



SIMPLY RICH

ZK

The Quick Start Guide

March 2006

Potix Corporation

Copyright © Potix Corporation. All rights reserved.

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made to assure its accuracy, Potix Corporation assumes no liability resulting from errors or omissions in this document, or from the use of the information contained herein.

Potix Corporation may have patents, patent applications, copyright or other intellectual property rights covering the subject matter of this document. The furnishing of this document does not give you any license to these patents, copyrights or other intellectual property.

Potix Corporation reserves the right to make changes in the product design without reservation and without notification to its users.

The Potix logo and ZK are trademarks of Potix Corporation.

All other product names are trademarks, registered trademarks, or trade names of their respective owners.

Table of Contents

1. Installation.....	4
Install ZK on Tomcat.....	4
Install ZK on Jetty.....	4
Deploy your Application as a WAR file or an EAR file.....	5
Working with MySQL.....	5
2. The Content of Distribution.....	6
demo.....	6
demo/bin.....	6
demo/src.....	6
demo/src/userguide.....	7
demo/src/test.....	7
demo/src/WEB-INF/web.xml.....	7
doc.....	7
dist/lib.....	7
dist/lib/zkforge.....	7
dist/lib/ext.....	7
dist/WEB-INF.....	7
3. My First ZK Application.....	8
My First Hello World.....	10

1. Installation

Install ZK on Tomcat

1. Download Tomcat from <http://tomcat.apache.org> and install it, if you haven't installed it yet.
2. Stop Tomcat.
3. Unzip zk-1.0.0.zip or zk-1.0.0.tar.gz
4. Copy `dist/lib/*.jar` to `$TOMCAT_HOME1/shared/lib`
5. Copy `dist/lib/ext/*.jar` to `$TOMCAT_HOME/shared/lib`
6. [Optional] Copy `dist/lib/zkforge/*.jar` to `$TOMCAT_HOME2/shared/lib`
It depends whether you need components from ZK Forge³, such as FCKeditor (<http://www.fckeditor.net>) and DOJO (<http://dojotoolkit.org/>).
7. Re-start Tomcat.
8. Deploy `demo/bin/zkdemo.war` to Tomcat. It can be done by use of the Tomcat manager, or by copying it to `$TOMCAT_HOME/webapps` directly. If you prefer copying directly, you have to stop Tomcat first.
9. Browse to `http://localhost/zkdemo/userguide` or `http://localhost:8080/zkdemo/userguide`, depending on your configuration.

Install ZK on Jetty

1. Download Jetty from <http://www.mortbay.org/jetty/index.html> and install it⁴, if you haven't installed it yet.
2. Stop Jetty.
3. Unzip zk-1.0.0.zip or zk-1.0.0.tar.gz
4. Copy `dist/lib/*.jar` to `$JETTY_HOME5/ext`

¹ \$TOMCAT_HOME is where you installed Tomcat.

² \$TOMCAT_HOME is where you installed Tomcat.

³ ZK Forge (<http://zkforge.sourceforge.net>) is a collection of components from the community collaboration.

⁴ Refer to <http://www.mortbay.org/jetty/tut/GettingStarted.html>

5. Copy `dist/lib/ext/*.jar` to `$JETTY_HOME/ext`
6. [Optional] Copy `dist/lib/zkforge/*.jar` to `$JETTY_HOME/ext`
It depends whether you need component from ZK Forge.
7. Deploy `demo/bin/zkdemo.war` to Jetty by copying it to `$JETTY_HOME/webapps` directly.
8. Start Jetty.
9. Browse to `http://localhost/zkdemo/userguide` or
`http://localhost:8080/zkdemo/userguide`, depending on your configuration.

Deploy your Application as a WAR file or an EAR file

There are two ways to make your Web application ready to use ZK. First, copy `dist/lib/*.jar`, `dist/lib/zkforge/*.jar`, and `dist/lib/ext/*.jar` to the class-path of the Web or JavaEE container as mentioned in the above sections.

Second, you could bundle them with the WAR or EAR file as follows.

- If your application is a WAR file, copy `dist/lib/*.jar`, `dist/zkforge/*.jar`, and `dist/lib/ext/*.jar` from `zk-1.0.0.zip` to the `/WEB-INF/lib` directory of your WAR file.
- If your application is an EAR file, copy `dist/lib/*.jar`, `dist/zkforge/*.jar`, and `dist/lib/ext/*.jar` from `zk-1.0.0.zip` to the root directory of your EAR file.

