Spellchecker for WebVoyage 8

1. Download spellcheck files

Download and unzip tomcat_spellcheck-2.4.8.zip or untar tomcat_spellcheck-2.4.8.tar.gz. Inside you will find:

- instructions.pdf
- spellcheck/
- spellcheck.js

Steps: Log in to your server as voyager and type the following commands:

```
cd /export/voyager/home

wget http://lib-serv.tccd.edu/code/webvoyage/spellcheck/tomcat/tomcat_spellcheck-2.4.8.tar.gz

gunzip tomcat_spellcheck-2.4.8.tar.gz

tar xvf tomcat_spellcheck-2.4.8.tar

cd tomcat_spellcheck-2.4.8
```

2. Install the spellcheck files

In /m1/voyager/{xxxdb}/tomcat/vwebv/, copy the entire spellcheck directory.

In /m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/jscripts, copy the spellcheck.js file.

Steps:

```
cp spellcheck/  /m1/voyager/{xxxdb}/tomcat/vwebv/

cp spellcheck.js /m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/jscripts/
```

3. Add the spellcheck context to tomcat

   A. Stop tomcat: **This is important! *server.xml* should not be edited while Tomcat is running!**

   ```
   /m1/voyager/{xxxdb}/tomcat/tsvrctl stop
   ```

   B. Back up server.xml:

   ```
   cd /m1/voyager/{xxxdb}/tomcat/conf
   cp server.xml server.xml.bak
   ```

   Note: To reduce downtime, you can also edit the server.xml.bak file, then stop Tomcat, then rename the *server.xml* files, then restart Tomcat.

   C. Edit server.xml:
      1. Copy the *vwebv* context in server.xml (highlighted here in yellow).
      2. Paste a copy below the *vwebv* context to create a new context.
      3. Make the changes indicated in **red** to the new context:

```
<Context className="org.apache.catalina.core.StandardContext"
    path="/vwebv"
    cachingAllowed="true"
    charsetMapperClass="org.apache.catalina.util.CharsetMapper"
    cookies="true"
    crossContext="false"
    displayName="Voyager WebVoyage"
    docBase="/m1/voyager/tardb/tomcat/vwebv/context/vwebv"
        privileged="true"
        reloadable="false"
        swallowOutput="false"
        useNaming="true"
        wrapperClass="org.apache.catalina.core.StandardWrapper">
</Context>

<Context className="org.apache.catalina.core.StandardContext"
    path="/spellcheck"
    cachingAllowed="true"
    charsetMapperClass="org.apache.catalina.util.CharsetMapper"
    cookies="true"
    crossContext="false"
    displayName="Voyager WebVoyage Spellcheck"
    docBase="/m1/voyager/tardb/tomcat/vwebv/spellcheck"
        privileged="true"
        reloadable="false"
        swallowOutput="false"
        useNaming="true"
        wrapperClass="org.apache.catalina.core.StandardWrapper">
</Context>
```

Steps:

```
/m1/voyager/{xxxdb}/tomcat/tsvrctl stop

cd /m1/voyager/{xxxdb}/tomcat/conf

cp server.xml server.xml.bak
```

Use your favorite code editor to make the changes to server.xml indicated above.

4. Tell apache to allow tomcat to display the new context by ading the code in **red**

A. Find the apache configuration files:
`cd /m1/shared/apache2/conf/ConfiguredVirtualHosts`

B. Back up *xxxdb_vwebv_httpd.conf:*
`cp xxxdb_vwebv_httpd.conf xxxdb_vwebv_httpd.conf.bak`

C. Back up *xxxdb.jkmounts.conf*:
`cp xxxdb.jkmounts.conf xxxdb.jkmounts.conf.bak`

D. Edit *xxxdb_vwebv_httpd.conf* thusly:

```
Alias /vwebv/ui/ "/m1/voyager/xxxdb/tomcat/vwebv/context/vwebv/ui/"
Alias /spellcheck/ "/m1/voyager/xxxdb/tomcat/vwebv/spellcheck/"
```

E. Edit *xxxdb.jkmounts.conf* by adding the following line to the bottom of the list:

```
JkMount /vwebv/vwe* ajp13_lb_xxxdb_vwebv
JkMount /spellcheck* ajp13_lb_xxxdb_vwebv
```

