



18th Annual CODI Conference
Pittsburgh Pennsylvania

NOVEMBER 2007

Customizing HIP 2.x/3.x Advanced XSLT Stylesheet Editing

Phil Feilmeyer
Hennepin County (MN) Library

Customizing at Hennepin County Library

- HIP 3.08 running on Windows 2003
- Horizon 7.34
- HCL is a single library with 26 branches
 - Each branch has a HIP profile, but they are essentially identical
 - Decision-making aspects of customization are centralized
 - Only one [strings.xsl](#), [other.xsl](#), [onoff.xsl](#)
- Experience with HTML and ColdFusion
 - ColdFusion is a web application platform that uses templates and a tag-based programming language

Customizing at Hennepin County Library

- Resources available for customization effort
 - Development server
 - Staff with experience in web application development
- Started with cosmetic changes
 - Reduction of white space
 - Removal of extraneous information
 - Color conformity with the library's web pages

Customizing at Hennepin County Library

- Breakthrough moment: realizing there are two parts to every HIP page:
 - Functional elements
 - links, form tags, input boxes, submit buttons, JavaScript
 - Display elements
- As long as the **functional elements** are left intact, you can do as you like with the **display elements**
- First major customization: **toolbar2.xsl**

Header Example (toolbar2.xsl)