Working with MySQL⁶

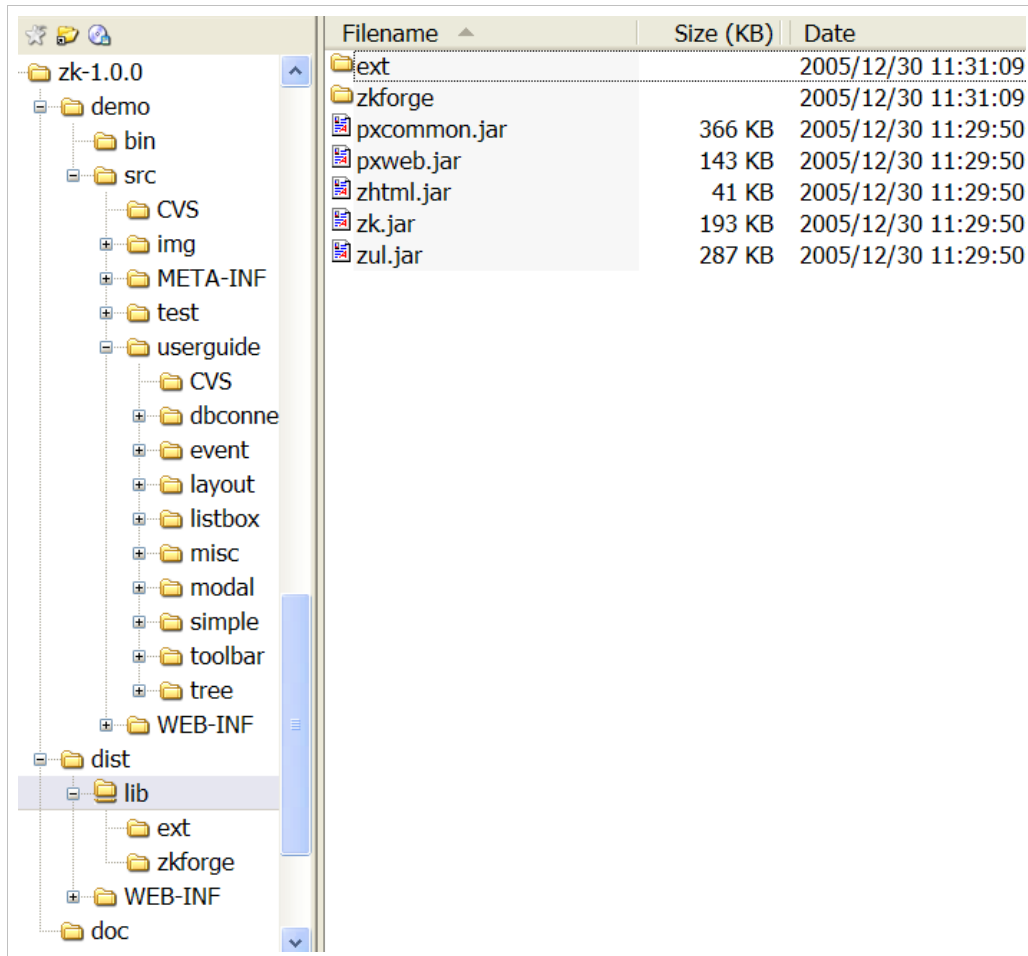
To open the connection under `zscript`, you have to put MySQL JDBC driver (`mysql-connector-*.jar`) under the same directory with ZK libraries (`zk.jar` and others). For example, if you prefer to put ZK libraries under `$TOMCAT_DIR/shared/lib`, you have to put MySQL JDBC driver under `$TOMCAT_DIR/shared/lib`, too.

⁵ `$JETTY_HOME` is where you installed Jetty.

⁶ <http://www.mysql.com>

2. The Content of Distribution

This chapter describes the content of zk-1.0.0.zip.



demo

This directory holds the demo codes, including executable and source codes.

demo/bin

This directory holds the executable, zkdemo.war.

demo/src

This directory holds the source codes of demo and samples.

demo/src/userguide

This directory holds the source codes of the live demo.

demo/src/test

This directory holds the source codes for testing ZK. It explores more ZK features.

demo/src/WEB-INF/web.xml

The file is a sample how to configure ZK for an application.

doc

This directory holds the documents including Quick Start Guide and User Guide.

dist/lib

This directory holds the libraries required to run ZK.

dist/lib/zkforge

This directory holds the components from ZK Forge, such as FCKeditor and Dojo. It is optional depending on whether you need them.

dist/lib/ext

This directory holds the external libraries required to run ZK. Since these libraries are common, you might have installed them in your container.

dist/WEB-INF

This directory holds the TLD files that are useful if you use JSP with ZK.

