Pubsearch 0.69 Unix Installation Guide

Danny Yoo (dyoo@acoma.stanford.edu)

1 System Requirements

This installation document assumes that the following software are already installed and running:

- A Java servlet container that implements the Java Servlet 2.3 and JSP 1.2 API's. For the examples below, we will use Apache Tomcat 4. (`http://jakarta.apache.org/tomcat`).
- MySQL 4.0 [1] RDBMS. (`http://www.mysql.com`)
- Perl. We recommend any release of Perl ≥ 5.6.1. If Perl is not installed, please talk to your system adminstrator.

2 Introduction

This document shows how to install *Pubsearch* on a Unix system. By the end of this guide, you should have a running version of *Pubsearch*.

This release of *Pubsearch* can be downloaded from:

`http://pubsearch.org/releases/pubsearch-0.68.tar.gz`.

If you have problems during the install, please contact the `gmod-pubsearch-dv` developers mailing list at:

`http://lists.sourceforge.net/lists/listinfo/gmod-pubsearch-dv`

with your questions.

For the examples in this guide, we will use `pubsearchdb` [2] as the name of the database, and `foobar.org` as the hostname where *Pubsearch* will be installed. We will also assume that Tomcat is installed under `/usr/local/jakarta-tomcat`.

[1] The database actually does not need to be run from the same server as *Pubsearch* itself; adjust the `HOSTNAME` configuration parameter in Section 5.1.

[2] Before you continue, you may want to set up a dedicated user in MySQL for *Pubsearch*. The documentation at `http://www.mysql.com/doc/en/User_Account_Management.html` explains how to create a new user.

3 Untar the web application

Untar `pubsearch-0.68.tar.gz` from the `webapp` root of the servlet container. In a typical Tomcat installation, this path is the `webapps` directory of Tomcat. (`jakarta-tomcat/webapps`).

```
$ cd /usr/local/jakarta-tomcat/webapps
$ wget http://pubsearch.org/releases/pubsearch-0.68.tar.gz
$ tar xzvf pubsearch-0.68.tar
```

This should create a subdirectory called `pubsearch`. Note that `GNU tar` is required to untar the archive, since some of the paths in the archive are more than 255 characters.

4 Configure the Database

4.1 Creating a New Database

We must first create a new database that *Pubsearch* will use to store data. If you are the MySQL root user, you can either do this by executing:

```
$ mysqladmin create pubsearchdb
```

at your command prompt shell, or

```
mysql> create database pubsearchdb;
```

from the MySQL interactive client.

Once the database is created, we must apply a database schema on it. There are two ways of doing this.

- Loading the sample database. This is the production database for Arabidopsis literature.
- Starting from scratch. A minimal database.

Let us first show how to load an existing sample database.

4.2 Loading the Sample Database

We have packaged[3] our database so that others can quickly get up and running.

The following commands should install the "Arabidopsis" data set into the database.

```
$ wget http://pubsearch.org/releases/pubsearch-sample-database-0.68.sql.bz2
$ bzcat pubsearch-sample-database-0.68.sql.bz2 | mysql pubsearchdb
```

[3] We use the `bzip2` utility to compress the database since it's a bit large; as of this writing, it's about 60MB compressed.

4.3 Starting from scratch

If you wish to start with an minimal database, here are the steps for setting it up.

- Apply the schema to the database.

  Within the *Pubsearch* web application directory, there is a schema file called `data/schema.mysql`. We will apply it to the newly created database:

  ```
  $ mysql pubsearchdb < data/schema.mysql
  ```

- Insert the minimal schema support data. Once the schema is installed, we need to apply a small support schema.

  ```
  $ mysql pubsearchdb < data/schema-support.mysql
  ```

The resulting database is ready to be populated with bulk data. See Section 7.1 to see how to load articles and terms into the system.

4.4 Checking the Database Contents

Once the sample dataset is loaded, we should check it and see that the data has loaded correctly. We can execute the following from the MySQL interactive client.

```
mysql> use pubsearchdb
mysql> show tables;
+-----------------------------+
| Tables_in_pub2              |
+-----------------------------+
| pub_alias                   |
| pub_allele                  |
| pub_allele_germplasm        |
| pub_analysisreference       |
| pub_annotationtask          |

[ ... text cut ]
```

If you haven't run into an error while doing this, then the database has been configured successfully.

5 Install the Web Application

Now that the database is primed, we can now enable the web application that uses *Pubsearch*.

5.1 Editing the configuration file

Pubsearch uses a configuration file pubsearch/WEB-INF/classes/pub/config/program.properties that needs to be modified before *Pubsearch* will work. This file must be modified to use the correct database that was initialized in Section 4.2.

Here is a sample of the first few lines of the file:

```
## Configuration file for Pubsearch
## Are we on a "test" environment?  This affects the header, and also
## may turn on additional debugging logs.
pub.is_test_environment=1
## The user authorized to connect to this database.
pub.database_username=$USERNAME
## The user's password for the database.
pub.database_password=$PASSWORD
## The JDBC connection string used to create new connections.
pub.database_connection_string=jdbc:mysql://$HOSTNAME/$DATABASE
## Directory where Pub can store auxillary data
pub.aux_data_dir=$INSTALL_TARGET/var
## Where the full text article PDF's live on the system.
pub.pdf_document_base = $INSTALL_TARGET/var/pdfs
```

(There are more lines near the bottom, meant for the Java developers; you should not need to edit them.)