F. Don't forget to change **xxxdb** to your database name.

G. Restart Apache server `/etc/init.d/httpd2 stop; /etc/init.d/httpd2 start`

H. Start Tomcat server `/m1/voyager/{xxxdb}/tomcat/tsvrctl start`

Steps:

```
cd /m1/shared/apache2/conf/ConfiguredVirtualHosts

cp xxxdb_vwebv_httpd.conf xxxdb_vwebv_httpd.conf.bak

cp xxxdb.jkmounts.conf xxxdb.jkmounts.conf.bak
```

Use your favorite code editor to make the changes to the two files as indicated above (steps D and E).

```
/etc/init.d/httpd2 stop; /etc/init.d/httpd2 start

/m1/voyager/{xxxdb}/tomcat/tsvrctl start
```

5. Tell the spellchecker where to find the desired dictionary

   A. Edit */m1/voyager/{xxxdb}/tomcat/vwebv/spellcheck/WEB-INF/spellcheck.properties*.
   The value of the *dictionary.english* must be the full path to *english.0*
   Example:

   ```
   dictionary.english=/m1/voyager/{xxxdb}/tomcat/vwebv/spellcheck/WEB-INF/english.0
   ```

   B. Test the spellchecker by opening a browser and entering:
   http://your.library.domain/spellcheck/spellcheck?searchArg=legel
   The result should be something like this:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
     <spellcheck>
       <word text="legel">
         <suggestion>legal</suggestion>
         <suggestion>level</suggestion>
       </word>
     </spellcheck>
   ```

   C. The arguments you can send to the Spellcheck servlet are:
      - ***searchArg*** - This is required and is the word against which the dictionary will be checked.

        ```
        Example URL parameter: searchArg=civill+warr
        ```

      - ***dict*** - This is the dictionary to use. It is the word after the first period of the dictionary parameter in spellcheck.properties. The english.0 dictionary is used by default.
        For example, if in spellcheck.properties, there exists the following entry:
        dictionary.cats=/m1/voyager/xxxdb/tomcat/vwebv/spellcheck/WEB-INF/felines.0

        ```
        Example URL parameter: dict=cats
        ```

      - ***reload*** - Once a dictionary is loaded, it remains in use until a dictionary is reloaded. If entries are added to the dictionary currently in use, the servlet will not see the new entries until the dictionary is reloaded. If a new dictionary is desired, it must be specified with *dict* and then reloaded.

        ```
        Example URL parameter: reload=true
        ```

      - A more complete example of how dictionaries are loaded:

        When tomcat is first started, the dictionary is loaded by a user call to
        http://your.library.domain/spellcheck/spellcheck?searchArg=medisine
        Because no dictionary is specified, the default english dictionary was used.

        I then add *Supercalifragilisticexpialidocious* to the english dictionary. A search for *supercalifragilsticexpialidocious* (missing an *i*) will return no results until I call the servlet with *reload=true*
        http://your.library.domain/spellcheck/spellcheck?searchArg=supercalifragilsticexpialidocious&reload=true

Now I would like to check the spelling of *trés*, which I'm quite sure is French for *very*. To do this, I ask for a new dictionary with *dict=french&reload=true* like so:
http://your.library.domain/spellcheck/spellcheck?searchArg=trés&dict=french&reload=true
assuming I have *dictionary.french=/path/to/a/french_dictionary.0* in my spellcheck.properties file.

- When using these various parameters with WebVoyage, the parameters will be sent from the spellcheck.js JavaScript file. For example, if your user is searching subjects, you may only want to use a subject keyword dictionary that you created. You can parse the query string for *searchCode=SUBJ*, then change the dictionary using *dict=subject&reload=true* in the call to the Spellcheck servlet. Again, this assumes you have *dictionary.subject=/path /to/subject_keyword_dictionary.0* entered in your spellcheck.properties file.

Steps:

```
cd /m1/voyager/{xxxdb}/tomcat/vwebv/spellcheck/WEB-INF
```

Use your favorite code editor to edit spellcheck.properties as indicated above.

Test the spellchecker in a browser.

6. Configure WebVoyage to use the spellchecker by adding the code in **red** in the designated files:

A. In */m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/xsl/userTextConfigs/pageProperties.xml*, make the following changes (I have it all on one line in our file, but it runs off the page in printing, so it is split over 4 lines in the instructions):

```
<pageMsg errorCode="searchResults.noHits">Search resulted in no hits.
<span id='spellcheck_message'></span>
<span id="close_box" onclick="hide_spelling()">X</span>
<script type='text/javascript'>spellcheck();</script>
</pageMsg>
```

B. In */m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/xsl/pageTools/frameWork.xsl*, make the following changes: (This will load the JavaScript only if a search page is hit.)

