vufind

Software Documentation

Falvey Memorial Library
Villanova University

July 20, 2007

Introduction

VuFind is a modular software package designed for libraries by libraries who wish to disseminate large amounts of bibliographic information in a very intuitive and highly findable manner to their users. The software is built upon a framework that incorporates many open source projects - notably Apache Solr. Solr is a well regarded search engine that offers speed, scalability and customizability. By incorporating these applications, the **VuFind** software has the ability to offer an extremely scalable and fast search and retrieve system with a low upfront investment cost.

Table of Contents

Software Structure

- **data**
  The data directory is the directory that stores the individual marcxml records generated by the import scripts.

- **import**
  The import directory contains the import scripts to import the marcxml data. There are 2 scripts, one for the initial import and the second for the nightly syncing.

- **solr**
  The solr directory contains 2 open-source packages already pre-configured for the ViewFind application. Apache Tomcat is the widely adopted Java Servlet Engine that runs Apache Solr. To start or stop solr, use the solr/tomcat/bin/startup.sh and solr/tomcat/bin/shutdown.sh scripts respectively.

- **web**
  The web directory holds the web-base front end component of the ViewFind application.

  - **conf**
    The conf directory contains the config.ini file which is the global configuration file for the application.

  - **css**
    The css directory holds the css files that are related to the HTML templates. This directory and its files can be changed or removed if you change the HTML templates to fit your own look and feel.

  - **images**
    This directory is optional and contains the images for the web interface.

  - **interface**
    The interface directory contains 4 subdirectories:
    - **cache**
      The cache directory holds all of the cached template files. The directory must be writeable by the user that the web server is run by. For example if the user 'apache' is the user that runs the web server, than the directory should be owned by that user with full write privileges. If you are unsure about this give full write privileges by everyone to the directory.
    - **compile**
      The compile directory is very similar to the cache directory and needs the same privileges. It contains all of the "compiled" templates.
    - **plugins**
      The plugins directory contains all of the smarty template custom plugins. Currently a plugin that highlights the matching text in the search results resides in this directory.
    - **templates**
      The templates directory contains the global HTML templates for the application. There are currently 3 global HTML templates. A layout.tpl file that is the overall template, an error.tpl file that is the

error page template and a unavailable.tpl file that displays the "system is currently unavailable" page for when the system might be down for maintenance, etc.

- **log**
  The log directory contains the web application logs, searches, errors, etc.

- **services**
  The services directory contains the individual modules for the application. Each module will have its own **templates** directory as well as any other needed directories.

- **sys**
  The sys directory contains the system libraries that are needed for the application. Currently the User Interface, Apache SOLR and a Datagrid widget are all abstracted into their own libraries. The User Interface library extends the Smarty Template Engine library and the Datagrid widget extends the PEAR Structures_Datagrid library.

- **xsl**
  The xsl directory contains some global XSL files and may need to be filtered into the various xsl directories within the various modules in the services directory. The most important is the solr-convert.xsl file that converts all of the solr responses into a format that is used by the **VuFind** application

- **Action.php**
  This file is a php file that contains the Abstract Base Class for all of the actions in the entire system.

- **bookcover.php**
  This file is a script that generates a book cover image from the Amazon Web Services. It is an optional file if you so choose to use the AWS book cover images.

- **index.php**
  This file is very important since it is the main controller for the application. It can be renamed to fit the default index file defined by your web server. This file calls all actions defined by the URL and includes error handling as well as connecting to the MySQL database.

- **scripts.js**
  This file is optional, it contains various javascript functions for the web interface.

install
The install file is a bash script that currently only installs the necessary PEAR packages and the Smarty Template Library. This will be expanded in the future to help automate the installation process

LICENSE
The LICENSE file is a copy of the GPL license.

mysql.sql
The mysql.sql file is an import sql file for the mysql database.

README
The README file is the installation documentation for ViewFind.

ILS Driver System

The **VuFind** software was initially developed to be ILS agnostic as to work with any ILS system. The portion of the software that interacts with the ILS is based on a driver system. A driver, or a single php class file, must exist for the ILS of your choice. Within the web/services/Search directory there is a directory named Drivers. The Drivers directory contains the drivers for connecting to the appropriate ILS backend. To use an existing driver, you must edit the config.ini file found in the web/conf directory. If there is not a driver for your ILS, you can develop your own by creating a php class that implements the driver interface found in the Interface.php file. You can review the Voyager.php file as an example.

A driver must have the following functionality:

- getHolding
  A method that is responsible for retrieving holdings information by specifying a 001 field. The information returned should be Current Status, Current Location and Current Callnumber

- patronLogin
  This method accepts a username and password and returns a patron Id from the ILS if the authentication is successful.

- placeHold
  This method places a hold on an item using a 001 id and the patron ID

A driver would benefit from the following functionality:

- getNewItems
- getDepartments
- getInstructors
- getCourses
- findReserves

SQL Database

The **VuFind** application does incorporate a SQL Database, MySQL, for many important reasons. A SQL Database is a perfect system for data that changes frequently. While the core data in an OPAC does not change frequently, data such as user profiles and tags and comments on records do need to be managed by a SQL Database. MySQL is a fast and lightweight database and lends quite well to this application. Due to the design of the software, most any SQL database can be used in its place such as Oracle or PostgreSQL by simply changing the configuration options under ‘Database’ in the config.ini file.