Using DevCenter 2003
For Developing Web Applications

I. The Development Framework

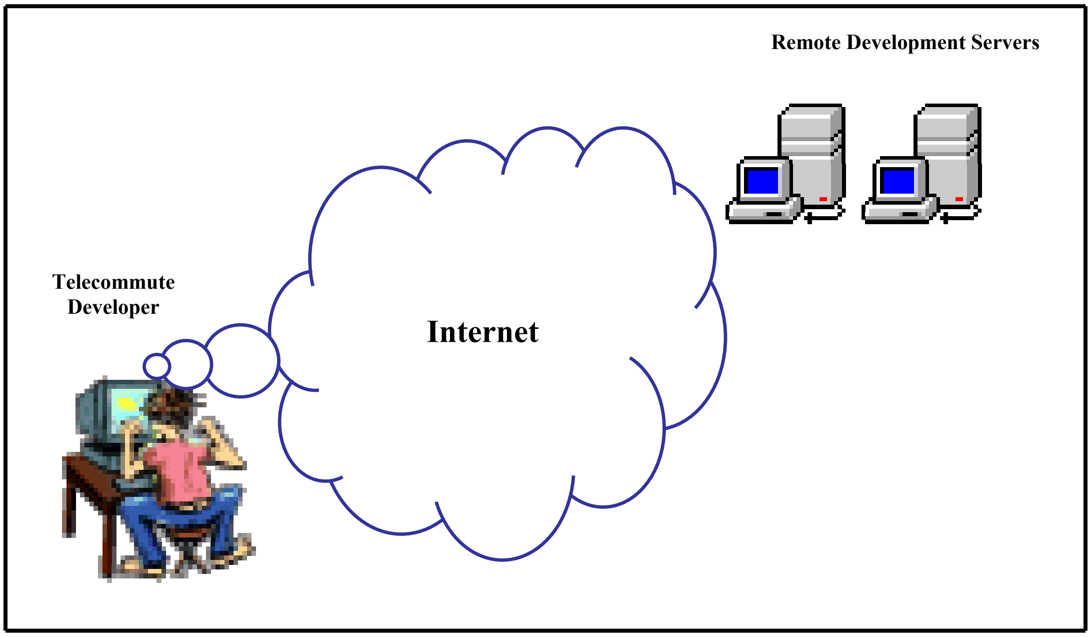
The Likha Software development framework is made up of two general components. The Likha DevCentre IDE and the Rapid Application Loader Engine (RALE).

The DevCentre is a simple, vertical Integrated Development Enviroment for developing web applications using Java's JSP and Servlets technologies. The output file out of DevCentre is a web archive (<filename>.war) that contains all required files needed for the web application to be deployed in any JSP/Servlet engine, e.g. Jakarta Tomcat.

The RALE is a server daemon that interfaces with the JSP/Server engine management tools. It is a layer that simplifies development and test deployment cycles. It receives remote instructions from DevCentre through a Secure Socket Layer connection and executes these instructions remotely against the JSP/Servlet engine and its server environment. This allows the developer to deploy, undeploy, stop, start, reload, unwar web applications without leaving the DevCentre IDE. This also allows the developer to run pre-written utility scripts that manages the server environment as a whole.

This development framework was meant to allow developers to work remotely and securely using any connection to the Internet. File transfer from DevCentre to the remote server uses Secure Shell (SSH) protocol. DevCentre can send single files or the web archive itself. All commands interfacing the DevCentre and RALE uses Secure Sockets Layer. A developer can work on his web application on even dialup connections.