It is important to modify program.properties with the correct values for the DATABASE, HOSTNAME, USERNAME, and PASSWORD appropriate to your MySQL database that you have just installed. Also, fill in INSTALL_TARGET with the absolute directory path where *Pubsearch* is installed.

Here is an example set of values for program.properties:

```
pub.is_test_environment=1
pub.database_username=janedoe
pub.database_password=rosebud
pub.database_connection_string=jdbc:mysql://foobar.org/pubsearchdb
pub.aux_data_dir=/usr/local/jakarta-tomcat/webapps/pubsearch/var
```

This ends the configuration of *Pubsearch*; now we must test it.

6 Test the Installation

We should now test that *Pubsearch* can access the database.

6.1 Testing the database connections

Go into the web application directory (i.e. `/usr/localjakarta-tomcat/webapps/pubsearch`) and enter the command

```
bin/test_database_connection.pl.
```

For example:

```
$ cd /usr/local/jakarta-tomcat/webapps/pubsearch
$ bin/test_database_connection.pl
```

You should see the message `Database connection looks good`.

But if the test fails, go back and check your configuration in

```
pubsearch/WEB-INF/classes/pub/config/program.properties
```

again: *Pubsearch* must be able to see your database. Otherwise it will stall trying to connect to it when it is brought up as a web application.

6.2 Testing the web application

Restart your servlet container. Tomcat can be restarted with the following commands:

```
$ cd /usr/local/jakarta-tomcat/bin
$ ./shutdown.sh
Using CATALINA_BASE:   /usr/local/jakarta-tomcat
Using CATALINA_HOME:   /usr/local/jakarta-tomcat
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat/temp
Using JAVA_HOME:       /usr
$ ./startup.sh
Using CATALINA_BASE:   /usr/local/jakarta-tomcat
Using CATALINA_HOME:   /usr/local/jakarta-tomcat
Using CATALINA_TMPDIR: /usr/local/jakarta-tomcat/temp
Using JAVA_HOME:       /usr
```

If everything has gone well, then *Pubsearch* should come up and be viewable from your web browser. Assuming default settings on Tomcat, the URL will be

```
http://foobar.org:8080/pubsearch/
```

Tomcat compiles JSP pages on the fly, so it may take a while for the page to first show up; subsequent reloads should be more responsive.

7 Loading custom data

Once the system is running, we can now start loading in data from different sources. *Pubsearch* contains several command-line utilities, as noted in the README. These utilities are located in `pubsearch/bin`. This section will show to to use some of these utilities.

7.1 Adding new curator users

Pubsearch requires login for certain sections of the web application. To add a new curator user, we provide the command line utility `pubsearch/bin/add_curator.pl`. Here is an example:

```
$ bin/add_curator.pl johndoe rosebud
```

7.2 Bulk loading articles

We include an command-line article bulk loader that talks to the Pubmed (`http://www.ncbi.nlm.nih.gov/entrez/query.fcgi`) and Agricola (`http://agricola.nal.usda.gov/`) databases. The utility, `fetch_articles.pl`, contacts both databases with a query and a date range, and extracts articles into the *Pubsearch* database.

For example:

```
$ perl bin/fetch_articles.pl arabidopsis 2004/01/01 2004/02/01
141 articles loaded.
```

7.3 Bulk loading terms

Another bulk loader is the Gene Ontology (`http://www.geneontology.org/`) bulk term loader. This loader reads GO XML files. Here is an example load, using the plant GOSLIM ontology:

```
$ bin/bulk_load_terms.pl data/test/goslim_plant.2003.xml
INFO  pub.db.command.AddTerm.execute(AddTerm.java:139) inserted term 123891
INFO  pub.db.command.AddTerm.execute(AddTerm.java:139) inserted term 123892
INFO  pub.db.command.AddTerm.execute(AddTerm.java:139) inserted term 123893
... [text cut]
```

The Gene Ontology Consortium provides a larger ontology at:

`http://www.geneontology.org/index.shtml#downloads`.

7.4 Generating hits

"Hit" associations between articles and terms are not generated on the fly, but are precomputed through an exact keyword matching against the title and abstract of each article.

```
$ bin/generate_hits.pl
```

This process takes a considerable amount of time, and should be scheduled during an off-hour. [fixme: change algorithm to something more efficient so we don't have to warn about this.]

7.5 Indexing full text

Articles in the database can be indexed through the Lucene full-text indexer.

```
$ bin/index_full_text.pl
```

This, too, may take some time, depending on the size of the document collection. This preprocessing step writes a set of indices into the `pub.aux_data_dir` defined in Section 5.1.

8 Troubleshooting the Installation

Here are some problems that may occur during the installation, and possible solutions to address those problems.

8.1 Untarring the file `pubsearch-newest.tar.gz` fails with the error `tar: directory checksum error!`

Some of the path names in *Pubsearch* are longer than 100 characters, and the native `tar` utility on certain Unixes can't handle long path names. Use `GNU tar` from:

```
http://gnu.org
```

to untar the archive.