HORIZON
Information Portal

 Login  My List - 0  Help

Search

My Account

Basic | Keyword | Format | Advanced | History

Basic Search



START OVER HOURS LIBRARY CARDS INTERLIBRARY LOAN SUGGEST A TITLE FEEDBACK

BASIC SEARCH

SEARCH WITH LIMITS

ADVANCED SEARCH

BOOKSPACE

MY ACCOUNT

 Exit  Login

 Help  My List - 0

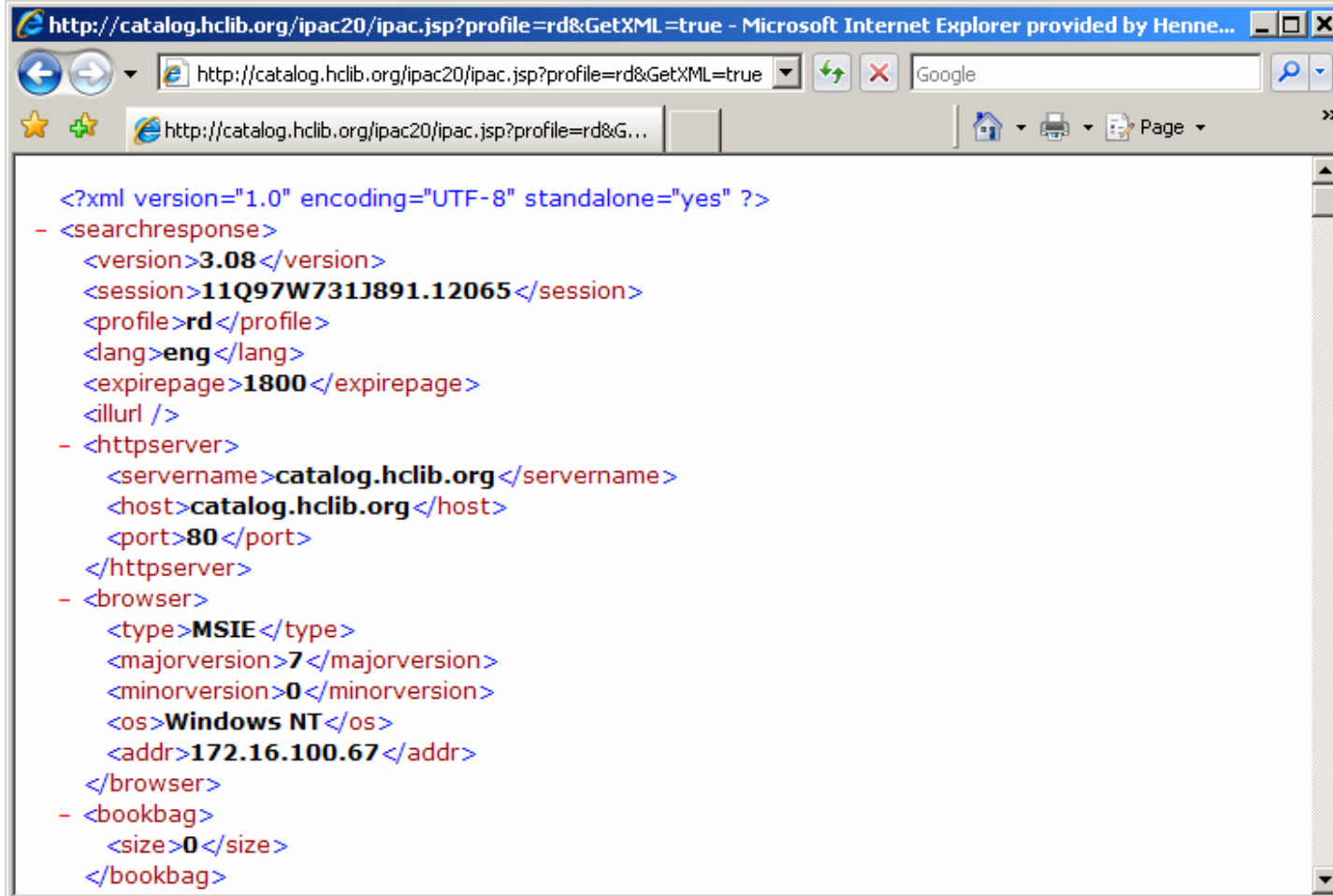


Basic Search

Viewing XML Data

- To see the XML behind most HIP pages, add **&GetXML=true** to the URL
 - Case sensitive
 - Remove "#focus" from the end of the URL first
 - Example:
 - <http://catalog.hclib.org/ipac20/ipac.jsp?profile=rd#focus>
 - <http://catalog.hclib.org/ipac20/ipac.jsp?profile=rd&GetXML=true>
 - You are re-submitting the HIP request, so you may be re-renewing or re-reserving a title
 - Occasionally it doesn't work
 - address bar only contains base URL

Viewing XML Data



The screenshot shows a Microsoft Internet Explorer browser window. The address bar contains the URL: `http://catalog.hclib.org/ipac20/ipac.jsp?profile=rd&GetXML=true`. The browser's content area displays the following XML data:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <searchresponse>
  <version>3.08</version>
  <session>11Q97W731J891.12065</session>
  <profile>rd</profile>
  <lang>eng</lang>
  <expirepage>1800</expirepage>
  <illurl />
- <httpserver>
  <servername>catalog.hclib.org</servername>
  <host>catalog.hclib.org</host>
  <port>80</port>
</httpserver>
- <browser>
  <type>MSIE</type>
  <majorversion>7</majorversion>
  <minorversion>0</minorversion>
  <os>Windows NT</os>
  <addr>172.16.100.67</addr>
</browser>
- <bookbag>
  <size>0</size>
</bookbag>
```

XML Path Language (XPath)

- Syntax for referencing elements of an XML document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<searchresponse>
  <version>3.08</version>
  <session>11Q97W731J891.12065</session>
  <profile>elibrary</profile>
  <expirepage>300</expirepage>
  ...
</searchresponse>
```

XPath reference to the user's profile:

`/searchresponse/profile`

XML Path Language (XPath)

- Absolute path starts with a 'slash'
`/patronpersonalresponse/patroninfo/name/full`
- Use double-slash to left truncate
`//profile`
`//session`
- Include position in brackets for 'multivalue fields'
`//searchresults/results/row[3]/TITLE/data/text`
- No initial 'slash' for a relative path
`TITLE/data/text`
 - Context can be provided by a `<xsl:for-each>` tag
`<xsl:for-each select="//searchresults/results/row">`
 ... `<xsl:value-of select="TITLE/data/text"/>` ...
`</xsl:for-each>`

Benefits of looking at the XML

Patron personal data is part of the XML on every HIP page after login

```
<searchresponse>
...
<security>
  <auth>>true</auth>
  <name>FEILMEYER, PHILIP A</name>
  <patronid>831256</patronid>
  <address>12601 RIDGEDALE DR, MINNETONKA, MN 55305</address>
  <emailaddress>pfeilmeyer@hclib.org</emailaddress>
  <phone>952-847-8768</phone>
</security>
...
</searchresponse>
```

