APPENDIX XXIX – USER MANUAL FOR SHREDDER

A1 USER MANUAL WEB-BASED SHREDDER

The online shredder is accessible from http://researchworks.cs.athabascau.ca/ web site. The user needs to create an account to login in order to use the shredder or the user can use the command line version of the shredder. Both these shredders have the same functionalities. In this section we will discuss how to use these shredders.

A1.1 REGISTER AND LOGIN

Under "Shredders" menu option (see Figure B-1), click on the "PHP Online" sub-menu option to get the login screen (see Figure B-2).

Figure B-1: Main Menu of CAMM with Submenu of Shredders

A new user, who does not have a Login ID (i.e., account in the database), and wants to use the shredder must click "**click here**" as in Figure B-2 to complete the registration. If the user already has an account, then the user can login and use the shredder.

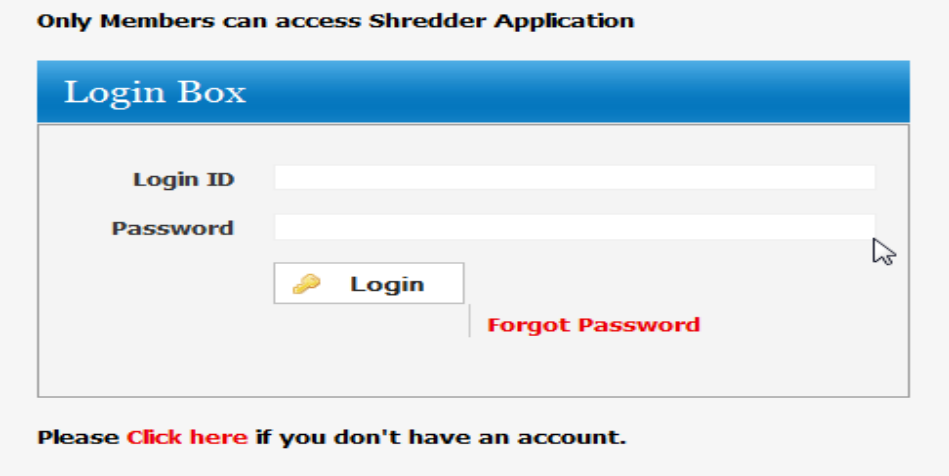


Figure B-2: Registration and Login Screen

Figure B-3 shows the Registration as well as login options for the user. During registration process, there are a couple of security checks such as the password must be

at least 6 characters long otherwise the user will get the warning as shown in Figure B-4.

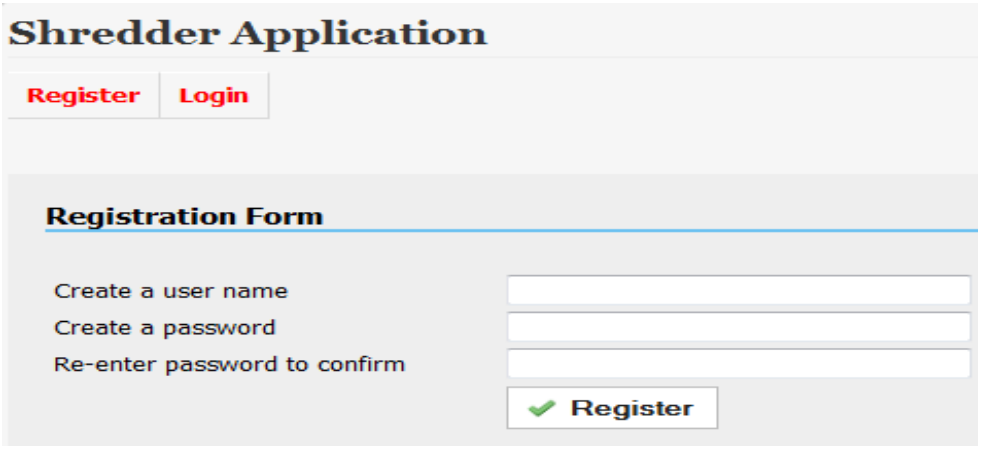


Figure B-3: Registration Form

If the user gets the message "You are now registered! Now login to begin", it means that the user's login information (login id and password) has been stored in the database.

