Rapid Application Development using

DbForms Developer's GUI

1 Preface

Understanding this paper requires

- Basic knowledge of XML and XSL
  http://www.xml.org
- Basic knowledge of Java and JSP
  http://www.javasoft.com
- Familiarity with DbForms Framework
  http://www.wap-force.net/dbforms

Applying the methods described in this paper requires

- Servlet 2.2-compliant JSP-Container
  http://jakarta.apache.org/tomcat
- DbForms release 0.9
  http://www.wap-force.net/dbforms
- XML parser and XSL engine Xalan
  http://xml.apache.org/
  http://java.sun.com/xml/

2 Installing DbForms

This is, basically, extracting the latest DbForms-distribution into a directory and setting an environment variable pointing to this directory. See files "readme" and "readmeGui" in DbForms-distribution for detailed instructions.

3 Creating your Web-Application

Create a directory representing you new web application in your web-server. Let us choose the name "dbforms_test" for our example.

If you are using Apache Tomcat, you would create a subdirectory "dbforms_test" in the "webapps" directory of your Tomcat installation.

The initial structure of your blank web application could look this:

dbforms_test\	Web Application "Root directory"
dbforms_test\WEB-INF\web.xml	DbForms Web Appliation Deployment Descriptor Check ($DbForms distribution)\etc\
dbforms_test\WEB-INF\taglib.tld	DbForms Custom tag library Descriptor Check ($DbForms distribution)\etc\
dbforms_test\WEB-INF\lib\dbforms_v09*x*.jar	DbForms Framework.
dbforms_test\WEB-INF\lib\struts.jar	Struts Framework. Needed for parsing XML configuration file.
dbforms_test\WEB-INF\lib\cos.jar	O'Reilly's / Jason Hunter's multipart support classes. Needed for managing file uploads

	(=> BLOB and DISKBLOB fields)
dbforms_test\WEB-INF\lib\log4j.jar	Logging Framework.

Table 1 – Initial structure of the new web application

Note: Of course you can put all those jar packages listed in Table 1 into the Web Server's global /lib directory or you can add them to your CLASSPATH, if you prefer.

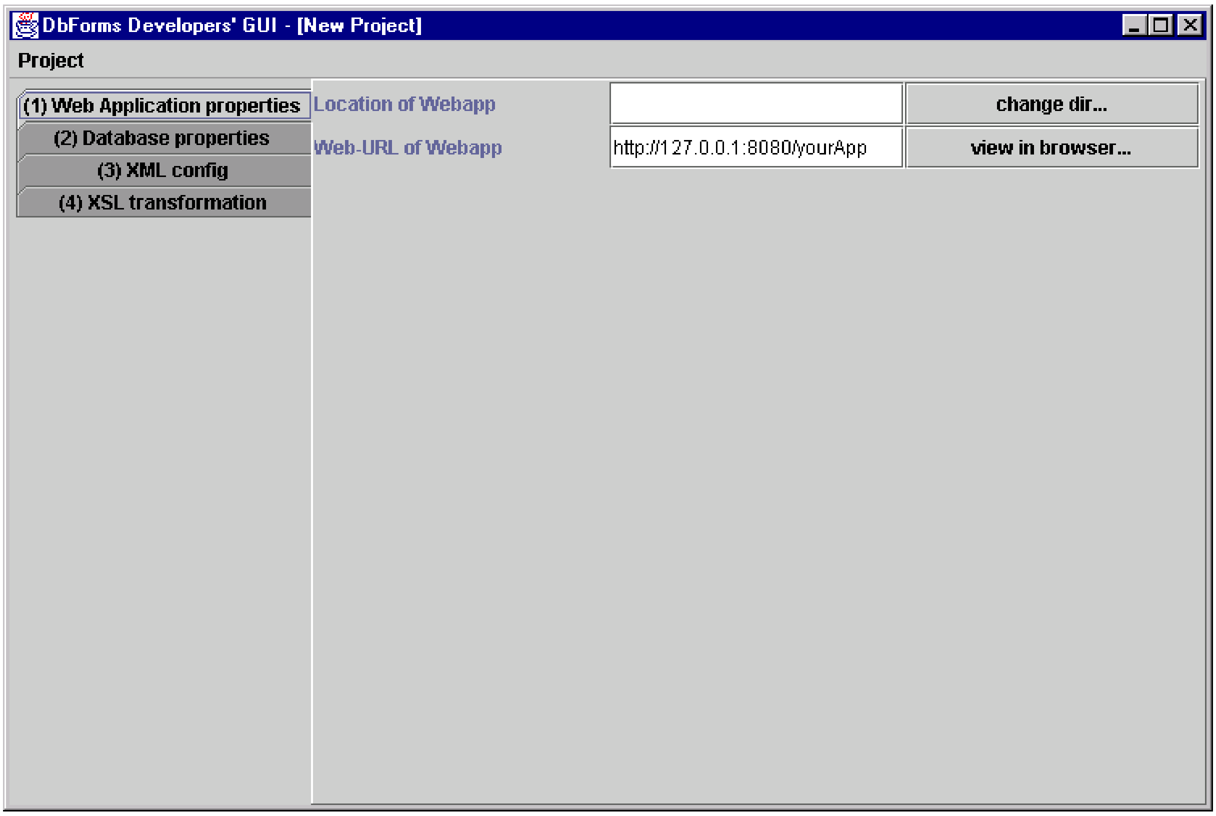
And keep in mind, that you should make your JDBC driver classes available to the web-application as well!