The email address in XPath: `/searchresponse/security/emailaddress`
`//security/emailaddress`

Benefits of looking at the XML

- Custom XSLT to automatically include the patron's email address in the input box whenever an email option appears

```
<input>  
  <xsl:attribute name="class">  
    <xsl:value-of select="$css_text_input_box"/>  
  </xsl:attribute>  
  <xsl:attribute name="type">text</xsl:attribute>  
  <xsl:attribute name="size">40</xsl:attribute>  
  <xsl:attribute name="name">emailaddress</xsl:attribute>  
  <xsl:attribute name="maxlength">100</xsl:attribute>  
  <xsl:if test="string-length(//security/emailaddress) > 0">  
    <xsl:attribute name="value">  
      <xsl:value-of select="//security/emailaddress"/>  
    </xsl:attribute>  
  </xsl:if>  
</input>
```

Benefits of looking at the XML

Your request has been successfully placed	
London calling	
Requested by: PHILIP A FEILMEYER	
You will be notified when this title is available	
Pickup Location: Ridgedale	
<input type="button" value="Return to Searching"/>	<input type="button" value="Logout and Return"/>

```
<tr>
  <td>Pickup Location: <xsl:value-of select="pickup_location"/></td>
</tr>
<xsl:if test="not(boolean(//security/emailaddress))">
  <tr><td>
    <img>
      <xsl:attribute name="src"/>/images/email.gif</xsl:attribute>
    </img>
  </td></tr>
  ...
</xsl:if>
```

Benefits of looking at the XML

Your request has been successfully placed

London calling

Requested by: **PHILIP A FEILMEYER**

You will be notified when this title is available

Pickup Location: **Ridgedale**



Did you know you can
get your reserve and
overdue notices by email
~ *instead of by phone
or U. S. mail?*

Make the switch to email notification now!

[Make the Switch](#)

[Return to Searching](#)

[Logout and Return](#)

Editing example

Make something conditional based on profile

```
<xsl:if test="not(//profile = 'elibrary')">  
  <tr>  
    <td>  
      To protect your personal information, close this  
      browser window before you leave the workstation.  
    </td>  
  </tr>  
</xsl:if>
```

Another editing example

Make something conditional based on profile

```
<xsl:choose>  
  <xsl:when test="//profile = 'elibrary'">  
    call your local library  
  </xsl:when>  
  <xsl:otherwise>  
    ask for assistance at the info desk  
  </xsl:otherwise>  
</xsl:choose>
```

Adding Customized Data Elements using the HIP Admin Tool

- OCLC# for staff (but not the public) in the Bib Display
 - Add new element to Full Bib Display using the Admin Tool

	Label	Fixed Element	Table Name	Marc Map
1	<input type="text" value="Title"/>	<i>Title</i>	<input type="text" value="bib"/>	<input type="text" value="Dtitle - Title"/> ▾
2	<input type="text" value="Author"/>	<i>Author</i>	<input type="text" value="bib"/>	<input type="text" value="D100p - Main Entry plus"/> ▾
3	<input type="text" value="Published"/>	<i>Publisher</i>	<input type="text" value="bib"/>	<input type="text" value="D260p - Publication info"/> ▾
4	<input type="text" value="ISBN"/>		<input type="text" value="bib"/>	<input type="text" value="D020p - ISBN/ISSN"/> ▾
5	<input type="text" value="Subject"/>	<i>Subject</i>	<input type="text" value="bib"/>	<input type="text" value="D600p - Subjects"/> ▾
6	<input type="text" value="Contents"/>		<input type="text" value="bib"/>	<input type="text" value="D505 - Contents Note"/> ▾
7	<input type="text" value="Summary"/>		<input type="text" value="bib"/>	<input type="text" value="D520 - Summary Note"/> ▾
8	<input type="text" value="OCLC#"/>		<input type="text" value="bib"/>	<input type="text" value="D001 - Control Number"/> ▾