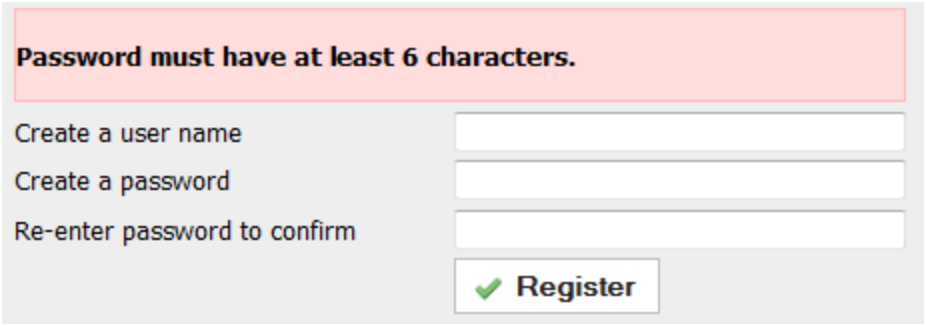


Figure B-4: Registration Form with error Message

A1.2 UPLOADING XML TEXT OR XML FILE(S)

The user may either upload an XML document by either typing or using the 'copy and paste' function in the first window in Figure B-5. The second part of this Figure is where the user can upload the XML Schema Definition (XSD) schema, which is optional. Once this is done, the user may then press the *Submit* button. If there is no error in the XML text, the user will get the message shown in Figure B-6. In the "Enter XML text below" area, the user types the XML document directly and presses *Submit* to shred. The typed text will be validated against the XML format, and if there is any error in the XML file, the user has to correct or fix that error before to upload the text into the database.