Furthermore you need a Xalan-1 compatible XSL parser available in your classpath. This is "xalan.jar" if using Xalan-1 or "xalan.jar" and "xalanj1compat" if using Xalan-2.

4 Starting the application

Run the shell script "devgui.bat" ("devgui.sh"), which can be found in the /bin directory of your DbForms distribution.

The following window will appear:



DbForms Developers' GUI - [New Project]

Location of Webapp

Web-URL of Webapp: http://127.0.0.1:8080/yourApp

Figure 1 – Web application properties

As you can see, the application is spit up into four panes:

- Web Application properties
- Database properties

- XML config
- XSL transformation

Furthermore you have got a menu "Project" with menu items "New", "Open", "Save", "Save As", "Exit" for loading and storing data you enter and for leaving the application.

It is recommended that you walk through the four panes in the correct order: 1-2-3-4!

This is what we will do in the next paragraphs:

5 Setting Web Application properties

At first, we tell the system more about the Web application we are about to create.

- "Location of Webapp": We provide the absolute local path of the web application.

  Example: "x:\jaktarta-tomcat\webapps\dbforms_test"

- "Web-URL of Webapp": We provide the HTTP Address a browser would need in order to access our web application

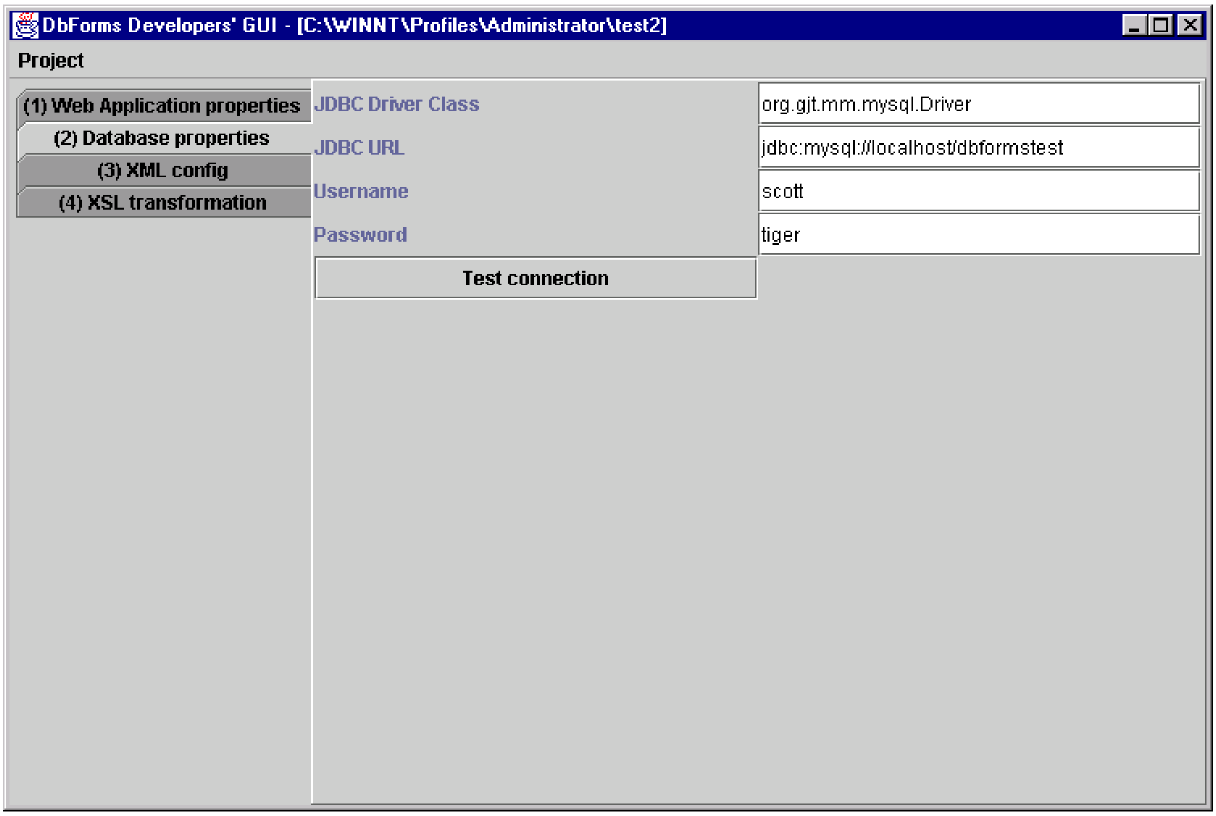
  Example: http://127.0.0.1/dbforms_test