Adding Customized Data Elements using the HIP Admin Tool

- Check the XML to find out how the data is structured

```
<searchresponse>
  <fullnonmarc>
    <searchresults>
      <header>
        <col>
          <label>ISBN</label>
        </col>
        <col>
          <label>Contents</label>
        </col>
        <col>
          <label>Summary</label>
        </col>
        <col>
          <label>OCLC#</label>
        </col>
      </header>
      <results>
        <row>
          <cell>
            <text>0375422889</text>
          </cell>
          <cell>
            <text></text>
          </cell>
          <cell>
            <text>Continues the author's ...</text>
          </cell>
          <cell>
            <text>ocm54005645</text>
          </cell>
        </row>
      </results>
    </searchresults>
  </fullnonmarc>
</searchresponse>
```

Adding Customized Data Elements using the HIP Admin Tool

- The original [fullnonmarcbib.xsl](#) stylesheet simply creates a row for each of the additional data elements

```
<!-- build nonfixed element display fields -->
<xsl:for-each select="//searchresults/results/row/cell">
  <xsl:variable name="pos">
    <xsl:value-of select="position()" />
  </xsl:variable>
  <xsl:choose>
    <xsl:when test="boolean(normalize-space(text))">
      [create a table row with label and data]
    </xsl:when>
    <xsl:otherwise>
      [don't do anything]
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each>
```

Adding Customized Data Elements using the HIP Admin Tool

- Identify the position of the label of your new element using `<xsl:variable>` at the upper-most level in the template

```
<xsl:variable name="oclc_pos">  
  <xsl:for-each select="//searchresults/header/col">  
    <xsl:if test="label = 'OCLC#' ">  
      <xsl:value-of select="position() "/>  
    </xsl:if>  
  </xsl:for-each>  
</xsl:variable>
```

Adding Customized Data Elements using the HIP Admin Tool

- Insert your custom code in the appropriate spot

```
<!-- fixed elements XSLT ends here -->
```

```
<xsl:if test="boolean(//searchresults/results/row/cell[$oclc_pos]/text)">  
  <tr>  
    <td>  
      <xsl:value-of select="//searchresults/header/col[$oclc_pos]/label"/>  
    </td>  
    <td>  
      <xsl:value-of select="//searchresults/results/row/cell[$oclc_pos]/text"/>  
    </td>  
  </xsl:if>
```

```
<!-- nonfixed elements XSLT starts here -->
```

Adding Customized Data Elements using the HIP Admin Tool

- Modify the original stylesheet to exempt this data from the generic display

```
<!-- build nonfixed element display fields -->
<xsl:for-each select="//searchresults/results/row/cell">
  <xsl:variable name="pos">
    <xsl:value-of select="position()" />
  </xsl:variable>
  <xsl:choose>
    <xsl:when test="boolean(normalize-space(text)) and ($pos != $oclc_pos)">
      [create a table row with label and data]
    </xsl:when>
    <xsl:otherwise>
      [don't do anything]
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each>
```

Data not available from the XML

Staff Catalog - Microsoft Internet Explorer provided by Hennepin County Library

http://hatchibombator.hclib.org/ipac20/ipac.jsp?session=1I920382R0G19.9438

Staff Catalog

HENNEPIN COUNTY Library

START OVER HOURS CIV INTERLIBRARY LOAN SUGGEST A TITLE FEEDBACK

BASIC SEARCH SEARCH WITH LIMITS ADVANCED SEARCH BOOKSPACE CUSTOMER ACCOUNT

PHILIP A FEILMEYER
21972014526334 (831256)

