Integration of VT ETD-db with Banner

James Volpe, Edward A. Fox
Digital Library Research Laboratory
Virginia Tech
Blacksburg, VA 24061, USA
{jvolpe, fox}@vt.edu

Gail McMillan
Digital Library and Archives
Virginia Tech
Blacksburg, VA 24061, USA
{gailmac}@vt.edu

Abstract

The Electronic Thesis and Dissertation database (ETD-db) was developed at Virginia Tech by Digital Library and Archives for the VT Graduate School and the Networked Digital Library of Theses and Dissertations (NDLTD). The software is freely available and over 100 universities worldwide have implemented the ETD-db system. One drawback of the system is the dependence on user keyed data. At Virginia Tech, like most other universities, there is an administrative database that could provide much of this information. The Banner Administrative System is the central administration system at Virginia Tech. Banner's underlying database software is from Oracle. This paper will demonstrate how the ETD-db can be seamlessly integrated with an Oracle database or more specifically the Banner Administrative System, to improve the integrity of the data for ETDs.

1. Introduction

The ETD-db system consists of a series of web pages to manage a collection of electronic theses and dissertations. There are various web pages that help different types of users submit, search, browse, catalog, maintain, and approve ETDs. Submitting, reviewing, approving, maintaining, searching and browsing are all functional processes that are required for the system. Some universities have been using the ETD-db system for almost ten years [1]. The feedback from a functional standpoint has been favorable [2].

Technically the system consists of Perl scripts for the web pages, a MySQL database, and a web server. Apache is the recommended web server but is not required. The software required for the system to run is widely used and free to download. The Perl scripts for the ETD-db system are also free to download for NDLTD members [3]. The implementation is straightforward and flexible enough to adapt to most universities. However, the system's flexibility comes at a cost. The submission process relies on data that is manually entered by the user. While most users do their best to accurately input their information, there can be data quality and integrity issues.

Virginia Tech's Banner system is responsible for managing the business processes and data for the entire university. Much of the data required in the ETD submission process resides in Banner. This study looked at the ETD submission process to see what data could be pulled from Banner. Although this integration is specific to Banner, this information can be applicable for integrating the ETD-db with another database, especially another Oracle database. Using data from Banner reduces the amount of data that is keyed in by the user, provides accurate data, and simplifies the ETD submission process.

The purpose of this study was to demonstrate a seamless integration of the ETD-db and Banner. One requirement for this integration was to use the existing toolset. Every attempt was made to only use Perl, MySQL, and the web (Apache) server in developing a solution. This requirement was met except Oracle

Instant Client was required for the Perl DBD-Oracle module. The last requirement was to maintain the same look and feel of the current ETD-db user interface.

2. ETD Submission Overview

The ETD-db consists of six modules. Each module supports a particular function. The six modules are: browse, search, submit, review, maintain, and withheld. This study only looks at the submit module. The submit module provides a way to electronically send an ETD to the graduate school for review. A
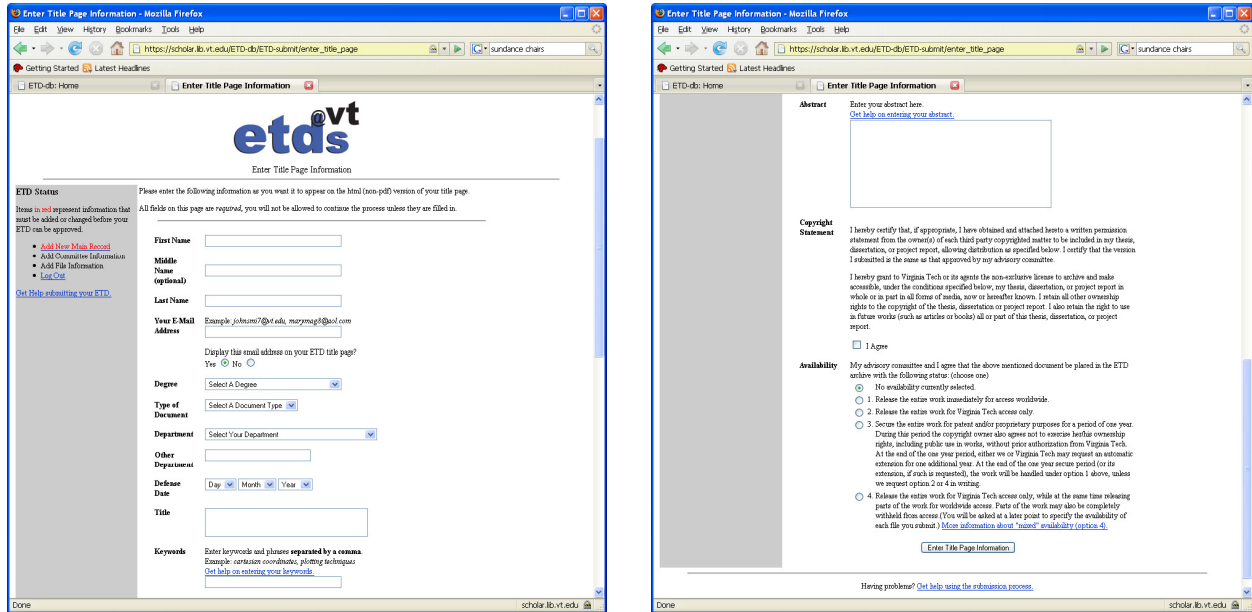