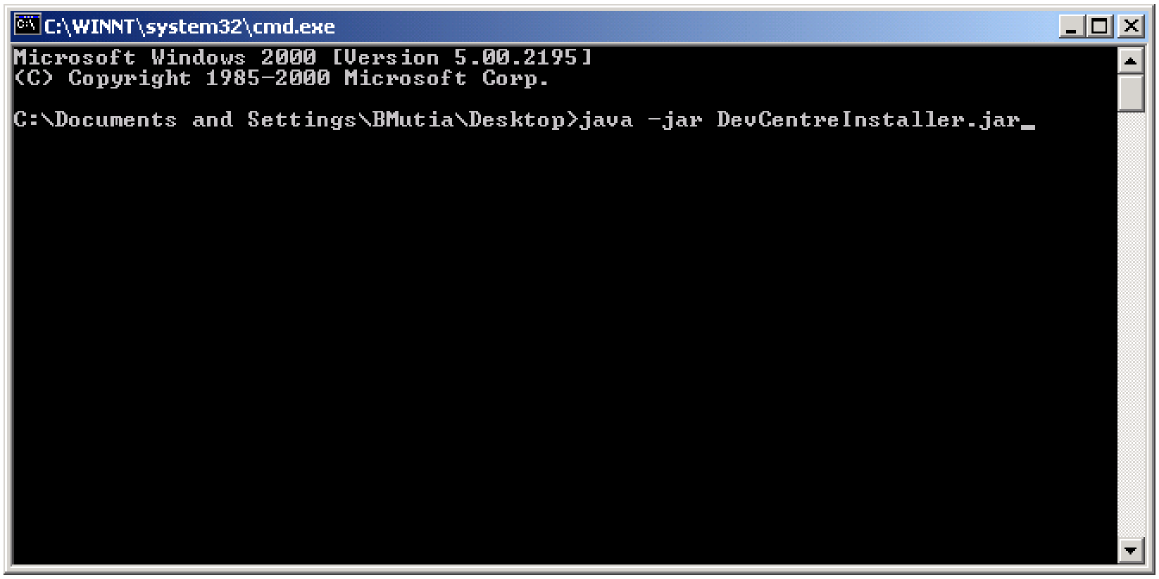
Figure 1



II. Installing DevCenter

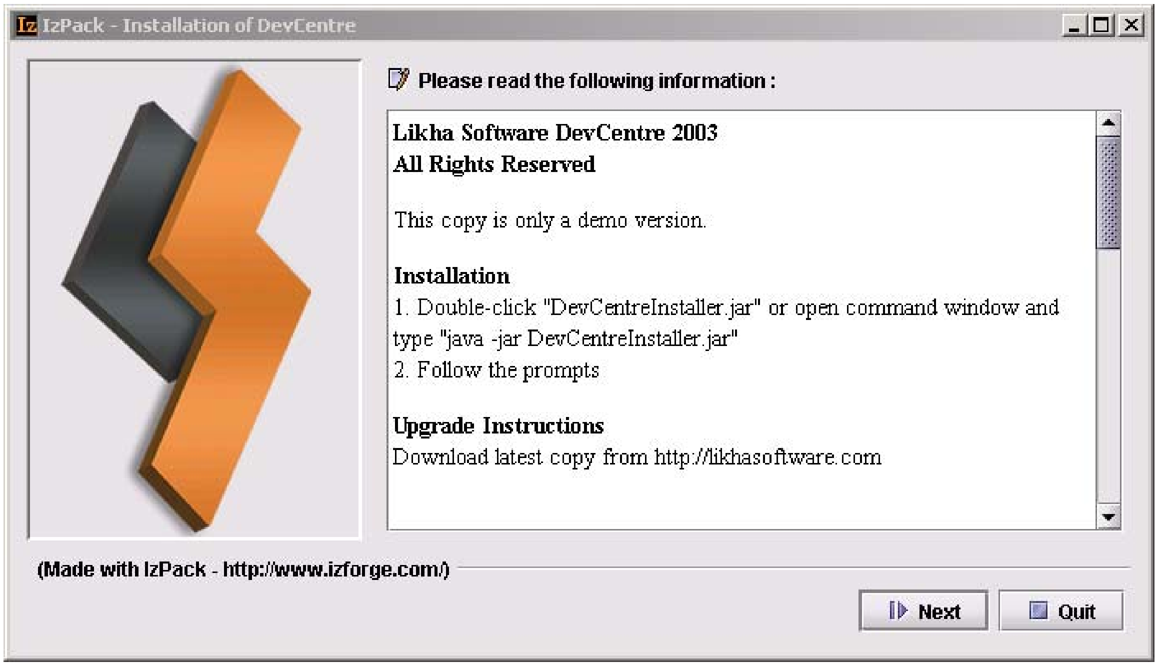
You should have “Java2 SDK 1.4.x” or higher installed in your system before installing DevCentre.

Download “DevCentreInstaller.jar” from http://www.likhasoftware.com website. To install, **double-click the installer jar** file or open command window and run **“java –jar DevCentreInstaller.jar”** like below.



```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\BMutia\Desktop>java -jar DevCentreInstaller.jar_
```

The installation opens up ReadMe text.


IzPack - Installation of DevCentre
Please read the following information :
(Made with IzPack - http://www.izforge.com/)
Next
Quit

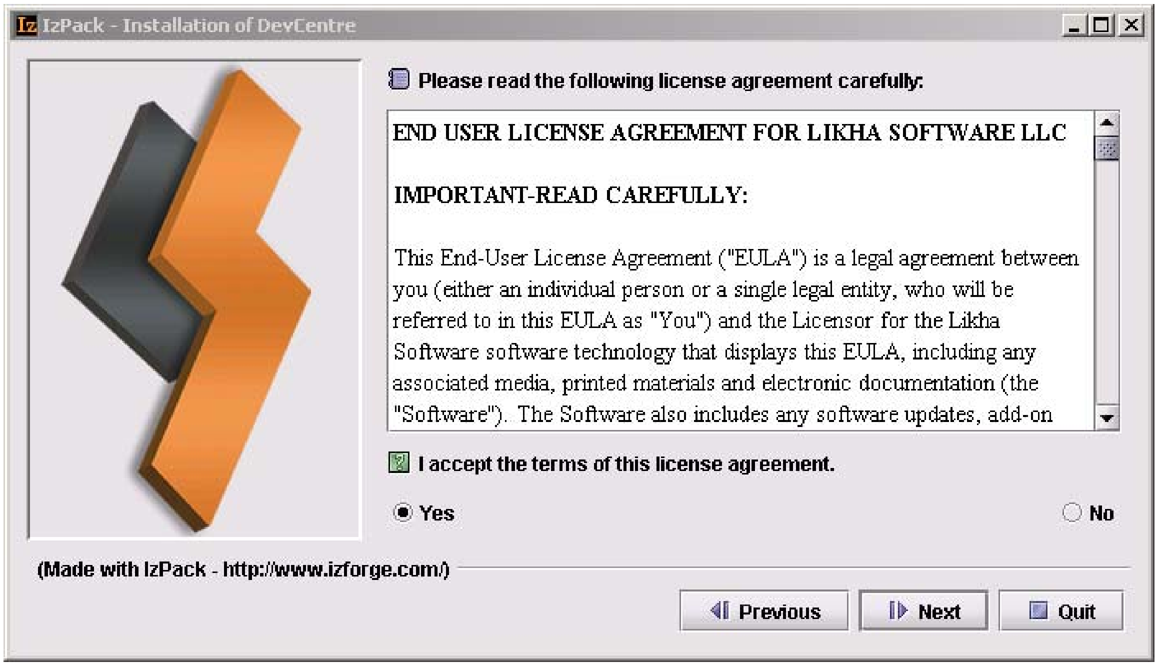
Likha Software DevCentre 2003
All Rights Reserved

This copy is only a demo version.

Installation
1. Double-click "DevCentreInstaller.jar" or open command window and type "java -jar DevCentreInstaller.jar"
2. Follow the prompts

Upgrade Instructions
Download latest copy from http://likhasoftware.com

The next screen opens the End User License Agreement. Please read carefully.