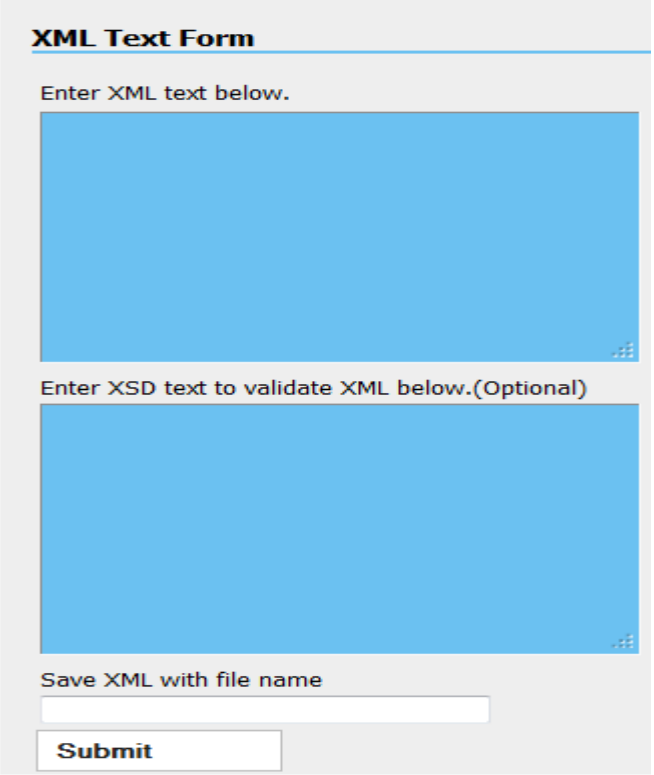


Figure B-5: XML Text Load Form

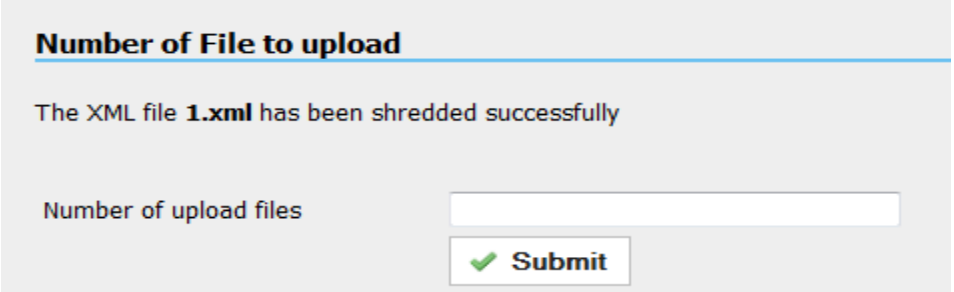


Figure B-6: Message screen for Uploaded XML File

In case the user wants to upload more than one XML file the user has to mention the number of files as shown in Figure B-7 and press the *Submit* button. In the text box, the user will input the total XML files to upload then press Submit button.

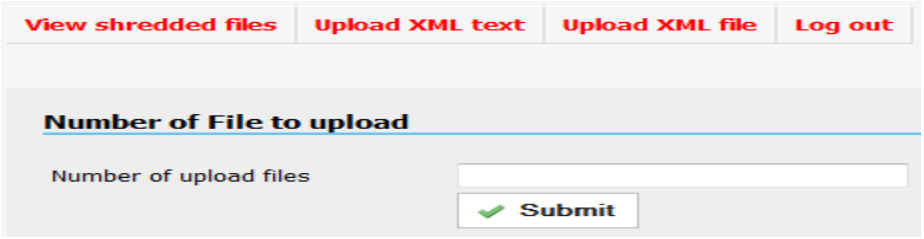


Figure B-7: Number of Files to Upload

The user's desired number of upload forms will be paired with an optional upload form for XSD schema if the user chooses to provide one (see Figure B-8). If the user chooses

to include an optional XSD schema, it must be in the same location as the XML file and the root element of XML file must indicate the file name as attribute as shown in [188] (see below):

eg:<shiporder orderid="889923" xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xsi:noNamespaceSchemaLocation="shiporder.xsd">

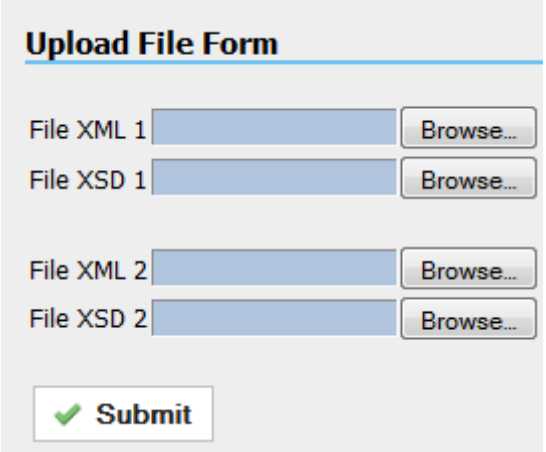


Figure B-8: Form to Upload the XML Files

Once the user uploads the XML file, the user may go into the option menu "View the Shredded File" as shown in Figure B-9.