  This information is just needed to enable quick testing of the JSPs. (No code will be generated based on this information!)

Okay, now the DevGui knows about the location of the WebApplication what it needs to know.

6 Setting database properties

Now we provide data needed to connect to the database we want to build a web-interface for:



DbForms Developers' GUI - [C:\WINNT\Profiles\Administrator\test2]
Project
(1) Web Application properties
(2) Database properties
(3) XML config
(4) XSL transformation
Test connection

JDBC Driver Class: org.gjt.mm.mysql.Driver
JDBC URL: jdbc:mysql://localhost/dbformstest
Username: scott
Password: tiger

Figure 2 – database properties

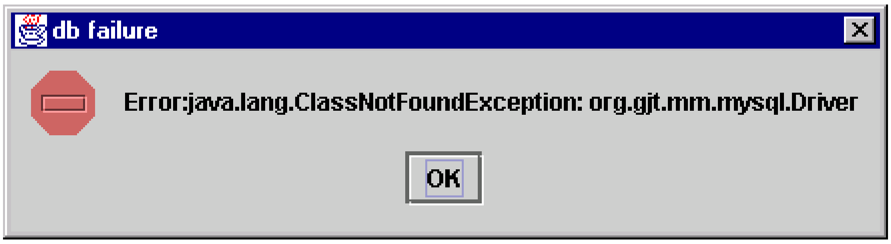
- "JDBC Driver Class" – must be in current CLASSPATH!
  example: "org.gjt.mm.mysql.Driver"

- "JDBC URL"
  example: "jdbc:mysql://localhost/dbformstest"

- Username
  example: "scott"

- Password
  example: "tiger"

You can test your connection by clicking the button "Test connection". If you get a message like that one in the dialog below, then you may have to check if your jdbc driver is in CLASSPATH, if your database is actually running, etc.



```
Error:java.lang.ClassNotFoundException: org.gjt.mm.mysql.Driver
```

Figure 3 – one of many errors that can appear when connecting to a database

If you get the message "Database connection successfully installed", then everything went right and you can go to the next tab.