Figure 1 - Add New Main Record

screen shot of the initial submit page is shown in Figure 1. To get to this screen, a user logs in using their Virginia Tech assigned personal identification (PID) and password. This is also the same login that a user would use to access Banner. This is useful because the PID is all that is needed to get information from Banner. The submission process requires name, email, degree, document type, defense date, title, keywords, abstract, copyright agreement, and availability. After the 'Add New Main Record' a user must add their committee and then add their ETD file(s).

3. Banner Administrative System

The Banner system is the collection of central administrative systems and data at Virginia Tech. The system is composed of modules designed to support the processes and functions of a higher education institution. These modules are alumni/development, human resources, finance, and student and financial aid [4]. Banner is a product of Sungard Higher Education and is implemented by more than 900 institutions worldwide [5].

Virginia Tech is currently using Banner version 7.4. Banner runs on an Oracle 10g database. The database has built in data constraints to help maintain the integrity of the data. The ETD-db provides a few data constraints but not to the extent in Banner. In addition to the data constraints, Banner is the central database at Virginia Tech. Banner is the main source for administrative and academic information at Virginia Tech. The quality of the data provides an incentive to use Banner data instead of keyed in

data. Since Banner is used in over 900 institutions, other universities may be able to take advantage of this implementation too.

4. Connecting ETD-db to Banner

Before a connection to Banner's Oracle database can be accomplished, Oracle client software must be installed on the ETD-db server. Oracle Instant Client was installed to satisfy this requirement. Installing the full Oracle client is the other alternative for accessing an Oracle database. However, Oracle Instant Client is the better choice for this application. The Oracle client requires 460MB of disk space where Instant Client only requires 80MB. Oracle Instant Client is also easier to install. Installation is accomplished by unzipping a file that Oracle provides free of charge on their web site.

After access to Oracle is established, the DBD-Oracle Perl module must be installed. The module is needed for Perl scripts to be able to query Banner's Oracle database. Perl uses a Database Interface (DBI) module that is capable of communicating with multiple databases. The DBI is a simple interface that can locate and load Database Driver (DBD) modules [6]. Queries are sent to the DBI and the DBI passes them to the appropriate DBD module. The DBD returns the query results to the DBI which returns the results to the program. The ETD-db already uses the DBD-MySQL module to connect to the MySQL database. The DBD-Oracle module is simply another driver to support querying Oracle databases. The Perl syntax for querying an Oracle database is the same as querying a MySQL database. Therefore extending the ETD-db for querying an Oracle database is no different than querying a MySQL database. The DBD-Oracle module is available to download from CPAN [7].

A prototype was implemented to provide a proof of concept. First, a Pentium 4 PC was setup with the CentOS 4.5 Linux distribution. Second, the ETD-db system was installed. See Appendix A for installation notes on the ETD-db setup. Third, Oracle Instant Client and DBD-Oracle was installed. See Appendix B for installation notes on Oracle Instant Client and DBD-Oracle.

