

How to Install CAT

You will need

- Apache Tomcat (ours is 6.0)
- MySQL (ours is 5.1)
- We use Apache Ant to manage compiling and copying files into Tomcat, using Eclipse. These directions assume a similar development environment but can be adapted for other tools such as NetBeans.
- a CAS server at your institution for staff to log in with their institutional accounts (ours are called NSIDs).
- (optional) LDAP to associate instructor NSIDs with course offerings.
- We assume that the institution's course offerings can be retrieved from an XML web service.

Instructions

1. In Eclipse, choose New ---> "Dynamic Web Project". (Might have to dig for this project type under Other --> Web --> Dynamic Web Project).
2. In the Project Name, type "cat". For "Source folders on build path", delete "src" because we'll be getting our own "src" folder from the Subversion checkout.
3. For "Content directory" change this to "web", again to match the Subversion checkout.
4. Now right-click on the new project named "cat" and choose Team --> Share Project --> then pick SVN, then type in the repo's URL (<http://ulcf1.usask.ca:8081/svn/cat/trunk/cat/>). Next you might get a warning saying the specified folder already exists in the repo. Say YES and then the checkout folder will override the blank ones you created, and they will be connected to the repo.

Most SVN accounts in our repository will have "read" access to the trunk and "read+write" access to cat/branch/ for sharing enhancements. Within Eclipse you should now have a 'cat' project folder that contains subfolders: conf, docs, lib, src, web, and build.xml.

5. In MySQL, create a new database and user with the following commands (replacing the user-id and password as desired). Our table collations are set to case **insensitive**.

```
CREATE DATABASE currimap; GRANT ALL PRIVILEGES ON currimap.* TO 'ca-  
tuser'@'%' IDENTIFIED BY 'mypwd'; FLUSH PRIVILEGES;
```

Next, import the starter database from the SQL file in docs/createStructureWithInitialData.sql. This will populate your currimap database with all necessary tables as well as sample fictional data.

Note: Later, when your CAT user interface is up and running, you can add more courses under Program Admin --> B.Sc. 4 year in Basket Weaving (or other program) --> Add a

course. Please be sure to associate this new course with an “organization”/department. A list of departments/colleges from the University of Saskatchewan is provided, but these can be edited to reflect the departments and colleges at your institution. To associate a course with an organization (e.g., Basket weaving) as a system admin, edit the organization by either:

- in system admin tab, click "edit/add organization", select the organization (e.g., "Basket weaving") from the drop down menu. --> in the popup window select the relevant course code and click the checkmark next to the course(s), click "Save Courses for Organization" then "save organization" and "close". (see page 32 of the user guide).
- in Program Admin tab --> click the "edit" icon next to the organization such as "Basket Weaving (Demo)" --> in the popup window select the relevant course code and click the checkmark next to the course(s), click "Save Courses for Organization" then "save organization" and "close".

CAT User guide available at the bottom of: <http://www.usask.ca/gmcte/CAT>

When uploading courses through the tables or course import, then courses should be associated automatically or through the tables with a particular organization (e.g., MATH courses with Mathematics department).

6. Within the new database, add a new record in `system_admin` table. Modifying the SQL query below should work -- simply replace appropriate values for the “name”, “first_name” and “last_name” fields. The value in the “name” field should be a valid CAS username (i.e. an NSID) - probably the initial NSID belonging to the developer. ;)

```
INSERT INTO system_admin
(id,name,type,created_userid,type_display,first_name,last_name, created_on)
VALUES(1,'myNSID','Userid','myNSID','Persons','Steph','Frost','2013-01-01 12:00:00');
```

The “type” field distinguishes whether this is an individual person (Userid) or if the permission is for an LDAP group (LDAP). The “created_userid” records the NSID of the person who granted the permission. The “type_display” can be “Persons” or “Departments”. Please replace the “created_on” value with the current date.

7. Back in Eclipse, right click on the “src” folder that now has checked out code in it, and choose Build Path --> “Use as source folder”. Also do this with the “lib” folder.
8. In `context.xml`, enter the database URL, username and password as created in step 5. (This will be used by Tomcat.)
9. In `database.properties`, again enter the database name, URL, userid, password and driver. (This will be used within the CAT java code. This properties file was created before we started using Hibernate and can probably be phased out.)
10. In `hibernate.cfg.xml` lines 9-12, enter the database URL, username, password and driver.

11. In `currimap.properties`, specify a directory to store temporary files. Set `ldap.enabled=N` if you will not be using LDAP to match instructorIDs with courseIDs.
12. In `ldapuser.properties`, there are credentials for the LDAP as well as the course offerings web services. The LDAP is used to determine the courses that each CAS user is an instructor for. If you are using LDAP at your institution, you'll need to adapt the Java class, `ca.usask.gmcte.ocd.ldap.LdapConnection.java`. LDAP can also be used to query a user's full name, department and other data given the userid (NSID).
13. Also in `ldapuser.properties` there are two web service addresses. The `departmentSourceURL` assumes that this URL will provide a list of departments at your institution. For example, at the U of S, that URL will return output like this:

```
department=Computer Science
department=Accounting
department=Agricultural Economics
department=Anatomy & Cell Biology
```