IzPack - Installation of DevCentre
Please read the following license agreement carefully:

END USER LICENSE AGREEMENT FOR LIKHA SOFTWARE LLC

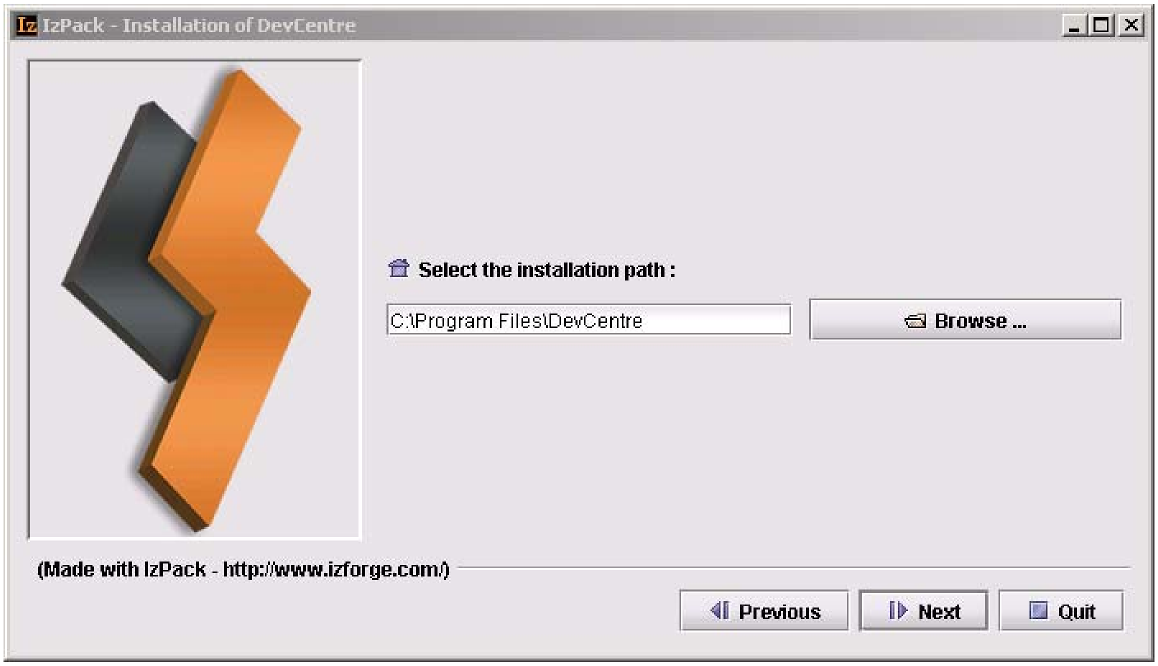
IMPORTANT-READ CAREFULLY:

This End-User License Agreement ("EULA") is a legal agreement between you (either an individual person or a single legal entity, who will be referred to in this EULA as "You") and the Licensor for the Likha Software software technology that displays this EULA, including any associated media, printed materials and electronic documentation (the "Software"). The Software also includes any software updates, add-on

I accept the terms of this license agreement.

● Yes ○ No

Select your installation path.

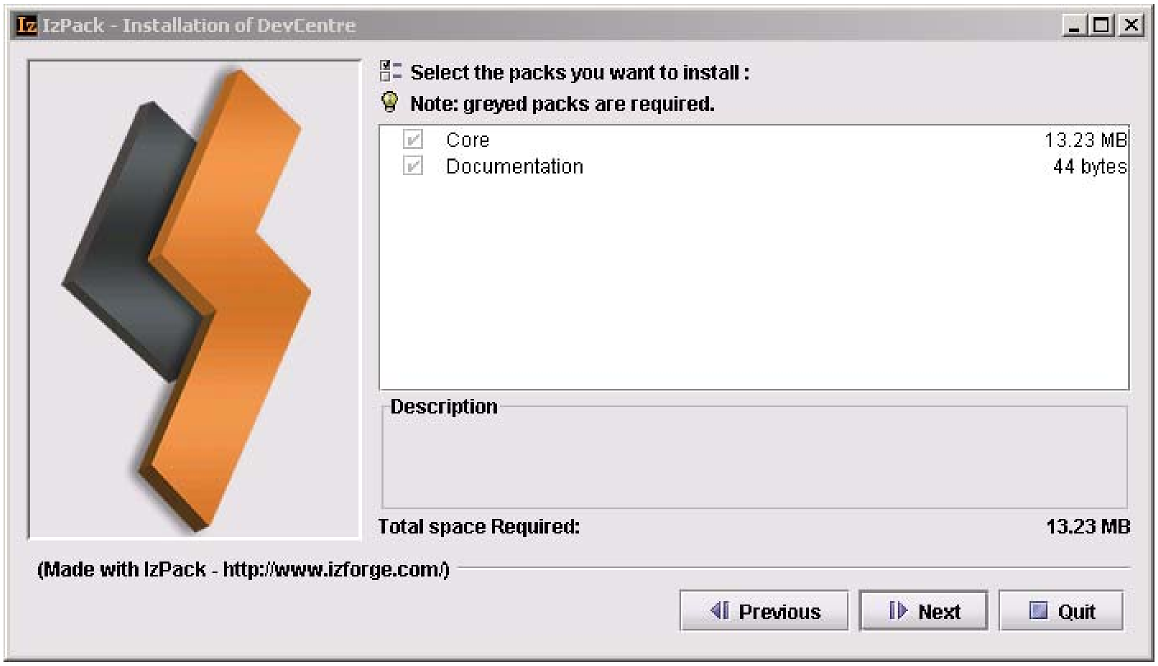

IzPack - Installation of DevCentre

Select the installation path :

C:\Program Files\DevCentre

Browse ...
(Made with IzPack - http://www.izforge.com/)
Previous
Next
Quit

Select the files you want installed. If this is a demo version, all files are required.



IzPack - Installation of DevCentre

Select the packs you want to install :

Note: greyed packs are required.