5. Modifying the ETD Submission Process

The current ETD submission process consists of three steps. The first step is adding the main record. In this step the user enters their name, email, degree, department, defense date, document type, title, keyword, abstract, copyright agreement, and document availability. The second step is to add their committee information. This involves entering each committee member's name, email, and role. The third and last step is to add the document. The first and second steps require user keyed data. There are HTML drop down lists to help minimize keying errors. However, there are text boxes that are susceptible to whatever is typed by the user.

A review of the submission process revealed that the data required in the first two steps could be provided by Banner. The user's name, email, degree, department, defense date, title, and committee information all reside in Banner. Since all this information can be pulled from Banner, the first two steps were combined into one step. This reduced the submission process to two steps, adding the main record followed by adding the document. With most of the data coming from Banner, adding the main record now consists of entering the document type, keywords, abstract, copyright agreement, and document availability. The last step of adding the document remains the same. An overview of the Perl scripts that were modified is provided in Appendix C.

6. Discussion

Integrating the ETD-db with Banner simplified the submission process while providing administrative data from Banner. The original submission process requires the user to fill out approximately twenty-five fields. The submission process integrated with Banner requires the user to fill out approximately ten fields. The integrated submission process focuses more on submitting a document and making a theses and dissertation available electronically. The integrated submission process supplies the data about the user so that the user only has to enter the document type, keywords, abstract, copyright agreement, and document availability. Once that information has been submitted the user uploads the document to the server and they are done.

One of the original goals for this study was to develop a module that other Banner schools using the ETD-db could download and implement. However half of the information pulled from Banner comes from tables developed at Virginia Tech. These tables were developed for the graduate school to handle degree requirements for graduation. These tables are responsible for the defense date, title and committee. Another field that is not supplied by Banner tables is department. Banner stores degree and major but not department. For example, at Virginia Tech, a computer science graduate student is looking to obtain a Master of Science degree by majoring in Computer Science and Applications. Using only Banner tables there is no way of knowing that a student majoring in Computer Science and Applications is in the Computer Science department. Banner supplied tables are only capable of providing a student's name, email, and degree. The database tables used for the integration are provided in Appendix D.

The requirement for using the existing toolset was met. The ETD-db and Banner integration requires the Oracle Instant Client install and DBD-Oracle Perl module. The DBD-Oracle module allowed for queries to be added to the existing ETD-db code. This helped with keeping the same look and feel for the user.

7. Conclusion

The ETD-db integration with Banner was accomplished with good results. The submission process was simplified and data integrity was improved by using data from Banner. While the exact implementation provided in this paper cannot be implemented at other schools, this does provide a blueprint on extending the ETD-db to other databases.

8. References

[1] NDLTD, http://www.ndltd.org/, 2007.

[2] McMillan, Gail. 8th International Symposium on Electronic Theses and Dissertations, University of New South Wales, Sydney, Australia. Sept. 28, 2005. http://scholar.lib.vt.edu/presentations/ETDsFullDisclosure.pdf

[3] ETD-db Home, http://scholar.lib.vt.edu/ETD-db/, 2007.

[4] Banner System at Virginia Tech, http://www.computing.vt.edu/administrative_systems/banner/, 2007.

[5] Sungard Higher Education, http://www.sungardhe.com/, 2007.