The `classesForDeptURL` is assumed to return an XML file with your institution's course offerings. Note that the "`<dept>`" and the "`<term>`" will be replaced in the URL with actual department names and terms. An example of a term is 201209 where the first 4 characters are the year and the last two characters are the month that the term begins, i.e. September 2012. If your institution has a different web service then the Java class you'll need to adapt is `ca.usask.gmcte.currimap.action.DataImporter`. For example, below we show a sample result when visiting

https://servername.yourdomain.edu/api.jsp?command=xml_getallclasses&dept=biology&term=201209

```
<departments>
  <department>
    <name>biology</name>
    <courses>
      <course>
        <crn>81663</crn>
        <term>201209</term>
        <title>The Living Earth</title>
        <subject>BIOL</subject>
        <classnum>107</classnum>
        <section>01</section>
        <numstudents>0</numstudents>
      </course>
      <!-- snipped for brevity -->
    </courses>
  </department>
</departments>
```

14. In [web.xml](#), on line 23 enter the name of the server upon which you are installing CAT. This is to tell CAS where to send the user after they successfully authenticate, so we want to send them to your CAT server. If you are installing CAT on a development machine, you would probably set this to localhost like this:

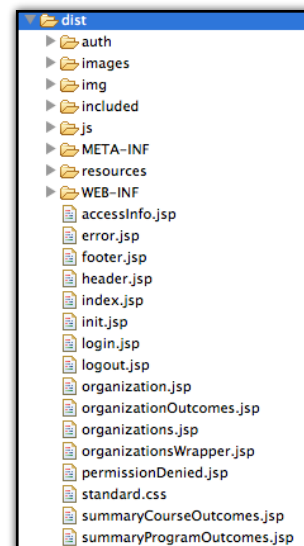
<param-value>localhost:8080</param-value>

15. In log4j.properties on line 9 enter a directory for the CAT log files, such as /Users/myuser/logs/cat.
16. Right click on the project folder “cat” and choose Properties. Next, go to Java Build Path, then the Libraries tab. Click “Add JARs” and individually add each Jar file that is in the lib folder. (there’s quite a few of them, sorry, hehe!). Click OK.
17. Right click on the project folder “cat” and choose Properties and go into Deployment Assembly. You will likely see that the source folder, “/web” is associated with the Deploy Path, “/”. We will have to add a few more mappings. Map “/src” to “WEB-INF/classes” by clicking Add --> Folder and navigate to “src”, click Finish, then double-click the cell under the “Deploy Path” column and make sure it is “WEB-INF/classes”. Similarly, map “/lib” to “WEB-INF/lib”. (For some reason, my /lib was mapped to WEB-INF/classes not WEB-INF/lib. so I changed it to lib) I also had to map the library subfolders too, ex “lib/log4j” to “WEB-INF/lib” and “lib/cas” to “WEB-INF/lib”, etc..
18. Also, map your conf folder (ex. /conf/example.yourdomain.edu to “WEB-INF/classes”). If this doesn’t work, you might have to physically move the [web.xml](#) file into “web/WEB-INF”, move context.xml into “web/META-INF” and move everything else into “WEB-INF/classes”.
19. Next, go under the Servers tab.

If you don’t have it, it’s under Window --> Show View --> Servers. Now if you don’t have any servers set up yet, you’ll see a link to Define a new server from the new server wizard. Click that and choose a Tomcat server, and browse to wherever you have Tomcat installed on your machine. (If you don’t have one yet, simply download and extract anywhere on your computer, such as /Users/myusername/bin).

If you already have the Tomcat setup under the Servers tab, just right-click on the Tomcat and choose “Add and Remove”.

20. Next, you can move any of your Dynamic Web Projects from the “Available” column to the “Configured” column. Add “cat” over to the right and click Finish.



21. Now under Servers you should be able to right-click Tomcat and choose Start. Then click the little globe icon in the toolbar (“Open Web Browser”) and type in <http://localhost:8080/cat>.
22. Hopefully, you should now see the CAT application running. Now that we have the development environment set up, we can go through the steps of creating a WAR file so you can deploy CAT to your production server.
23. Add the build file to the Eclipse Ant window (Window --> Show View --> Ant) by clicking on the “Add buildfiles” button and choosing cat/build.xml.

24. In build.xml, edit the location of 'conf' folder line 11. We have set the default location to the sample conf folder in cat/conf/example.yourdomain.edu. You may wish to create your own conf folder by copying the example provided and renaming it, and performing the following steps on your copy.
25. Within the Eclipse Ant window, in the Ant window double click 'dist'. This should create a new folder, 'cat/dist' with contents similar to the image below. This will also create a war file (cat/cat.war).
26. In build.properties starting on line 27, enter the information about your Tomcat server where CAT will be running. This allows ANT to automatically send the WAR file to Tomcat and re-start the web app. Note, the Tomcat userid and password are usually configured in Tomcat's conf/tomcat-users.xml file. Example:

```
<tomcat-users>
  <role rolename="manager">
    <user username="admin" password="pass" roles="manager">
</tomcat-users>
```

27. To deploy to your Tomcat, double-click 'deploy'. Alternatively, you could manually copy the war file to the Tomcat webapps folder and manually restart the Tomcat from the command line with Tomcat's bin/shutdown.sh and startup.sh.
28. Now you should be able to open a web browser to <http://localhost:8080/cat> and be prompted with your institution's CAS prompt. After logging in, you will see the CAT main screen. You can begin entering settings under the "System Admin" link.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](http://creativecommons.org/licenses/by-sa/3.0/).