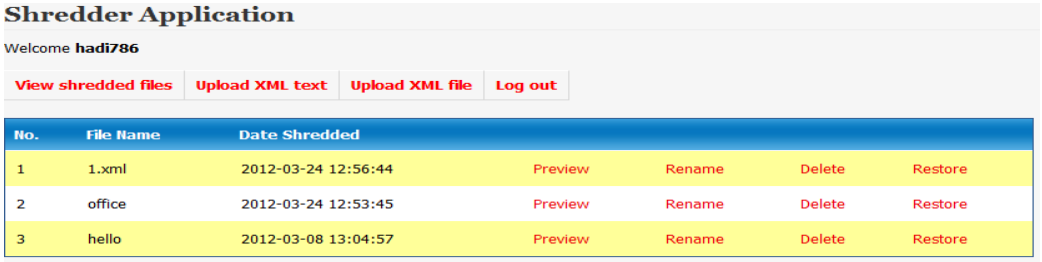


Figure B-9: View of Shredded Files

A1.3 PREVIEW

In Figure B-9, we can see different options set with each uploaded file like Preview, Rename, Delete and Restore. If there is more than one uploaded file, the user may have to select the record/file to apply any listed options. By clicking on the Preview link, the user will be able to see the output as shown in Listing B-1 (implies XML tree has node level indentation).

```
<text b_name="Ranshofen, CanReg" id="305825" n="Ranshofen, CanReg$266" type="charter">
  <front>
    <sourceDesc>
      <sourceDescRegest>
        <bibl>Bayerisches Hauptstaatsarchiv, Regesten Nr. 248</bibl>
      </sourceDescRegest>
    </sourceDesc>
  </front>
  <body>
    <idno>1616 VI 01</idno>
    <issued>
      <dateRange from="16160601" to="16160601">1. Juni 1616</dateRange>
    </issued>
  </body>
</text>
```

Listing: B-1: XML file with Review Option

A1.4 RENAME FILE

Click the Rename hyperlink to rename a file. Usually the user can use this option if he/she forgot to enter a file name during the uploading of the XML text. A default file name of "Noname.xml" is automatically assigned. Figure B-10 is the form where the user can rename the saved file with an alternative name. The existing file name will be displayed in this form to make sure that the user really wants to rename this file. The file with the new name can be seen through View Shredded file option.

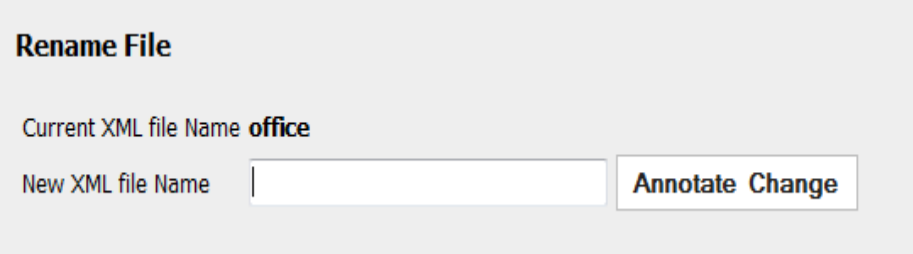


Figure B-10: Rename File Form

A1.5 DELETE FROM THE DATABASE

Click the Delete hyperlink and the selected file will be deleted from the user's account. The user should be very careful as there is no verification set. Once the user has pressed the “DELETE” that file will be gone from the database.

A1.6 RESTORE FROM DATABASE

Using this option causes the XML shredded file to be saved in the defined folder as in this example the file has been saved to the directory path: *./XMLFile/hadi786* as ./XMLFile/hadi786/2012-03-24_12-56-44_1.xml

By clicking the Restore hyperlink a download link will be created. The restored file will automatically be saved to a default location which is as follows:
\XMLFile\[Xdocument.user_name]\[timestampFileShredded].fileName

- XMLFile is the default folder
- [Xdocument.user_name] is the user's user name used as the folder within XMLFile folder
- [timestampFileShredded] is the prefix appended to each filename to ensure the restored files is unique

The user must define "*XMLFile*" folder, and set the proper permission to restore the files from the database.

A1.7 LOGOUT

By pressing the Logout option, the user will receive a message "You are logged out. Good bye!" with the following Figure B-11 in case the user wants to login again.