Exit Logout Help My List - 5

Basic Search

New Search: TITLE starts with GO

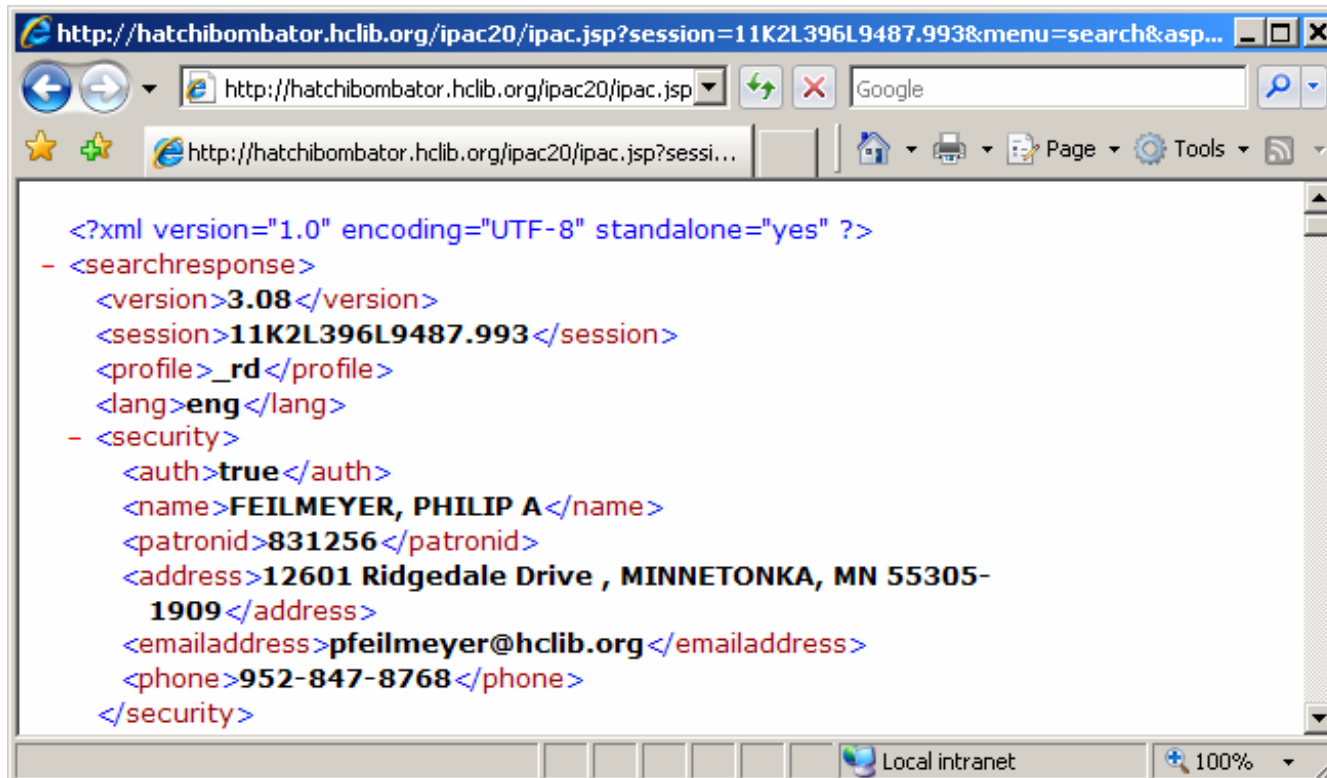
Previous 10 Titles Next 10 Titles

Title	Author	Format	Date
1. Syllable from sound.		Video Tape	2002
2. Sylvanus Now : a novel /	Morrissey, Donna, 1956-	Book	2006
3. Sylvester and the magic pebble.	Steig, William, 1907-2003	Book	1969

Local intranet 100%

Data not available from the XML

Borrower ID, but no barcode number in the XML



```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <searchresponse>
  <version>3.08</version>
  <session>11K2L396L9487.993</session>
  <profile>_rd</profile>
  <lang>eng</lang>
- <security>
  <auth>true</auth>
  <name>FEILMEYER, PHILIP A</name>
  <patronid>831256</patronid>
  <address>12601 Ridgedale Drive , MINNETONKA, MN 55305-1909</address>
  <emailaddress>pfeilmeyer@hclib.org</emailaddress>
  <phone>952-847-8768</phone>
</security>
```

