SolrResin

Solr with Caucho Resin

- Solr runs fine with Resin, see the instructions in the generic Solr installation page

Logging

For information about controlling JDK Logging (aka: java.util logging) in Resin please consult the Resin docs... http://www.caucho.com/resin-3.0/config/log.xtp#log

Configuring Solr Home with JNDI

`<env-entry>` settings can be used in your resin.conf to specify the location of your Solr Home...

```
<web-app id="/solr" character-encoding="utf-8">
  <env-entry>
     <env-entry-name>solr/home</env-entry-name>
     <env-entry-type>java.lang.String</env-entry-type>
     <env-entry-value>/path/to/your/solr-home</env-entry-value>
  </env-entry>
</web-app>
```

Multiple Solr Webapps

Multiple Solr instances can be run on the same resin port using JNDI to specify the Solr Home, and the `archive-path` property to point both web-app instances at the wame war file...

```
<web-app id="/solr1" document-directory="webapps/solr1" archive-path="/path/to/the/solr.war" character-
encoding="utf-8">
  <env-entry>
     <env-entry-name>solr/home</env-entry-name>
     <env-entry-type>java.lang.String</env-entry-type>
     <env-entry-value>/path/to/your/solr-home</env-entry-value>
  </env-entry>
</web-app>

<web-app id="/solr2" document-directory="webapps/solr2" archive-path="/path/to/the/solr.war" character-
encoding="utf-8">
  <env-entry>
     <env-entry-name>solr/home</env-entry-name>
     <env-entry-type>java.lang.String</env-entry-type>
     <env-entry-value>/path/to/your/alteranate/solr-home</env-entry-value>
  </env-entry>
</web-app>
```

Restricting Updates (Security)

As discussed in SolrSecurity, whenever possible you should configure Resin to restrict access to any "Update" request handlers so that modifications to your index can only be made by the clients you expect. This can be donw by adding a `<security-constraint>` to the `<web-app>` configuration for Solr in your resin.conf...

```
<web-app id="/solr" document-directory="webapps/solr" character-encoding="utf-8">
 <security-constraint>
  <web-resource-collection>
    <url-pattern>/update/*</url-pattern>
  </web-resource-collection>
  <ip-constraint>
    <allow>127.0.0.1</allow>
  </ip-constraint>
 </security-constraint>
</web-app>
```

Loading Custom Plugins

In addition to using a `${solr.home}/lib` dir for including SolrPlugins, Resin has a general purpose mechanism for including arbitrary classes in the context class loader of an application like Solr. To do this, add a `<class-loader>` declaration to your `<web-app>` configuration in your resin.conf...

```
<web-app id="/solr" document-directory="webapps/solr" character-encoding="utf-8">
  <class-loader>
    <library-loader path="/path/to/your/lib/dir/containing/some/jars"/>
  </class-loader>
</web-app>
```

Resin's XML Parsing Problems

Note that many versions of Resin have shipped with an XML parser implementation that's a bit sketchy.

This typically shows up with you get an error during schema parsing.

There's a section near the beginning of the `web.xml` file, which (if uncommented) switches the XML parser to Xerces.

```
  <!-- Uncomment if you are trying to use a Resin version before 3.0.19.
    Their XML implementation isn't entirely compatible with Xerces.
    Below are the implementations to use with Sun's JVM.
  <system-property javax.xml.xpath.XPathFactory=
             "com.sun.org.apache.xpath.internal.jaxp.XPathFactoryImpl"/>
  <system-property javax.xml.parsers.DocumentBuilderFactory=
             "com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl"/>
  <system-property javax.xml.parsers.SAXParserFactory=
             "com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl"/>
   -->
```

Note that versions after 3.0.19 have also been reported as having issues with parsing of odd (but still legal) XML.