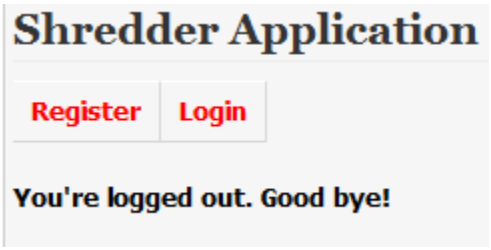


Figure B-11: Register/Login Screen

A2 COMMAND LINE SHREDDER– CMDRUNNER

The command line shredder has the same functionalities as the online shredder. The user can use/run this shredder on any operating system platforms like Unix/Linux and Windows. Here are the steps that the user initially has to take:

- Create a schema named xmlRunner by using MySQL by using the script listed under Data Structure for xmlRunner schema under Appendix B
- Set the parameter values for user name and password in **configuration.php:**

```
<?php
$dbServer="localhost";
$dbDatabaseName = "xmlRunner";
$dbUserName ="root";
$dbPassword = "";
?>
```

- Default working directory is the directory where this cmdRunner.php script resides.
- Open command line prompt and enter directory where the cmdRunner.php is located to execute this script.

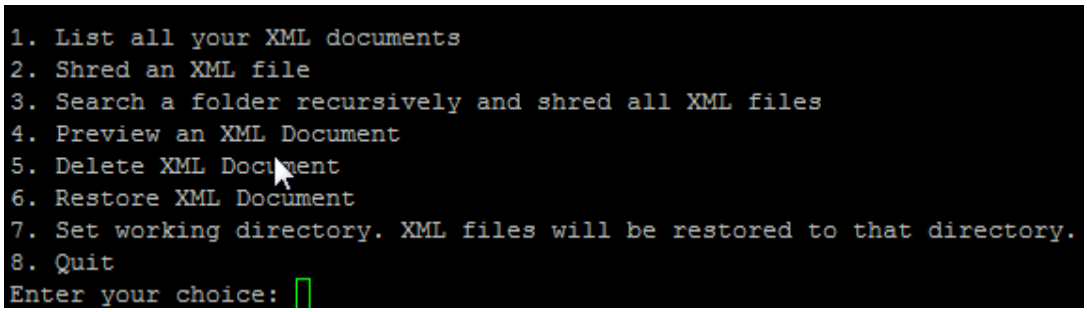
e.g: c:\directory name\>php cmdRunner.php

- In a Unix/Linux environment, change the folder to where you have the cmdRunner.php file, and at $ prompt type:

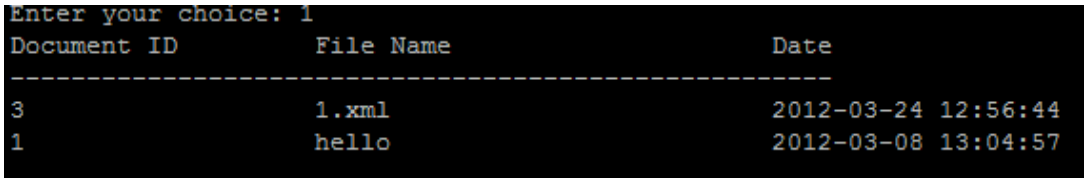
$php cmdRunner.php (type php cmdRunner.php and press Enter)

The user will get the menu with the following options. A new user should choose option 2, "Shred an XML file" to get the registration done so that they may login to shred the XML file. Once the login process completes, the user will have the following options for shredding the documents:

1. List all your XML documents
2. Shred an XML file
3. Search a folder recursively and shred all XML files
4. Preview an XML Document
5. Delete XML Document
6. Restore XML Document
7. Set working directory. XML files will be restored to that directory.
8. Quit



Option 1: In front of "Enter your choice:" enter 1 and press the Enter key. The user will get the list of existing shredded document(s):



Option 2: by selecting option 2 "Shred an XML file" (press 2 and Enter Key), the user will be asked to enter the XML file and XSD file if there is any. Again the XSD file is optional. In this example the user enters the file name "*office.xml*" for shredding. If there is any error in the XML, the user will get an error message. The user will then have to fix that error and try again. Once the files have been shredded the shredder will respond with the message that states the files have been shredded successfully (see below). The user can enter more than one file at a time to shred like: file1.xml/file2.xml, and these two files should be in the same directory where cmdRunner.php is located.

```
Enter your choice: 2

Please enter the XML file name:
office.xml
Please enter the XSD file name (if you don't have xsd, press ENTER):


The XML file has been shredded successfully.
```