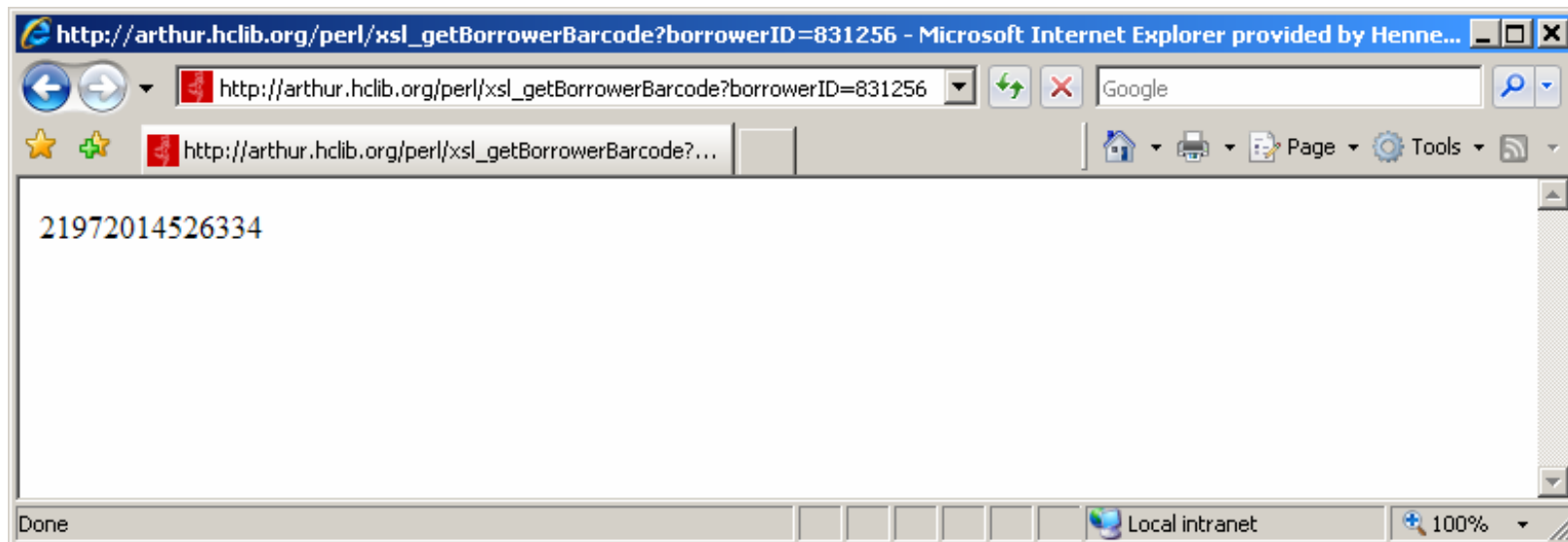
- **getURL()** - Custom java function in the XSL Processor that pulls in content from a web page
 - written by Casey Durfee (formerly of Seattle Public Library)
 - <http://extranet.spl.org/rss-0.5.zip>
 - Instructions: getURL.txt
 - Code: Utils.jar (in the optional folder)
- Create a custom web application to deliver the desired information
 - we use Perl CGI and ColdFusion

Data not available from the XML

http://arthur.hclib.org/perl/xsl_getBorrowerBarcode?borrowerID=831256

- Takes the borrower ID number from the http request
- Queries the Horizon database to find out the borrower barcode

```
select bbarcode from borrower_search_cf_view where borrower#  
= 831256
```
- Returns the borrower barcode



Data not available from the XML

toolbar2.xsl

```
<td>
  <xsl:value-of select="//security/name"/>
  <xsl:if test="$isStaffProfile = 'true'">
    <xsl:variable name="url">
      http://arthur/perl/xsl_getBorrowerBarcode?borrowerID=
      <xsl:value-of select="//security/patronid"/>
    </xsl:variable>
    <br/>
    <xsl:value-of select="Utils:getURL($url)"/>
  </xsl:if>
</td>
```