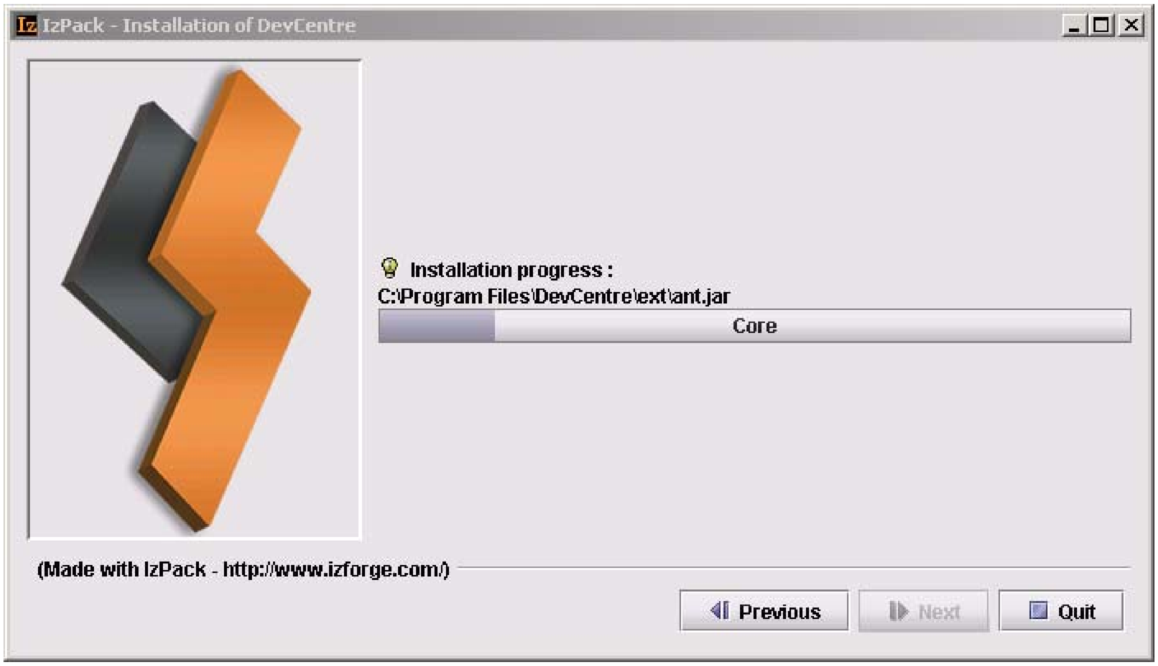
☑	Core	13.23 MB
☑	Documentation	44 bytes

Description

Total space Required: 13.23 MB

(Made with IzPack - http://www.izforge.com/)

The installer will copy the files.



IzPack - Installation of DevCentre

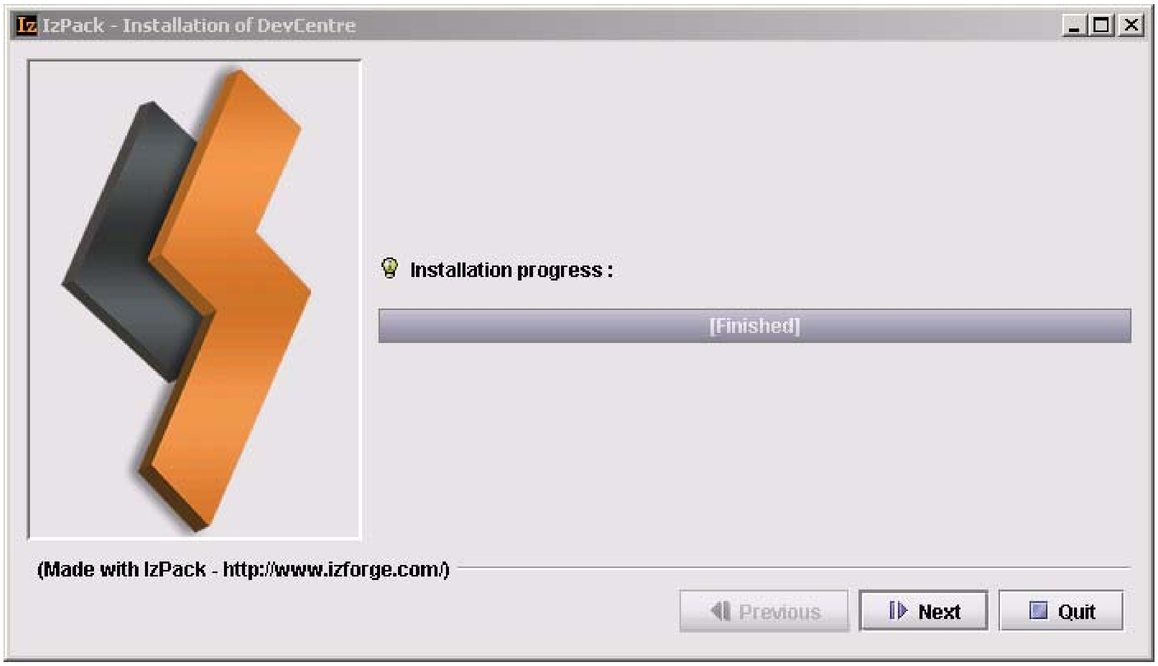
Installation progress :

C:\Program Files\DevCentre\ext\ant.jar

Core

(Made with IzPack - http://www.izforge.com/)

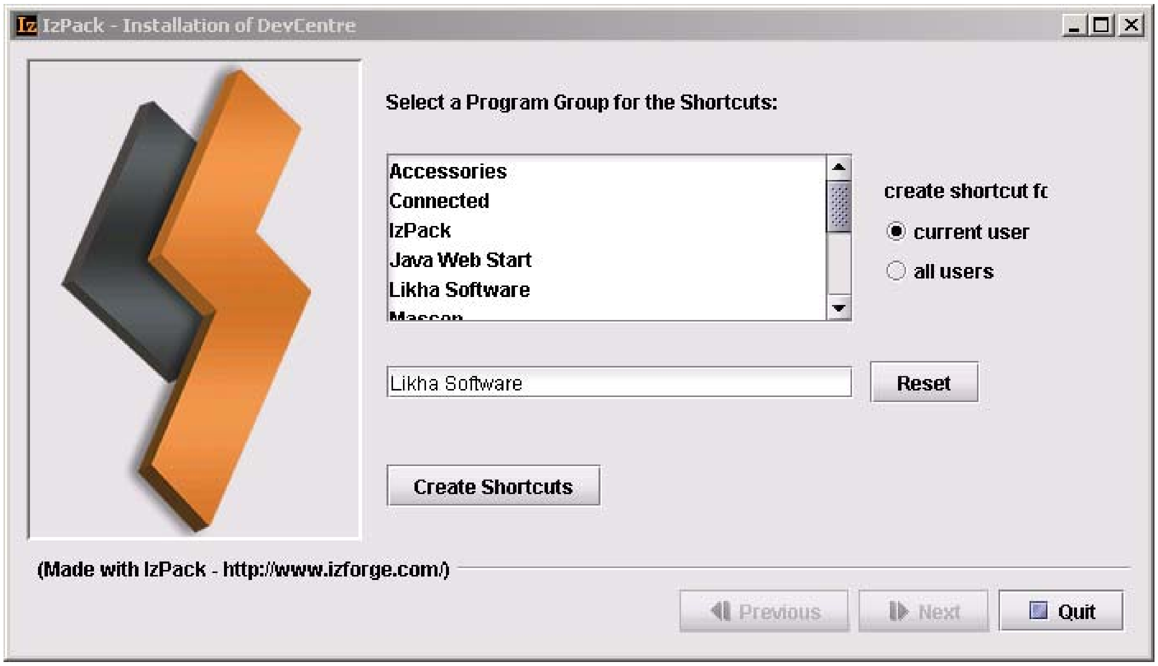
File copy is done.