For verification here is the screen shot from the database with office.xml:

No.	File Name	Date Shredded				
1	office.xml	2012-03-24 16:10:07	Preview	Rename	Delete	Restore
2	1.xml	2012-03-24 12:56:44	Preview	Rename	Delete	Restore
3	hello	2012-03-08 13:04:57	Preview	Rename	Delete	Restore

Option 3 – Search a folder recursively and shred all XML files: To search a folder recursively, just copy and paste the complete directory. As it is shredding, it will output all the errors for each XML file that is not well-formed. It will also report those files that are successfully shredded. The user does not have to worry if user forgets the '\' as it will automatically insert it before shredding each XML file. If the entered file is not properly formatted, the user will get the error message and return to the main menu (see below). User can use the following format when entering a folder to shred all the XML files under that folder: c:\dir\subdir and press enter key to run.

```
Enter your choice: 3
Please enter the directory for searching:
/var/www/alim/researchworks/shredder
<br/>
<b>Warning 1549</b>: failed to load external entity "/var/www/alim/researchworks/shredder\SIGMOD.xml" on line <b>0</b>
<br/>
<b>Warning 1549</b>: failed to load external entity "/var/www/alim/researchworks/shredder\office.xml" on line <b>0</b>
```

Option 4 – Preview: since the command line can only display so much text, you may need to enter 1 to redisplay all your currently shredded files. Then enter the document ID, and press enter to see it displayed in the command line (*see below).* Pressing 4 and the document id, in this case document id is 1, and press Enter key to process it.

```
Enter your choice: 4

Please enter the XML document ID: 1


<?xml version="1.0" encoding="utf-8"?>
<h2 xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <a href="profile?id=userA">John</a>
  <a href="index.php">my home page!</a>
</h2>
```

Option 5 – Delete option is to delete a selected file from the database. If the user cannot see the current list of the shredded document(s) since the command line view restricts how much the user can scroll back up. The user will need to re-enter 1 to show the shredded files. Then press option 5, then the *doc_id* to delete as in this case the user has deleted a document whose ID is 1.

```
Enter your choice: 5

Please enter the XML document ID: 1
The XML File with document ID 1 has been deleted.
```

Option 6 – Restore option is used to restore a shredded file back into an XML format. The user may enter the *doc_id*, and that file will be restored to where the cmdRunner.php script resides. The file name format will be saved as: timestampshredded+fileName.xml e.g., (see below):

```
1. List all your XML documents
2. Shred an XML file
3. Search a folder recursively and shred all XML files
4. Preview an XML Document
5. Delete XML Document
6. Restore XML Document
7. Set working directory. XML files will be restored to that directory.
8. Quit
Enter your choice: 6

Please enter the XML document ID: 2
The XML File has been restored with name .\2011-12-18_19-09-49_SIGMOD.xml.
```

Option 7 – Set working directory is used to set the directory where the user wants to save the files.

Option 8 – Quite the application can be used by the user when the user has wants.

A3 How to Use the Java Shredder – XMLParser Application

DPEV shredder for JAVA language has been named an XMLParser. This shredder only is run on a command line in any operating system like Unix/Linux as well as in a Windows environment. The main components of this tool are as follow:

- Schema created file
- xmlParser.jar file
- hibernate.cfg.xml to configure the database and password

XMLParser application takes two parameters for shredding XML file(s) or folders that contain XML file(s) and 3 parameters for un-shredding from relational database into XML format:

- First Parameter: it should be either S or U, where S indicates shredding i.e. users want to shred a file or all files (in a folder) as specified by 2nd parameter, and U indicates Un-shredding - converting data back to an xml file.
- Second Parameter is the file with complete path or folder that contains files to be shredded. For un-shredding, the 2nd parameter must be a folder location (file path) where users want to put users un-shredded XML file.
- Third parameter is only required for un-shredding, and it is the file ID of the file to be un-shredded. The file ID must be in the database

A3.1 SHREDDING A SINGLE FILE

If user wants to shred just one file, pass in file name with full path, application will shred only that file.