By the way: we don't need to use any connection pooling here, because we just use one connection and we do not need full speed when developing (ahrg! forget about the last one ;=)

7 Creating XML config file for DbForms

The XML configuration file is the **most important piece of data for DbForms**. It tells DbForms important fact about the database: its tables, fields, field types, keys, etc.

All our further work will be based on these configuration file!

To create a raw version of this file, perform the following steps:

- Click on the "browse" button, navigate down to the "WEB-INF" directory of your Web-Application and enter the default name dbforms-config.xml.

  example: "x:\jakarta-tomcat\webapps\dbforms_test\WEB-INF\dbforms-config.xml"

  The name "dbforms-config.xml" is **default**. Any other name would have to be declared explicitly in your Web.xml file as init-parameter "dbformsConfig" for the servlet named "config".

- After the path of the config file is declared, you can press the button "Generate XML!". The application now connects to the database described in the last paragraph. It tries to read some metadata about tables and fields and uses this information to create the dbforms-config.xml file for you.

- Review the result and apply changes, if necessary. Manual changes can be required, for two different reasons:

  - **1. Technical limitations…**
    - …in JDBC driver

      If your JDBC driver does not correctly implement the methods for metadata extraction, you have to provide that data by hand.

      For example if your JDBC driver does not correctly deliver a list of key-fields for a table, then you 'll have to add the *isKey="true"* attributes by hand. Another widespread problem is the extraction of "auto-incremental" keys.

    - …in DbForms DevGui Tool

      some limitations may stem from the bad (alpha) quality of code responsible for metadata extraction used by the DevGui tool. Therefore, bug reports and bug fixes are very useful and welcome!