```
<script type="text/javascript" src="{$jscript-loc}ajaxUtils.js"/>
<xsl:variable name="searchPages" select="'page.searchBasic page.searchAdvanced page.searchSubject page.searchAuthor' "/>
<xsl:if test="contains($searchPages,/page:page/@nameId)">
   <script type="text/javascript" src="{$jscript-loc}spellcheck.js"/>
</xsl:if>
```

C. In */m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/xsl/pageTools/tools.xsl*, make the following changes: (This lets us control the noHits area with JavaScript.)

```
<p class="{$messageClass}" id="{$messageClass}">
```

D. In */m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/css/searchPages.css*, add:

```
p.noHitsError
{
   background:#FFFF90 none repeat scroll 0%;
   border:1px solid red;
   color:#FF0000;
   font-weight:bold;
   padding:5px;
   text-align:center;
   width:300px;
   position:relative;
}
#close_box {position:absolute;right:1px;top:1px;border:1px solid red;cursor:pointer;}
```

E. Test your spellchecker by doing a WebVoyage Basic Search for *comonly mispelled wordz*

Files to change:

`/m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/xsl/userTextConfigs/pageProperties.xml`

`/m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/xsl/pageTools/frameWork.xsl`

`/m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/xsl/pageTools/tools.xsl`

`/m1/voyager/{xxxdb}/tomcat/vwebv/context/vwebv/ui/{skin}/css/searchPages.css`

Test your spellchecker by doing a WebVoyage Basic Search for comonly mispelled wordz.

7. Customize your dictionaries

Rush right over to Michael Doran's site to get create-dict.pl, a nice tool for creating custom dictionaries.

Copy everything from ***#!/m1/shared/bin/perl*** to ***exit(0);***.

Paste this into a new document using your favorite coding tool.

Change $username, $password, and $db_name to those used at your institution

Save it to */m1/voyager/**{xxxdb}**/sbin/create-dict.pl*. (This is where your newBooks.pl should reside.)

```
cd /m1/voyager/{xxxdb}/sbin

chmod 755 create-dict.pl

./create-dict.pl

mv /m1/voyager/{xxxdb}/tomcat/vwebv/spellcheck/WEB-INF/english.0 /m1/voyager/{xxxdb}/tomcat/vwebv/spellcheck/WEB-INF/english.0.bak

cp dictionary.2 /m1/voyager/{xxxdb}/tomcat/vwebv/spellcheck/WEB-INF/english.0
```

Open your web browser to *http://your.library.domain/spellcheck/spellcheck?searchArg=legel&reload=true* to load the new dictionary.

Disclaimer

Code and instructions for implementing a Spell Checker for Tomcat WebVoyage. Use at your own risk. TCC is not responsible for damage or problems caused by the use of this code. No warranties or guarantees of any kind are implied or assumed.

Version 2.4.8 by Jim Robinson
Tarrant County College
828 W. Harwood Rd.
Hurst, TX 76054-3299
(817) 515-6116
Email jim.robinson @ tccd.edu

"WebVoyage" is a trademark of Ex Libris™ LTD.
"Jazzy" is a Java Open Source Spell Checker

Purpose: Spellcheck servlet and JavaScript use Jazzy to check the spelling when no hits are found using WebVoyage 8.

No warranties or guarantees of any kind are implied. Use at your own risk. It works for us, so we thought we'd share... ;-)
By using this software the USER indicates that he or she has read, understood and and will comply with the following:
TCC hereby grants USER permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee, provided that:

1. the above copyright notice appears in all copies of the software and its documentation, or portions thereof, and
2. a full copy of this notice is included with the software and its documentation, or portions thereof, and
3. neither the software nor its documentation, nor portions thereof, is sold for profit. Any commercial sale or license of this software, copies of the software, its associated documentation and/or modifications of either is strictly prohibited without the prior consent of TCC.
4. Title to copyright to this software and its associated documentation shall at all times remain with TCC. No right is granted to use in advertising, publicity or otherwise any trademark, service mark, or the name of TCC.
5. This software and any associated documentation are provided "as is," and TCC MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING THOSE OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THAT USE OF THE SOFTWARE, MODIFICATIONS, OR ASSOCIATED DOCUMENTATION WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER INTELLECTUAL PROPERTY RIGHTS OF A THIRD PARTY. TCC, Tarrant County College, its Regents, officers, and employees shall not be liable under any circumstances for any direct, indirect, special, incidental, or consequential damages with respect to any claim by USER or any third party on account of or arising from the use, or inability to use, this software or its associated documentation, even if TCC has been advised of the possibility of those damages.

Purpose

This code was written to be used with the WebVoyage 8 product by Ex Libris™ LTD.™ When a WebVoyage 8 search receives no hits, the search criteria is sent to the Spellcheck servlet. The Spellcheck servlet will return an XML document in the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
   <spellcheck>
      <word text="misspelled word w/ diacritics removed">
         <suggestion>suggestion 1</suggestion>
         <suggestion>suggestion 2</suggestion>
      </word>
      <word text="misspelled word w/ diacritics removed">
         <suggestion>suggestion 1</suggestion>
         <suggestion>suggestion 2</suggestion>
         <suggestion>suggestion 3</suggestion>
      </word>
      <word text="correctly spelled word"></word>
      <errors>
         <error>error 1</error>
         <error>error 2</error>
      </errors>
   </spellcheck>
```