[6] A Short Guide to DBI, http://www.perl.com/pub/a/1999/10/DBI.html/, 2007

[7] Comprehensive Perl Archive Network, http://www.cpan.org/, 2007

Appendix A. Setting up ETD-DB test machine

1) Install CentOS-4.5-i386-binDVD

- ENTER to install in graphical mode
- perform media test or skip
- Welcome: click Next
- Language Section: accept defaults and click Next
- Keyboard Config: accept defaults and click Next
- Install Type: select Server and click Next
- Disk Partitioning: select Automatically partition and click Next
- Automatic Partitioning: select Remove all partitions, uncheck Review and click Next
- Warning: click Yes
- Boot Loader Config: accept defaults and click Next
- Network Config: check eth0, check eth1, and click Next
- Firewall Config: check Web Server, disable SELinux, and click Next
- Additional Lang Support: accept defaults and click Next
- Time Zone Selection: accept defaults and click Next
- Set Root Password: enter password and click Next
- Package Group Selection: make sure the following are checked and click Next
  - ✓ X Window System
  - ✓ GNOME Desktop Environment
  - ✓ Editiors
  - ✓ Graphical Internet
  - ✓ Text-based Internet
  - ✓ Office/Productivity but uncheck OpenOffice and planner apps in Details
  - ✓ Server Configuration Tools
  - ✓ Web Server
  - ✓ Window File Server
  - ✓ Development Tools
  - ✓ Administration Tools
  - ✓ Printing Support
- About to Install: click Next
- Remove disk and click Reboot

2) Boot fresh install

- Welcome: click Next
- License: agree and click Next
- Date and time: select Network Time Protocol tab, check Enable Network Time Protocol
  select clock.redhat.com from the Server drop down box
  click Next
- Display: click on Configure and select Dell 1400x1050 Laptop Display
  change resolution to 1400x1050

click Next
- System User: add user and password and click Next
- Sound Card: test sound and click Next
- Additional CDs: click Next
- Finish Setup: click Next

3) Log in under username created above

- if wireless, configure under Application:System Settings:Network
- open terminal and switch to "root"

  # yum update

- answer 'y' when prompted
- reboot just in case it is needed and log back in

4) Install Perl DBI

- login as 'root'
- check if perl-DBI is installed and if not install

  # rpm -q <package-name>
  # yum -y install <package-name>

5) Install etd-db required software (Apache, MySQL, and Perl modules)

- go to: http://scholar.lib.vt.edu/ETD-db/developer/
- start with requirements

  # yum -y install httpd php
  # yum -y install perl-DBI mysql perl-DBD-MySQL mysql-server php-mysql

- download and install CGI.pm and Tie-IxHash from link on requirements page
- create export folder of root and download ETD-db install file

  # curl http://scholar.lib.vt.edu/ETD-db/developer/download/ETD-db-v1.8.tar.gz >
     /tmp/ETD-db-v1.8.tar.gz
  # mkdir /export
  # cd /export
  # tar xzvf /tmp/ETD-db-v1.8.tar.gz

- create a symbolic link between /export/ETD-db-v1.8 and /export/ETD-db

> \# ln -s /export/ETD-db-v1.8/ /export/ETD-db

- create a symbolic link between /usr/bin/perl and /usr/local/bin/perl

> \# ln -s /usr/bin/perl /usr/local/bin/perl

- verify location of mysql binaries edit location if different
- verify root password to mysql
- there is an error with a filename to a script, use command below to fix

> \# mv /export/setup-ETD-db/create_etd_available_tables.mysql /export/setup-ETD-db/create_etd_available_tables.sql