To run XMLParser.jar program on Windows, start a command prompt. Change to the folder with the JAR file is located and run the program with a "java" command to upload the XML file(s) in the database:

Java -jar XMLParser.jar S c:\folder name\xml file name or folder name where all the XML files are located, even in subfolders

For example:

"C:\Program Files (x86)\Java\jdk1.6.0\bin\Java" -jar XMLParser.jar S c:\DPEV\1.XML

In the example above application will shred only example.xml located on C: drive

A3.2 SHREDDING AND UN-SHREDDING DOCUMENTS

If the user wants to shred all files in a folder pass in the path for that folder. Application will shred all file in that folder and any sub-folders.

Example:

Java -jar XMLParser.jar S c:\Localdata\db

In this example, application will shred all files located in C:\Localdata\db folder and its sub-folders.

Un-Shredding:

Java -jar XMLParser.jar U c:\Un-Shredd 4

Where

U = tells system to unshared

c:\Un-Shredd = tells system the location of output file on file system

4 = tells system to un-shred file having fileID 4

A3.3 To Run Parser through Linux/Unix

USE THE COMMAND:
java -cp .:XMLParser.jar miw.dpev.business.ConvertXMLFile2RDB **S** /dir/folder name or xml file name

Example
java -cp .:XMLParser.jar miw.dpev.business.ConvertXMLFile2RDB S /home/mushtaq/db

java -cp .:XMLParser.jar miw.dpev.business.ConvertXMLFile2RDB S /home/mushtaq/dictionary

Hibernate.cfg.xml

User have to use the Hibernate.cfg.xml file in along with the shredder jar file to run the shredder. In this file, the user has to change the "*password, username and database/schema name*".

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate CONFIGURATION DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory name="database connection setting">
  <property name="hibernate.connection.password">PASSWORD</property>
  <property name="hibernate.connection.url">jdbc:MySQL://localhost/MySQL</property>
  <property name="hibernate.connection.usersname">root</property>
  <!-- property name="hibernate.default_schema">dpev</property -->
  <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
  <!-- property name="hibernate.connection.driver_class">sun.jdbc.odbc.JdbcOdbcDriver</property -->
  <property name="hibernate.connection.driver_class">com.MySQL.jdbc.Driver</property>
  <!-- Use the C3P0 connection pool provider -->
  <property name="hibernate.c3p0.min_size">5</property>
  <property name="hibernate.c3p0.max_size">20</property>
  <property name="hibernate.c3p0.timeout">300</property>
  <property name="hibernate.c3p0.max_statements">50</property>
  <property name="hibernate.c3p0.idle_test_period">3000</property>
  <!-- Show and print nice SQL on stdout -->
  <property name="hibernate.format_sql">true</property>
  <property name="hibernate.show_sql">false</property>
  <property name="hibernate.hbm2ddl.auto">update</property>
  <mapping class="miw.dpev.business.Value" package="miw/dpev/business"
resource="miw/dpev/business/Value.hbm.xml"/>
  <mapping class="miw.dpev.business.Xdocument"
   package="miw/dpev/business" resource="miw/dpev/business/Xdocument.hbm.xml"/>
  <mapping class="miw.dpev.business.Pathexpress"
   package="miw/dpev/business" resource="miw/dpev/business/Pathexpress.hbm.xml"/>
  <mapping class="miw.dpev.business.Edge" package="miw/dpev/business"
resource="miw/dpev/business/Edge.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```