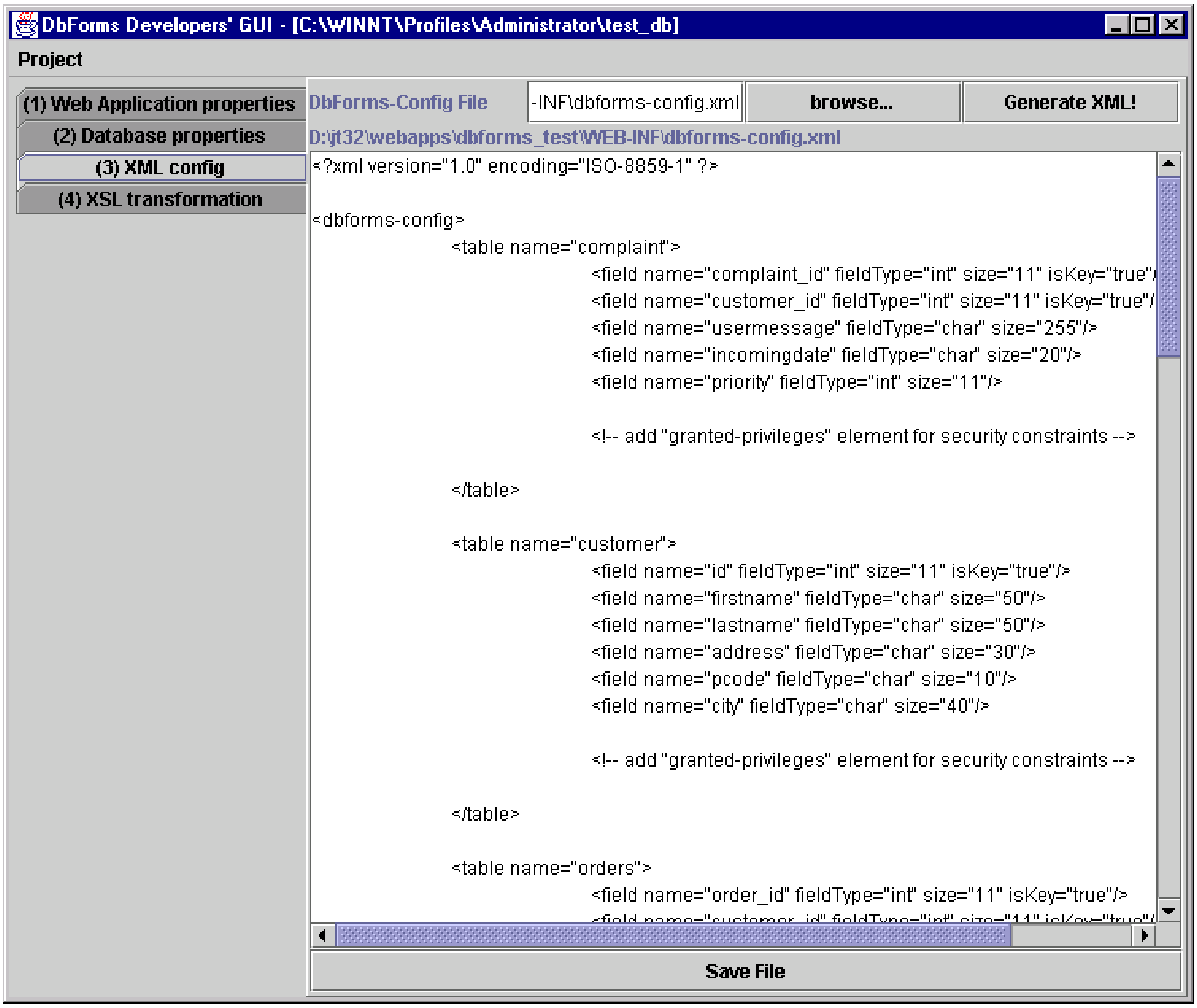
- **2. Conceptual limitations**

  the scope addressed by the DbForms XML config file goes far beyond the information stored in metadata of a database:

  By utilizing the DbForms XML config file you can
  - install Database Event interceptors
  - install fine grained security constraints
  - use simple CHAR fields for so called "DISKBLOB" fields
  - declare fields as "sortable"
  - and last but not least, you can even configure a connection pool instead of the default connection

  It is impossible for the database to "know" about all those things ;=) therefore you need to handle those configurations by hand.

- Press button "Save File".
  This stores your config file info the location specified by the "DbForms-Config File" textfield.


DbForms Developers' GUI - [C:\WINNT\Profiles\Administrator\test_db]
Project
(1) Web Application properties
(2) Database properties
(3) XML config
(4) XSL transformation
browse...
Generate XML!
Save File

DbForms-Config File: -INF\dbforms-config.xml

D:\jt32\webapps\dbforms_test\WEB-INF\dbforms-config.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<dbforms-config>
		<table name="complaint">
				<field name="complaint_id" fieldType="int" size="11" isKey="true"/
				<field name="customer_id" fieldType="int" size="11" isKey="true"/
				<field name="usermessage" fieldType="char" size="255"/>
				<field name="incomingdate" fieldType="char" size="20"/>
				<field name="priority" fieldType="int" size="11"/>

				<!-- add "granted-privileges" element for security constraints -->

		</table>

		<table name="customer">
				<field name="id" fieldType="int" size="11" isKey="true"/>
				<field name="firstname" fieldType="char" size="50"/>
				<field name="lastname" fieldType="char" size="50"/>
				<field name="address" fieldType="char" size="30"/>
				<field name="pcode" fieldType="char" size="10"/>
				<field name="city" fieldType="char" size="40"/>