- run create_everything.pl script
- add the following to /etc/httpd/conf/httpd.conf

```
#
# additional directories for ETD-db (must be added before Alias for ETD-db)
# ScriptAlias /ETD-db/ETD-submit/ /export/ETD-db/cgi-bin/ETD-submit/
ScriptAlias /ETD-db/ETD-review/ /export/ETD-db/cgi-bin/ETD-review/
ScriptAlias /ETD-db/ETD-withheld/ /export/ETD-db/cgi-bin/ETD-withheld/
ScriptAlias /ETD-db/ETD-maint/ /export/ETD-db/cgi-bin/ETD-maint/ ScriptAlias /ETD-db/ETD-catalog/ /export/ETD-db/cgi-bin/ETD-catalog/
ScriptAlias /ETD-db/ETD-browse/ /export/ETD-db/cgi-bin/ETD-browse/
ScriptAlias /ETD-db/ETD-search/ /export/ETD-db/cgi-bin/ETD-search/

Alias /ETD-db   /export/ETD-db/web_root

<Directory /export/ETD-db/cgi-bin/ETD-review>
  AllowOverride None
  AuthType Basic
  AuthUserFile /etc/httpd/conf/.htpasswd
  AuthName review
  order deny,allow
  deny from all
  require user etd-admin etd-grad
  satisfy any
</Directory>

<Directory /export/ETD-db/cgi-bin/ETD-catalog>
  AllowOverride None
  AuthType Basic
  AuthUserFile /etc/httpd/conf/.htpasswd
  AuthName catalog
  order deny,allow
```

```
    deny from all
    require user etd-admin etd-grad etd-catalog
    satisfy any
</Directory>

<Directory /export/ETD-db/cgi-bin/ETD-withheld>
    AllowOverride None
    AuthType Basic
    AuthUserFile /etc/httpd/conf/.htpasswd
    AuthName withheld
    order deny,allow
    deny from all
    require user etd-admin etd-grad
    satisfy any
</Directory>

<Directory /export/ETD-db/cgi-bin/ETD-maint>
    AllowOverride None
    AuthType Basic
    AuthUserFile /etc/httpd/conf/.htpasswd
    AuthName maint
    order deny,allow
    deny from all
    require user etd-admin etd-grad
    satisfy any
</Directory>

# directory the actual author-submitted content lives in
Alias /theses    /theses_1

# security settings for campus only ETDs, including campus only files
# that are part of mixed access ETDs <Directory /theses_1/available/*/restricted>
    AllowOverride None
    AuthType Basic
    AuthUserFile /etc/httpd/conf/.htpasswd
    AuthName restricted
    order deny,allow
    deny from all
    #allow from 128.173 .vt.edu
    #require user etd-admin etd-grad etd-catalog
    ErrorDocument 401 /theses/vt-only.html
    ErrorDocument 403 /theses/vt-only.html
    satisfy any
</Directory>

# security settings for the ETDs that are entirely withheld from access
```

```
# (i.e. patent restricted)
<Directory /theses_1/withheld>
  AllowOverride None
  AuthType Basic
  AuthUserFile /etc/httpd/conf/.htpasswd
  AuthName withheld
  order deny,allow
  deny from all
  require user etd-admin etd-grad etd-catalog
  satisfy any
</Directory>

# security settings for the patent restricted files within mixed access ETDs <Directory /theses_1/available/*/withheld>
  AllowOverride None
  AuthType Basic
  AuthUserFile /etc/httpd/conf/.htpasswd
  AuthName withheld
  order deny,allow
  deny from all
  require user etd-admin etd-grad etd-catalog
  satisfy any
</Directory>
```

- add users from all the 'required user' lines above using htpasswd utility

  ```
  #   htpasswd -c /etc/httpd/conf/.htpasswd etd-admin
  #   htpasswd /etc/httpd/conf/.htpasswd etd-grad
  #   htpasswd /etc/httpd/conf/.htpasswd etd-catalog
  ```

- set up the necessary directories to store your actual ETDs

  ```
  #   mkdir -p /theses_1/submitted
  #   mkdir -p /theses_1/available
  #   mkdir -p /theses_1/withheld
  #   mkdir -p /theses_1/submitted/trash
  ```

- set the cgi-bin directories to 755

  ```
  #   chmod 755 /export/ETD-db/cgi-bin/ETD-browse/*
  #   chmod 755 /export/ETD-db/cgi-bin/ETD-catalog/*
  #   chmod 755 /export/ETD-db/cgi-bin/ETD-maint/*
  #   chmod 755 /export/ETD-db/cgi-bin/ETD-review/*
  #   chmod 755 /export/ETD-db/cgi-bin/ETD-search/*
  #   chmod 755 /export/ETD-db/cgi-bin/ETD-withheld/*
  ```

6) Copy certificate file to connect to LDAP server.