IzPack - Installation of DevCentre
(Made with IzPack - http://www.izforge.com/)
Previous
Next
Quit

Installation progress :

[Finished]

Click “Create shortcuts”.



Select a Program Group for the Shortcuts:

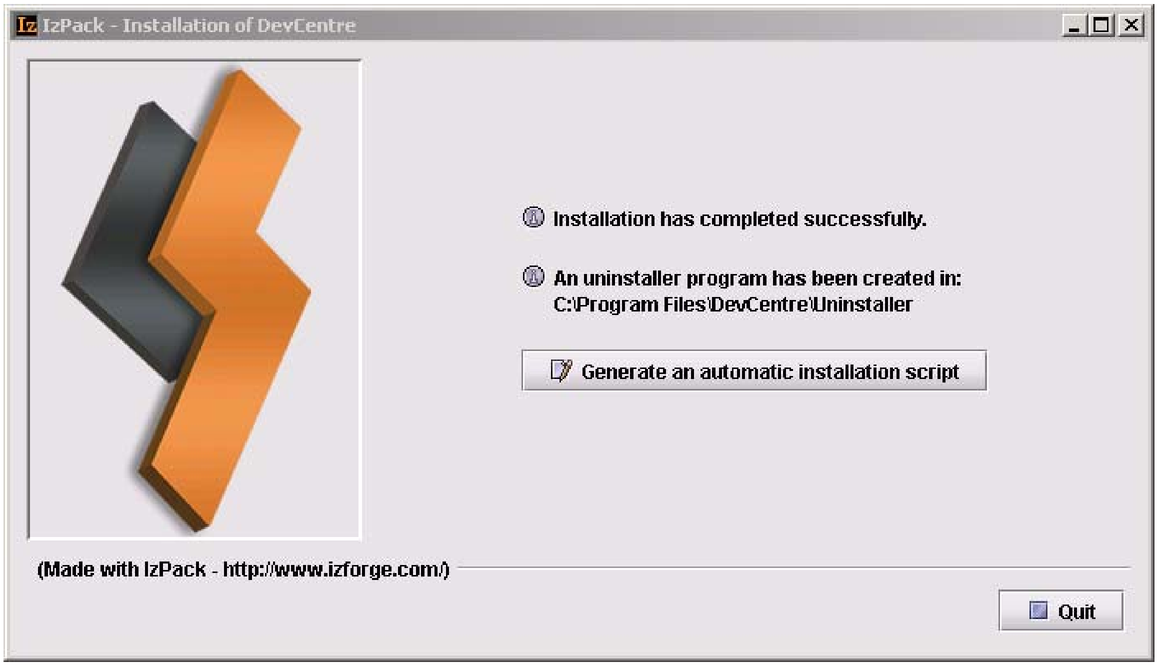
Likha Software

create shortcut fc

● current user

○ all users

Click “Quit”. It’s done.



IzPack - Installation of DevCentre

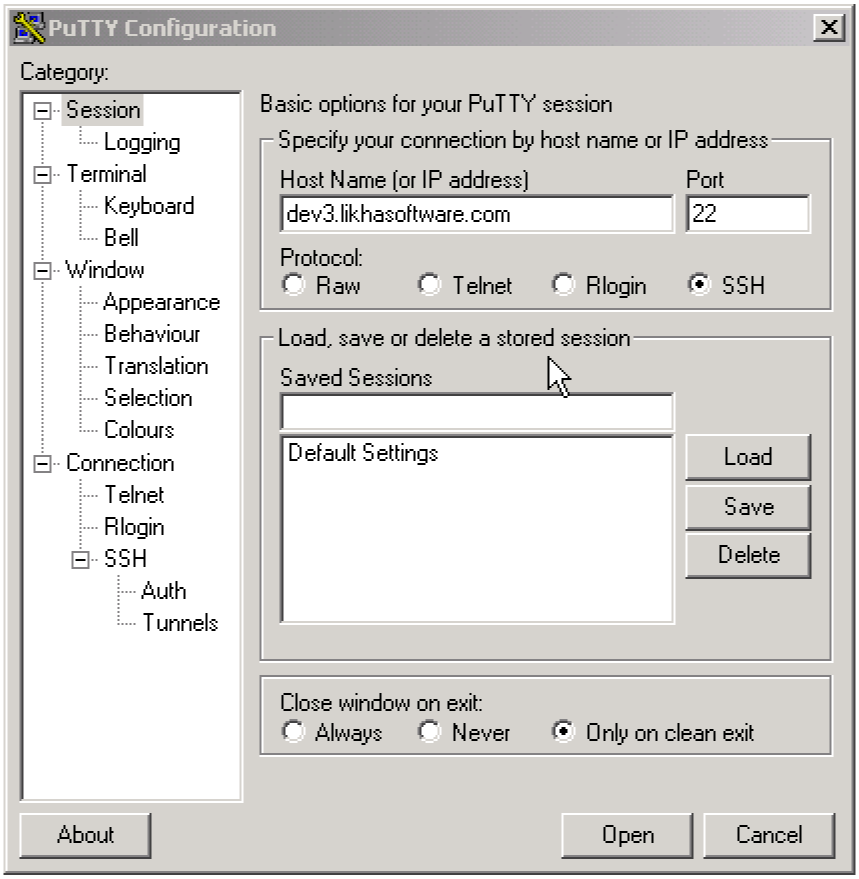
Installation has completed successfully.