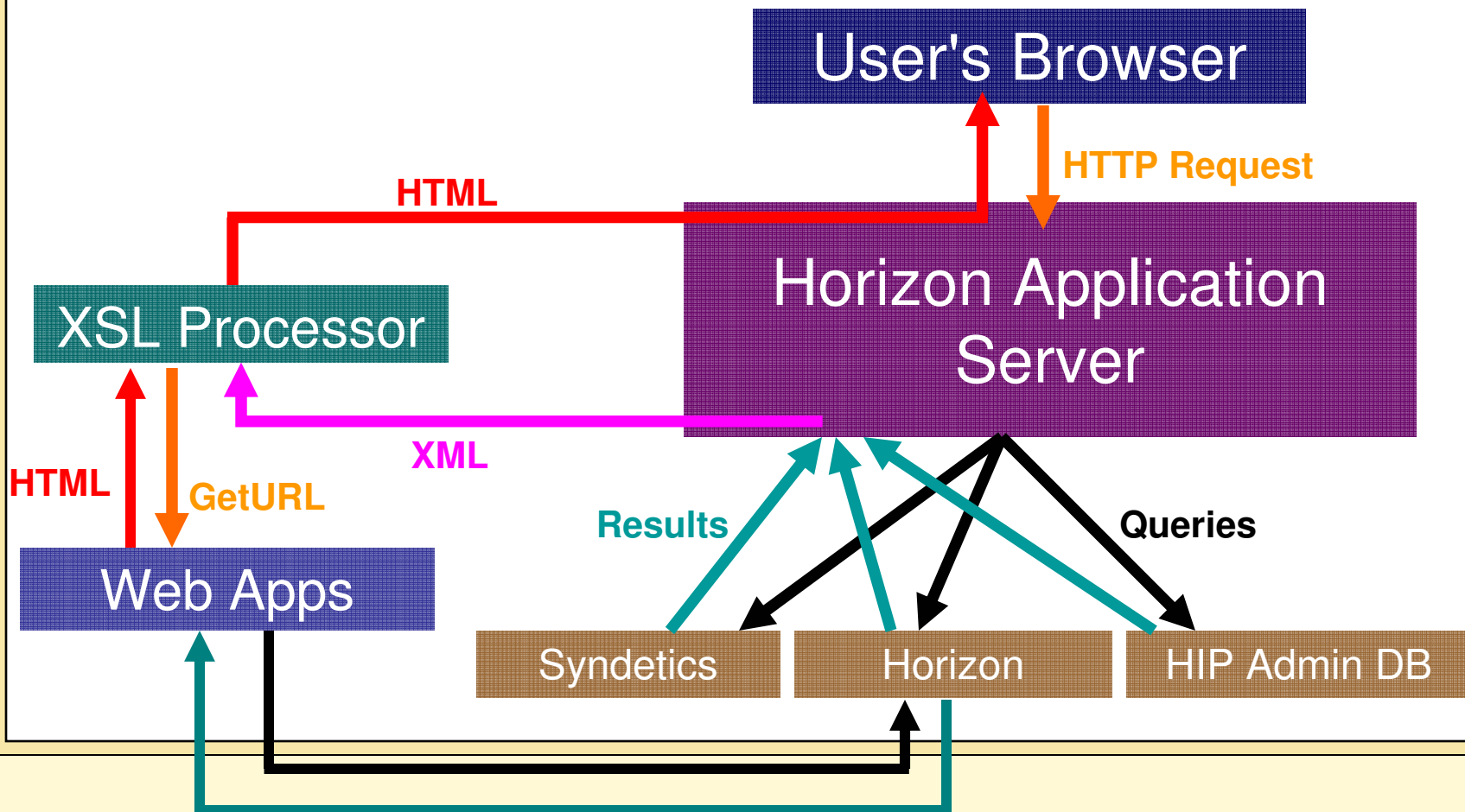
Data not available from the XML

Taken to extremes...

[xsl_getSummaryViewData?bibID=3092569&loc=rd](#)

```
numRequests=0+checkedIn=somewhere~s+refRent=all~  
24~ref~0~bsx~0~onorder~0~onorder_avail~0~+reqData  
=22~22+ldr06=a+ldr07=m+desc00=+desc01=+desc04=+  
fixed21= +fixed23= +fixed29=0+URL=+excerptURL=+  
niceTitle=Sleepover Larry
```

HIP Architecture: Anatomy of a HIP Request



Advanced Customizing - Other Possibilities

- Add javascript to a page
 - "Validate" email address submission
 - Confirmation dialog box when canceling requests
- Connect dynamically to another web application
 - Due date calendar
 - Show location of available copies
 - Update borrower information