- login as root and do:

\#   yum install perl-IO-Socket-SSL
\#   mkdir -p /usr/local/ssl/certs
\#   curl https://vtmwra.eprov.iad.vt.edu/cacert/cachain.pem >
       /usr/local/ssl/certs/ed_auth_cert.pem

Appendix B. Setting up ETD-db machine with Oracle Instant Client:

1) Install and setup Oracle Database 10g Instant Client

- go to: http://www.oracle.com/technology/tech/oci/instantclient/index.html
- download the latest basic, sdk and sql*plus instant client rpms
- install the rpms

    #   rpm -i oracle-instantclient-basic-10.2.0.1-1.i386.rpm
    #   rpm -i oracle-instantclient-devel-10.2.0.1-1.i386.rpm
    #   rpm -i --replacefiles oracle-instantclient-sqlplus-10.2.0.1-1.i386.rpm

- get tnsnames.ora and sqlnet.ora from DBMS development server

    #   cd /usr/lib/oracle
    #   sftp username@db.vt.edu
    #   sftp> cd /apps/dbms/oracle/product/10.2.0.2/network/admin
    #   sftp> get tnsnames.ora
    #   sftp> quit

- set LD_LIBRARY_PATH and TNS_ADMIN environment variables

    #   export LD_LIBRARY_PATH=/usr/lib/oracle/10.2.0.1/client/lib
    #   export TNS_ADMIN=/usr/lib/oracle

- check if sqlplus can logon to development server

    #   sqlplus username@develop

2) Install Perl DBI and DBD::Oracle

- login as 'root'
- check if perl-DBI is installed and if not install

    #   rpm -q <package-name>
    #   yum -y install <package-name>

- get DBD-Oracle from www.cpan.org

    #   cd /tmp
    #   curl http://mirror.cc.vt.edu/pub/projects/CPAN/modules/by-
        authors/id/P/PY/PYTHIAN/DBD-Oracle-1.19.tar.gz > DBD-Oracle-1.19.tar.gz

- follow the README and README.client documentation in the DBD-Oracle
- need ORACLE_HOME, ORACLE_SID, ORACLE_USERID and LD_LIBRARY_PATH
  environment variables defined

    #   export ORACLE_HOME=/usr/lib/oracle/10.2.0.1/client/lib

```
#   export ORACLE_SID=DVLP
#   export ORACLE_USERID=username/pasword@instance
#   export LD_LIBRARY_PATH=/usr/lib/oracle/10.2.0.1/client/lib
#   export TNS_ADMIN=/usr/lib/oracle
```

- run make files to install DBD-Oracle (if problems do 'make realclean' and start over)

```
#   perl Makefile.PL
#   make
#   make test       ** test 26exe_array.t fails but that is okay
#   make install
```

- when setting up accounts to access VT databases make sure LD_LIBRARY_PATH and TNS_ADMIN environment variables are defined

3) Set environment variables for web users to access oracle

For the Apache web server, users come in under the apache user. In order for the apache user to use the Oracle client the apache user needs to be a member of the oinstall group. After that, add the following to /etc/httpd/conf/httpd.conf to set environment variables for the apache user:

SetEnv LD_LIBRARY_PATH /usr/lib/oracle/10.2.0.1/client/lib
SetEnv TNS_ADMIN /usr/lib/oracle

4) Make modifications to ETD-db software . . .

Appendix C. Modifications to ETD-db files

File path: /etd-db/cgi-bin/ETD-submit/	
File	*Description*
etd-submit-lib.pl	Added Oracle DBI connect variables and removed the add committee links
enter_title_page	Added Banner queries to get student's: • name • email • degree and major • department • defense date • defense title • plan of study approval
add_etd	Added Banner queries to get student's: • advisor names • advisor emails • advisor committee roles
manage_committee	Removed add committee links
modify_etd	Added code to translate defense date to MySQL

File path: /etd-db/language/english/	
File	*Description*
submit_messages.pl	Modified the html forms text
submit_errors.pl	Added Banner DBI error handling messages

Appendix D. Banner Tables Used in the ETD Submission Process

Banner Developed Tables	
Banner Table	*Data Field*
spriden	student name
spbpers	suffix for student name
sgbstdn	degree and major
stvdegc	degree description
stvmajr	major description
goremal	student email

Virginia Tech Developed Tables	
Banner Table	*Data Field*
szvdegc	degree long name
szvmajr	major long name
gzbtpid	student pidm
szrency	department
szrgplm	defense date
szrgthe	defense title
szbgpln	plan of study approval
szrgcom	committee