An uninstaller program has been created in:
C:\Program Files\DevCentre\Uninstaller

Generate an automatic installation script
(Made with IzPack - http://www.izforge.com/)
Quit

III. Preparing Putty SSH Client

Select “Start” > “Programs” > “Likha Software” > “Putty SSH Client”. Enter the target server hostname. Select SSH as the connection protocol. Then click the “Open” button.



Basic options for your PuTTY session

Specify your connection by host name or IP address

Host Name (or IP address): dev3.likhasoftware.com

Port: 22

Protocol: ○ Raw ○ Telnet ○ Rlogin ● SSH

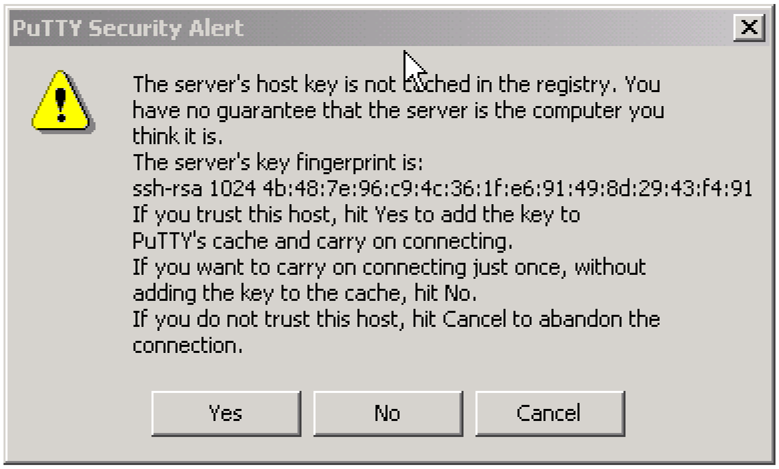
Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

IF the server host key is not cached in the registry, a message like below will display. Click “Yes” to cache the server key.

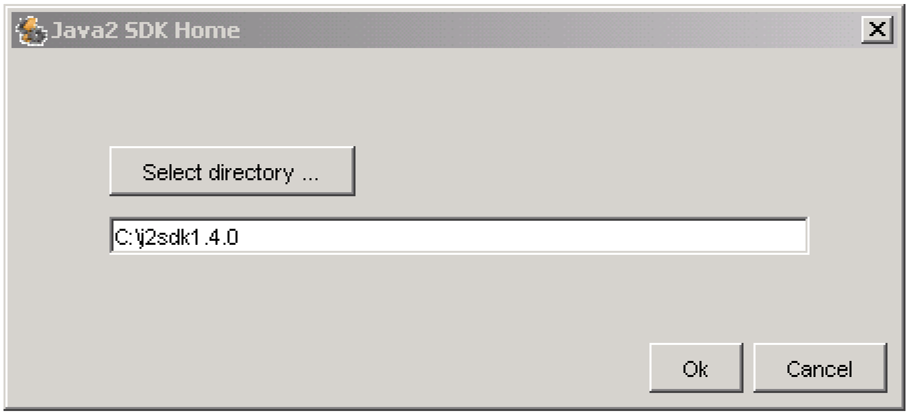


The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's key fingerprint is:
ssh-rsa 1024 4b:48:7e:96:c9:4c:36:1f:e6:91:49:8d:29:43:f4:91
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

IV. Creating your first web application project

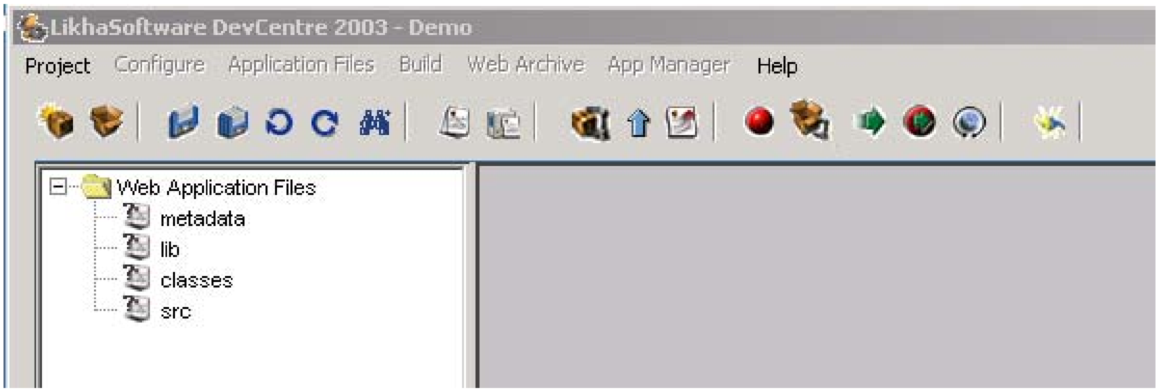
To run DevCentre IDE, click “Start button ” > “Programs” > “Likha Software” > “DevCentre 2003”.

The first fime DevCentre is run, it will prompt you to select your Java2 SDK Home.



C:\j2sdk1.4.0

Once, DevCentre finds the “bin/javac.exe” file it will open the main screen below.