Managing Customized Stylesheets - the Hennepin Method

- Make a backup copy of the original `xsltransformer.xml` file for the current version
 - I called ours `xsltransformer_306a_original.xml`
- Identify stylesheets that have been customized using unique filenames
 - Whenever a customization is made to a stylesheet (no matter how small), give the edited file a new name and leave the unedited, original version in the XSL folder
 - `generalfooter.xsl` becomes `hcl_generalfooter_306a.xsl` and the original `generalfooter.xsl` remains in the XSL folder

Managing Customized Stylesheets

- Update any and all references to the customized files in **xsltransformer.xml**

```
<Root>
```

```
  <Name>./xsl/mylistresponse.xsl</Name>
```

```
  <Includes>
```

```
    <Include type="timer">expiretimer.xsl</Include>
```

```
    <Include type="features">onoff.xsl</Include>
```

```
    <Include type="image">image.xsl</Include>
```

```
    <Include type="attributes">hcl_other_306a.xsl</Include>
```

```
    <Include type="mylist">hcl_mylist_306a.xsl</Include>
```

```
    <Include type="string">hcl_string_306a.xsl</Include>
```

```
    <Include type="error">error.xsl</Include>
```

```
    <Include type="toolbar">hcl_toolbar2_306a.xsl</Include>
```

```
    <Include type="footer">hcl_generalfooter_306a.xsl</Include>
```

```
  </Includes>
```

```
</Root>
```

Managing Customized Stylesheets

- If you use a 'Find and Replace' tool to update the entries in [xsltransformer.xml](#), watch out for similarly named stylesheets
 - [items.xsl](#)
 - [summaryitems.xsl](#)
- The **root** files cannot be renamed, so if you customize them you'll need to keep track "manually"
 - Keep copies of the unedited originals around with an identifying filename (example: [searchresponse_original_3.06a.xsl](#))

Managing Customized Stylesheets

- Upgrade Procedure - use for upgrades, hotfixes, service packs, patches
 - Step 1: Before the upgrade, copy the full **xsl** folder/directory to an available location (e.g. elsewhere on the HIP server, workstation, network drive, thumb drive, etc.)
 - Step 2: Rename the working version of **xsltransformer.xml** to something else, and then rename the original version back
 - **xsltransformer.xml** becomes **xsltransformer_306a.xml**
 - **xsltransformer_original_306a.xml** becomes **xsltransformer.xml**

Managing Customized Stylesheets

- Step 3: If you have customized any of the **root** stylesheets:
 - Rename the current customized versions using an identifiable filename
 - **searchresponse.xsl** becomes **searchresponse_306a.xsl**
 - Put the original, unedited versions back to their original filename
 - **searchresponse_original_306a.xsl** becomes **searchresponse.xsl**

Now you should be back to "out-of-the-box" and ready for the upgrade.

Managing Customized Stylesheets

- Step 4: Run the upgrade
 - After it's finished, you should have a default installation of the new version of the XSL Processor
- Step 5: After the upgrade
 - For each of the stylesheets you've customized, compare the pre-upgrade **originals** to the post-upgrade **originals**.
 - file size
 - date/time modified
 - If there are no changes in the original versions, then your customized version can be re-integrated without modification (except for a filename change)
 - **hcl_generalfooter_306a.xsl** becomes **hcl_generalfooter_309.xsl**

Managing Customized Stylesheets

- Step 5: After the upgrade (continued)
 - If the post-upgrade original **IS** different than the pre-upgrade original, use a "Compare Files" utility to determine the nature of the changes
 - If the changes are relatively minor, incorporate them (if appropriate) into your customized version
 - If the changes are major compared to your customizations, recreate your customizations into the new version of the stylesheet
 - In either case, make sure that the new, original version of each stylesheet with it's original filename is left in the XSL folder
 - Don't forget the **root** stylesheets if they have been customized

Managing Customized Stylesheets

- Step 6: Update **xsltransformer.xml**
 - Make a backup copy of the post-upgrade **xsltransformer.xml**
 - Edit the new version of **xsltransformer.xml**, updating any and all references to the customized stylesheets



18th Annual CODI Conference
Pittsburgh Pennsylvania

NOVEMBER 2007

Customizing HIP 2.x/3.x: Advanced XSLT Stylesheet Editing

Phil Feilmeyer

Hennepin County (MN) Library

pfeilmeyer@hclib.org

Slides: www.hclib.org/extranet