3. My First ZK Application

Prepare WEB-INF/web.xml

Copy or merge the following content to the web.xml in the WEB-INF directory in your application. This step must be done once each time you created a new Web applications. Then, what you need to do is to copy files with .zul or .zhtml extension to the proper directories in your Web applications.

```
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

    <!-- /// -->
    <!-- DSP -->
    <servlet>
        <description><![CDATA[
The servlet loads the DSP pages.
        ]]></description>
        <servlet-name>dspLoader</servlet-name>
        <servlet-
class>com.potix.web.servlet.dsp.InterpreterServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>dspLoader</servlet-name>
        <url-pattern>*.dsp</url-pattern>
    </servlet-mapping>
    <!-- /// -->

    <!-- //// -->
    <!-- ZK -->
    <listener>
        <description>Used to cleanup when a session is
destroyed</description>
        <display-name>ZK Session Cleaner</display-name>
        <listener-
class>com.potix.zk.ui.http.HttpSessionListener</listener-class>
    </listener>
    <servlet>
        <description>ZK loader for evaluating ZK pages</description>
        <servlet-name>zkLoader</servlet-name>
        <servlet-class>com.potix.zk.ui.http.DHtmlLayoutServlet</servlet-
class>
```



```

        <!-- Must. Specifies URI of the update engine
        (DHtmlUpdateServlet). -->
        <init-param>
            <param-name>update-uri</param-name>
            <param-value>/zkau</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup><!-- MUST -->
    </servlet>
    <servlet-mapping>
        <servlet-name>zkLoader</servlet-name>
        <url-pattern>*.zul</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>zkLoader</servlet-name>
        <url-pattern>*.zhtml</url-pattern>
    </servlet-mapping>
    <servlet>
        <description>The asynchronous update engine for ZK</description>
        <servlet-name>auEngine</servlet-name>
        <servlet-class>com.potix.zk.au.http.DHtmlUpdateServlet</servlet-
class>
    </servlet>
    <servlet-mapping>
        <servlet-name>auEngine</servlet-name>
        <url-pattern>/zkau/*</url-pattern>
    </servlet-mapping>
    <!-- //// -->

    <!-- MIME mapping -->
    <mime-mapping>
        <extension>dsp</extension>
        <mime-type>text/html</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>gif</extension>
        <mime-type>image/gif</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>html</extension>
        <mime-type>text/html</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>htm</extension>
        <mime-type>text/html</mime-type>
    </mime-mapping>
    <mime-mapping>
        <extension>jpeg</extension>
        <mime-type>image/jpeg</mime-type>
    </mime-mapping>

```

```

<mime-mapping>
  <extension>jpg</extension>
  <mime-type>image/jpeg</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>js</extension>
  <mime-type>application/x-javascript</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>png</extension>
  <mime-type>image/png</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>txt</extension>
  <mime-type>text/plain</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>xml</extension>
  <mime-type>text/xml</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>zhtml</extension>
  <mime-type>text/html</mime-type>
</mime-mapping>
<mime-mapping>
  <extension>zul</extension>
  <mime-type>text/html</mime-type>
</mime-mapping>

<welcome-file-list>
  <welcome-file>index.zul</welcome-file>
  <welcome-file>index.zhtml</welcome-file>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.dsp</welcome-file>
</welcome-file-list>
</web-app>

```

My First Hello World

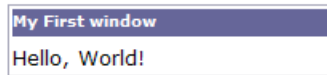
Create a file called `hello.zul` with the following content. Then, you could use the browser to see the result, say `http://localhost:8080/zkdemo/hello.zul`.

```

<window title="My First window" border="normal" width="200px">
  Hello, World!
</window>

```

Then, the result is depicted as follow.



Notice that, though the content of `hello.zul` is very similar to XUL⁷, it is actually written in ZUML. ZK Loader parse it into a valid HTML page which can be interpreted correctly by a regular browser, such as Internet Explorer and Mozilla Firefox. Refer to the Developer's Guide for more details.

⁷ <http://xul.sourceforge.net/mozilla.html>