- The JavaScripts, java source, images, and html are enclosed in tomcat_spellcheck-2.4.8.zip or tomcat_spellcheck-2.4.8.tar.gz.
- You can see the spellchecker in action at http://library.tccd.edu/vwebv/searchBasic. Enter a misspelled word, such as *pikcle* or *"captian kook"*.

Requirements:

- This spellchecker uses a Java servlet. It must have an implementation of the Java Servlet technology to run, preferably the Apache Tomcat installation that runs your WebVoyage version 7.
- The spelling suggestion uses JavaScript. Browsers with JavaScript disabled will not be able to take advantage of this code, but it will not prevent them from using WebVoyage normally.

Changes

- v2.4.8 - (2011.07.25) Updated instructions for Voyager 8.0
- v2.4.7 - (2010.02.18) Fixed JavaScript to handle apostrophes (e.g. Amerika's Cup). Thanks to Tom Mendenhall at HSU Library for pointing out this glaring oversight.
- v2.4.6 - Updated instructions for Voyager 7.1.x
- v2.4.5 - (2009.04.29) Revised character encoding so diacritics would play (and display) nice. Added a close box to the No Hits. Thanks again to Michael Doran at University of Texas at Arlington for patiently troubleshooting and contributing code to the JavaScript and for writing the easy-to-use custom dictionary creation tool.
- v2.4.2 - (2009.04.16) Added character encoding so diacritics would play (and display) nice. Thanks to Michael Doran at University of Texas at Arlington for bringing this to my attention and for patiently troubleshooting.
- v2.3 - (2008-10-06) Revised spellcheck.js to convert words to be checked to lowercase because the spellcheck engine (Jazzy) ignores words in all uppercase. Thanks to Jamie Denman at University College Falmouth for bringing this to my attention.
- v2.2 - (2008.09.19) Revised and recompiled the *DictionaryMaker* and *DictionaryCombiner* programs to work with older versions of Java on Windows. Updated PATH information in the instructions.
- v2.1 - (2008.09.05) Revised the *Spellcheck* servlet. Correctly spelled words are no longer returned as suggestions in the XML document, only as an attribute to the <word> tag. Thanks to Andrew Brown at Swansea University for playing the role of guinea pig and discovering this less than useful 'feature'. ;-)
- v2.0 - (2008.09.02) WebVóyage 7 version utilizing servlets.
- v1.3 - (2007.08.14) Changes Global Keyword searches to Command Searches for use with the AND, OR and NOT Booleans in the suggestions.
- v1.2 - (2007.05.07) Added support for Simple and Combined Searches.
- v1.1 - (2006.10.01) Initial release using Perl and aspell.
- v1.0 - (2006.08.01) Initial release using PHP and aspell.
- v0.1 - (2004.08.01) Used Google's SOAP interface (available here). Developer keys are no longer available from Google.

Troubleshooting

- Make sure your paths are correct.
- spellcheck.properties must exist in /m1/voyager/{xxxdb}/tomcat/vwebv/spellcheck/WEB-INF/ and must point to a valid dictionary.
- Be sure to change all the {xxxdb} to your database.
- Be sure to change all the {skin} to your skin.
- **Use Firefox and install Firebug!**

To Do

- Fix character encoding for direct access to the spellcheck servlet. Currently, precomposed diacritics pose a problem that is resolved by passing it through the JavaScript.

Questions and comments can be sent to Jim Robinson at Tarrant County College. jim.robinson @ tccd.edu