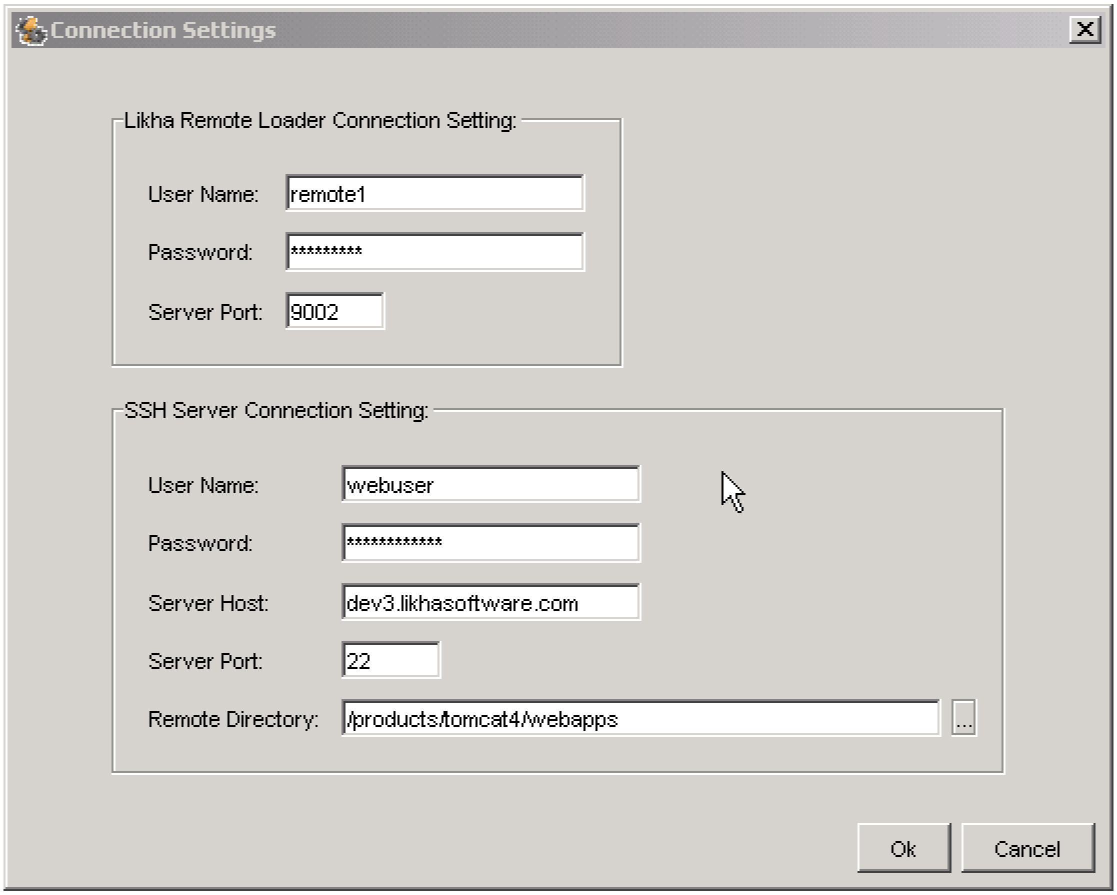
You need to create a new project to activate the rest of the menu items and toolbar shortcut icons.

Project Directory – This path specifies where you will create the web application file structure.

Archive Name – This specifies the name of the web archive to create. Specifying “myapp” will create “myapp.war”.

Remote Loader Connection Setting – This specifies your account in RALE. If you are using a demo version of DevCentre, just enter any value for username and password. The port value is RALE’s server port.

SSH Server Connection Setting – This specifies your Secure Shell connection account on the web server. The remote directory specifies where the web archive is stored on the server. Click the right button to retreive the remote directory info.



Likha Remote Loader Connection Setting:

User Name: remote1
Password: *********
Server Port: 9002

SSH Server Connection Setting:

User Name: webuser
Password: ************
Server Host: dev3.likhasoftware.com
Server Port: 22
Remote Directory: /products/tomcat4/webapps

For Likha Remote Loadeer Connection Setting:

Password is “password”.

For SSH Server Connection Setting:

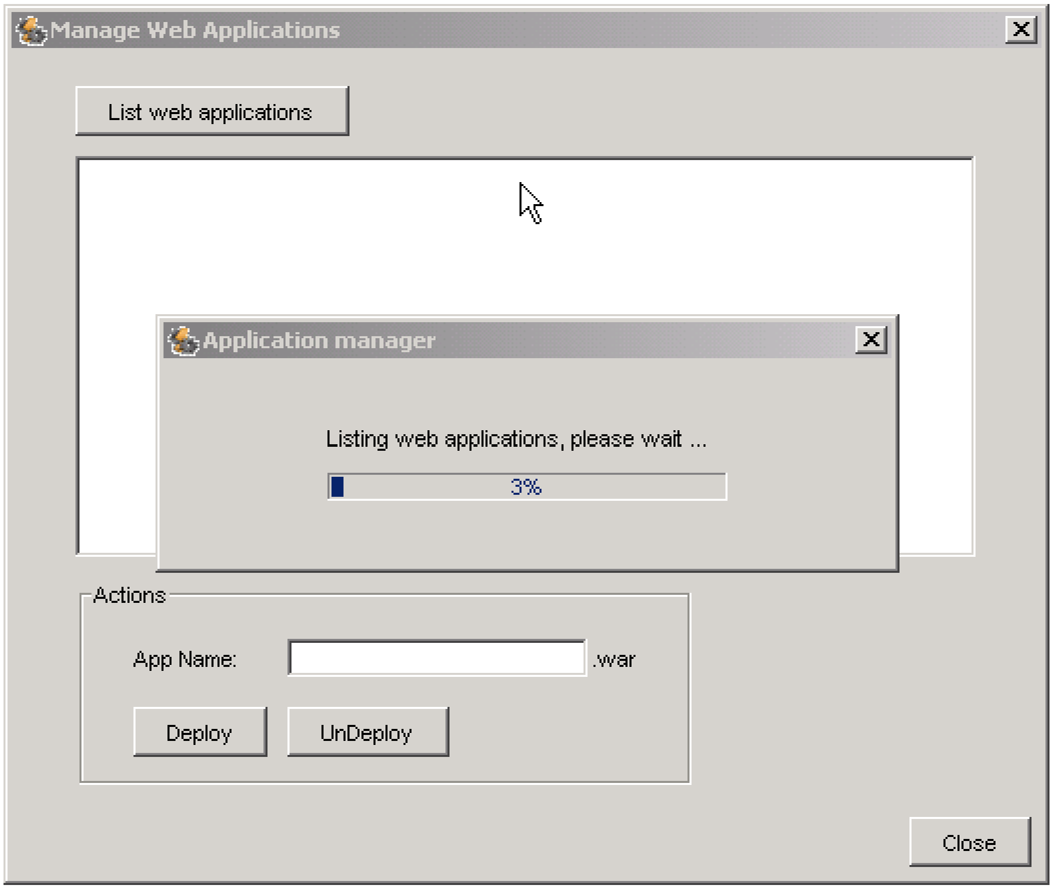
Password is “718293”.

