl:if test="$isbn!=''">

<a target="_blank">

  <xsl:attribute name="href">http://worldcatlibraries.org/isbn/<xsl:value-of select="$isbn"/>
  &loc=united+states</xsl:attribute>Search WorldCat for Item</a>

</xsl:if>

</xsl:template>

</xsl:stylesheet>

```

Edit displayFacets.xsl File

Once you have created your xsl file, you need to add it to the displayFacets.xsl file located here:

/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/your_SKIN/xsl/pageFacets/

1) It needs to be added as an include at the top of the file (the below example shows four of these includes that I've added, along with the googleBooks one that came out-of-the-box):

Code Example

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<!--
#(c) #=====
#(c) #
#(c) #      Copyright 2007 ExLibris Group
#(c) #      All Rights Reserved
#(c) #
#(c) #=====
-->
<!--
**      Product : : WebVoyage : displayFacets
**      Version : 1.0
**      Created : 26-OCT-2007
**      Orig Author : David Sellers
**      Last Modified : 03-APR-2008
** Last Modified By : Mel Pemble
**      Modified By : Laura Guy - 5/15/08 - to add staticMore Info links and search redirects
-->

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:page="http://www.exlibrisgroup.com/voyager/webvoyage/page"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:include href="../local_googleBooksAvail.xsl"/>
<xsl:include href="./infoLocation.xsl"/>
<xsl:include href="./infoStatus.xsl"/>
<xsl:include href="./isbnprSearch.xsl"/>
<xsl:include href="./isbnwcSearch.xsl"/>

<!-- ##### -->

```

2) Then you need to call that file lower down in the displayFacets file (look for the 'div id="infoStatus"' below):

Code Example

```

<!-- ##### -->
<xsl:template name="buildActionBox">
<xsl:param name="pageRecordType"/>

<xsl:for-each select="page:element[@nameId='actionBox.group']">
  <div class="actionBox">
    <a id="actionBoxQuickLink" name="actionBoxQuickLink"></a>
    <!--h3>Action Box</h3-->
    <!--ul class="actionBoxList"-->

    <xsl:for-each select="page:element[@nameId='actionBox.thisItem.group']">
      <div class="thisItem">
        <label><xsl:value-of select="page:label"/></label>
        <h2 class="nav">Action Box</h2>
        <ul title="Action Box">

          <xsl:for-each select="page:element">
            <xsl:choose>
              <xsl:when test="@nameId=$pageRecordType">
                <li><span class="recordPointer"> </span><label><xsl:value-of
select="page:linkText"/></label></li>
              </xsl:when>
              <xsl:otherwise>
                <li><span class="recordLinkBullet">•</span><a
href="{page:URL}"><span><xsl:value-of select="page:linkText"/></span></a></li>
              </xsl:otherwise>
            </xsl:choose>
          </xsl:for-each>
        </ul>
      </div>
    </xsl:for-each>

    <xsl:for-each select="page:element[@nameId='actionBox.actions.group']">
      <div class="actions">

```

```

<label><xsl:value-of select="page:label"/></label>

<h2 class="nav">Action Navigation</h2>

<ul title="Action Navigation" class="actions">

    <xsl:for-each select="page:element">

        <li><span class="recordLinkBullet">•</span><a
href="{page:URL}"><span><xsl:value-of select="page:linkText"/></span></a><br/><span
class="fieldSubText"><xsl:value-of select="page:postText"/></span></li>

        </xsl:for-each>

    </ul>

</div>

</xsl:for-each>

<xsl:for-each select="page:element[@nameId='actionBox.moreAboutThisItem.group']">

    <div class="moreInfo">

        <label><xsl:value-of select="page:label"/></label>

        <h2 class="nav">Action Box More Info Navigation</h2>

        <ul title="Action Box More Info Navigation" class="moreInfo">

            <xsl:for-each select="page:element">

                <li>

                    <xsl:call-template name="buildImageUrl">

                        <xsl:with-param name="eleName" select="@nameId"/>

                    </xsl:call-template>

                </li>

            </xsl:for-each>

        </ul>

    </div>

    <div id="infoLocation">

<ul><li><span class="recordLinkBullet">•</span>

        <xsl:call-template name="infoLocation" />

</li></ul>

```

```

        </div>

        <div id="infoStatus">
<ul><li><span class="recordLinkBullet">--</span>

        <xsl:call-template name="infoStatus" />
</li></ul>

        </div>

        <div id="prospSearch">
<ul><li><span class="recordLinkBullet">--</span>

        <xsl:call-template name="recordIsbnSearch" />
</li></ul>

        </div>

        <div id="wcSearch">
<ul><li><span class="recordLinkBullet">--</span>

        <xsl:call-template name="recordIsbnSearchwc" />
</li></ul>

        </div>

        <br />

        <!-- ## mdp add the google book template ## -->

        <div id="googleBooksRow">

        <xsl:call-template name="googleBooksAvail"/>

        </div>

        </xsl:for-each>

        <!--/ul-->

        </div>

</xsl:for-each>

</xsl:template>

<!-- ##### -->

</xsl:stylesheet>

```

Finally update your webvoyage.properties file

Lastly, you need to add your new section to the webvoyage.properties file. Note the 'actionBox.moreAboutThisItem' line:

Code Example

```
#####  
#  
# Holdings Information Section  
#  
#####  
holdingsInfo.page.title=Catalyst Holdings Information  
holdingsInfo.page.heading=Holdings Information  
holdingsInfo.page.label=Bibliographic Details  
  
#=====  
# Action box on the holding information page  
#=====  
actionBox.thisItem=This item  
actionBox.recordView.linkText=Record View  
actionBox.staffView.linkText=Staff View  
actionBox.actions=Actions  
actionBox.actions.makeRequest.order=1  
actionBox.actions.makeRequest.linkText=Make a Request  
actionBox.actions.makeRequest.link.posttext=(e.g. Hold, Recall, Request)  
actionBox.actions.print.order=2  
actionBox.actions.print.linkText=Print  
actionBox.actions.export.order=3  
actionBox.actions.export.linkText=Export  
actionBox.actions.email.order=4  
actionBox.actions.email.linkText=E-mail  
#NOTE the option.email variable must be Y in order for the email link to appear.  
actionBox.actions.addToList.order=5  
actionBox.actions.addToList.linkText=Add to My List  
# This additional section is additional Library Information  
# LG
```

```
actionBox.moreAboutThisItem=More Library Information:  
# actionBox.openURL properties in holdingsInfo.X.properties  
# actionBox.requestURI properties in holdingsInfo.X.properties
```

[Report](#)