				<!-- add "granted-privileges" element for security constraints -->

		</table>

		<table name="orders">
				<field name="order_id" fieldType="int" size="11" isKey="true"/>
				<field name="customer_id" fieldType="int" size="11" isKey="true"/
```

Figure 4 – XML Config

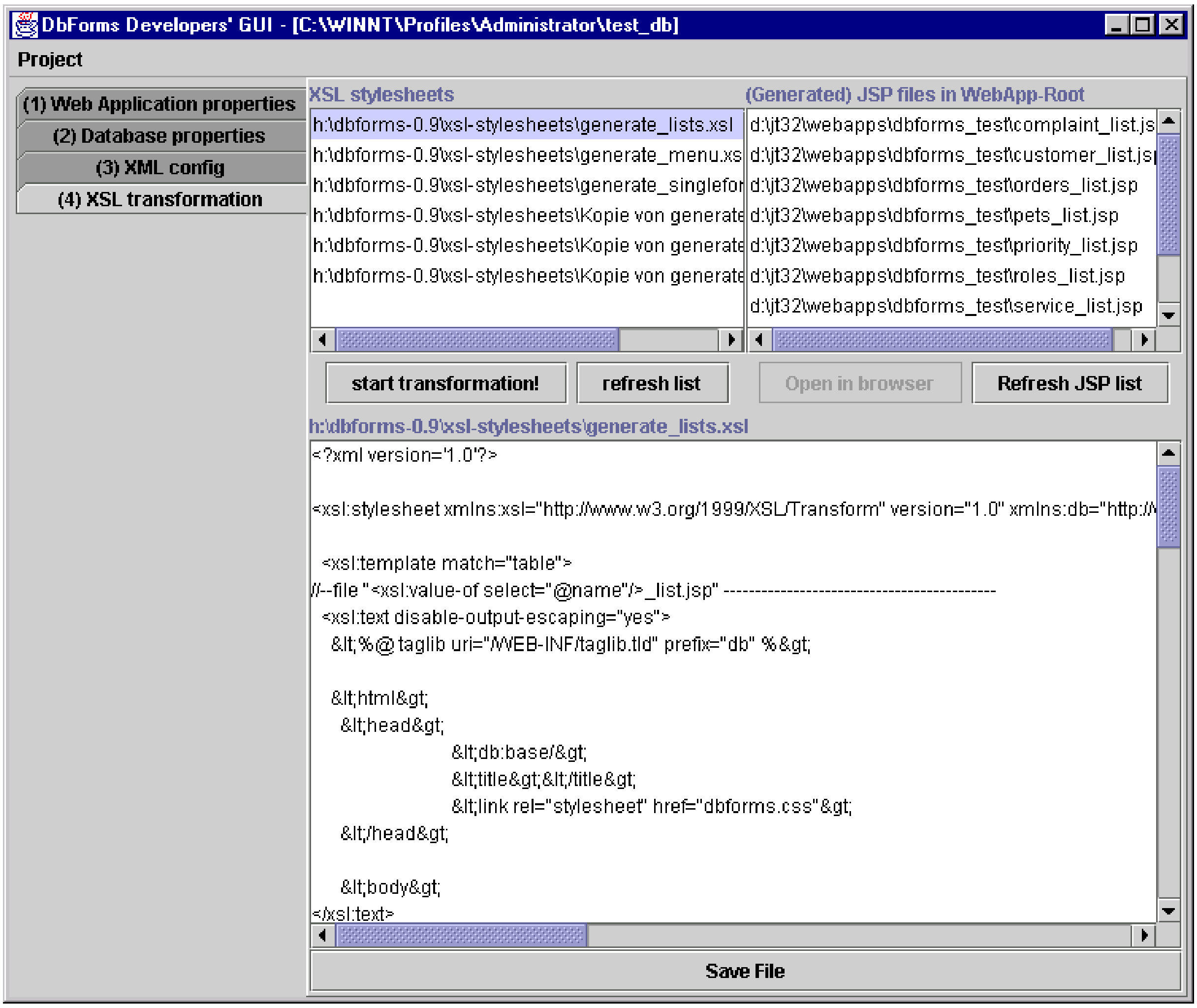
8 XSL Transformation

We had to do pretty much configuration work, but now we have reached or goal: we are in the nice position to use the power and ease of XSL for autmatically (!) generate JSP views according to the XML config file we generated!

If you have not saved your project, then save it now. You can open it every time when you want to generate new JSPs! Simply choose Menu item "Project/Save" and choose any name you like.

In the list box "XSL stylesheets" you have got a list of the XSL files in the "xsl-stylesheets" folder of your DbForms-distribution. You can add your own stylesheets there. If you hit the "refresh list" button, the directory list will be regenerated to bring up the latest XSL files. The "xsl-stylesheets" directory is thought to be a kind of *growing repository*, and it would be useful if people start sharing powerful DbForms - XSL stylesheets.

If you select a XSL stylesheet, it will be shown in the editor box below, where it can be edited. Note: later versions of this GUI will provide XML syntax highlighting and other features.



```
h:\dbforms-0.9\xsl-stylesheets\generate_lists.xsl
h:\dbforms-0.9\xsl-stylesheets\generate_menu.xs
h:\dbforms-0.9\xsl-stylesheets\generate_singlefor
h:\dbforms-0.9\xsl-stylesheets\Kopie von generate
h:\dbforms-0.9\xsl-stylesheets\Kopie von generate
h:\dbforms-0.9\xsl-stylesheets\Kopie von generate
```

```
d:\jt32\webapps\dbforms_test\complaint_list.js
d:\jt32\webapps\dbforms_test\customer_list.js
d:\jt32\webapps\dbforms_test\orders_list.jsp
d:\jt32\webapps\dbforms_test\pets_list.jsp
d:\jt32\webapps\dbforms_test\priority_list.jsp
d:\jt32\webapps\dbforms_test\roles_list.jsp
d:\jt32\webapps\dbforms_test\service_list.jsp
```

h:\dbforms-0.9\xsl-stylesheets\generate_lists.xsl

```
<?xml version='1.0'?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:db="http://

  <xsl:template match="table">
//--file "<xsl:value-of select="@name"/>_list.jsp" -----------------------------------------
  <xsl:text disable-output-escaping="yes">
   &lt;%@ taglib uri="/WEB-INF/taglib.tld" prefix="db" %&gt;

   &lt;html&gt;
     &lt;head&gt;
                  &lt;db:base/&gt;
                  &lt;title&gt;&lt;/title&gt;
                  &lt;link rel="stylesheet" href="dbforms.css"&gt;
     &lt;/head&gt;

     &lt;body&gt;
</xsl:text>
```

Figure 5 – Generating JSP Views

To generate JSPs, select the XSL stylesheet you want to use and then click the button "start transformation!".

If everything goes well, then you should see the resulting JSP files in the list "(Generated) JSP files in WebApp-Root". If something goes fundamentally wrong (for instance if the XSL file is malformed), then you should get an error-dialog window. XSL-*warnings*, on the other hand, are printed on the system's standard output console.

Always check the results and check for XSL warning-messages on the Standard-Output! If you get errors or warnings when using XSL stylesheets from the official DbForms Project Homepage, then please report those warnings and errors.

If your Web-server is running, you can immediately test the results. You just have to select a JSP file from the list "(Generated) JSP files in WebApp-Root" and then click the button "Open in Browser".

This should invoke your default browser which should render the fresh JSP view !

9 Related Documents

- **DbForms User's Guide**, Joachim Peer, 2001
- **XSL-Preprocessing Layer for DbForms**, Peer, 2001
- **How to create a Bug tracking system in 10 minutes**, Joachim Peer, 2001 [– not available yet]