Then, click “Ok” to create the project files. The connection setting is also stored in the project file directory structure.

There is a “sampleProject.zip” file included in this download. You can extract it and open it using Likha DevCentre. This has connection settings already set so you do not have to enter it.

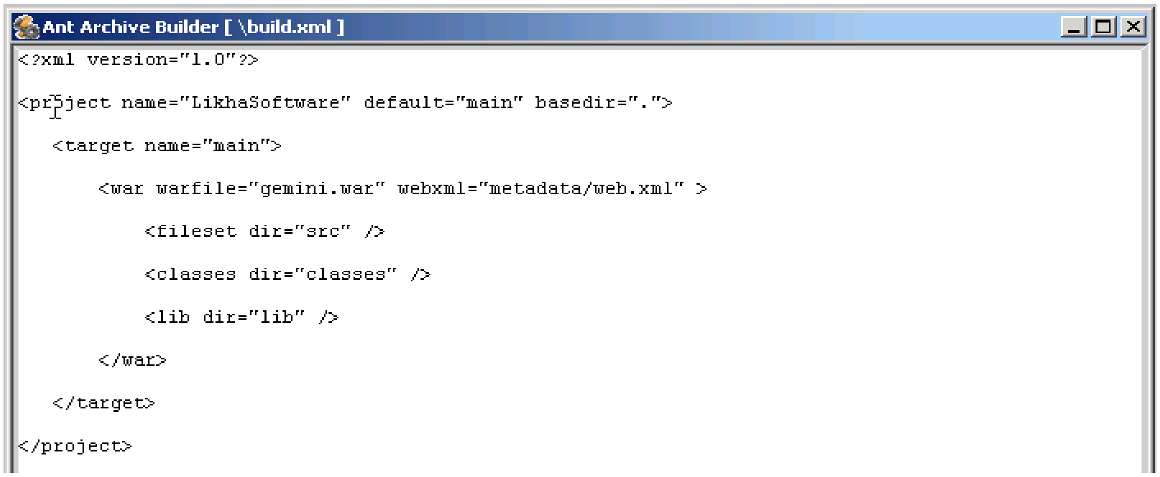
V. Creating a Dummy Web Application in Tomcat for your Project

Select “App Manager” > “Manage Applications”. Click “List web applications”.



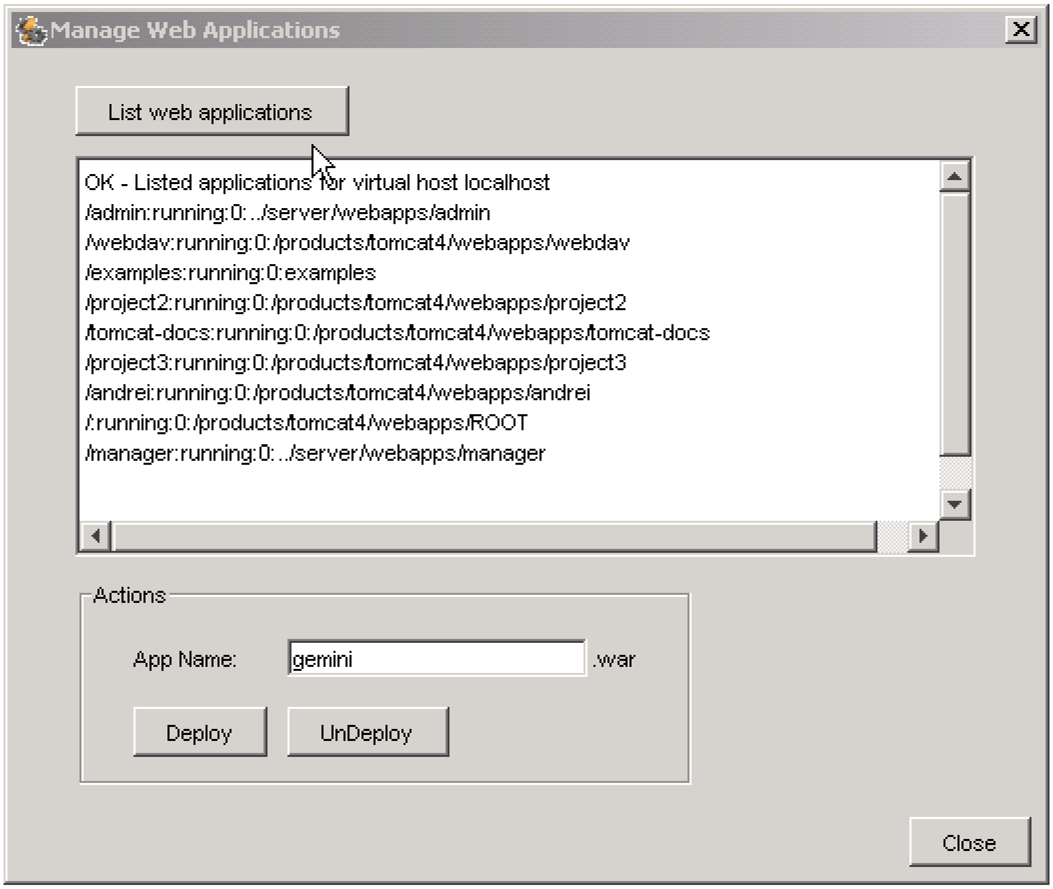
Listing web applications, please wait ...

A list of web applications will be display. Make sure the web application archive you specified is not listed. To verify, select “Configure” > “Ant Web Archive Builder”. This will display something similar below.



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

<project name="LikhaSoftware" default="main" basedir=".">

   <target name="main">

       <war warfile="gemini.war" webxml="metadata/web.xml" >

           <fileset dir="src" />

           <classes dir="classes" />

           <lib dir="lib" />

       </war>

   </target>

</project>
```

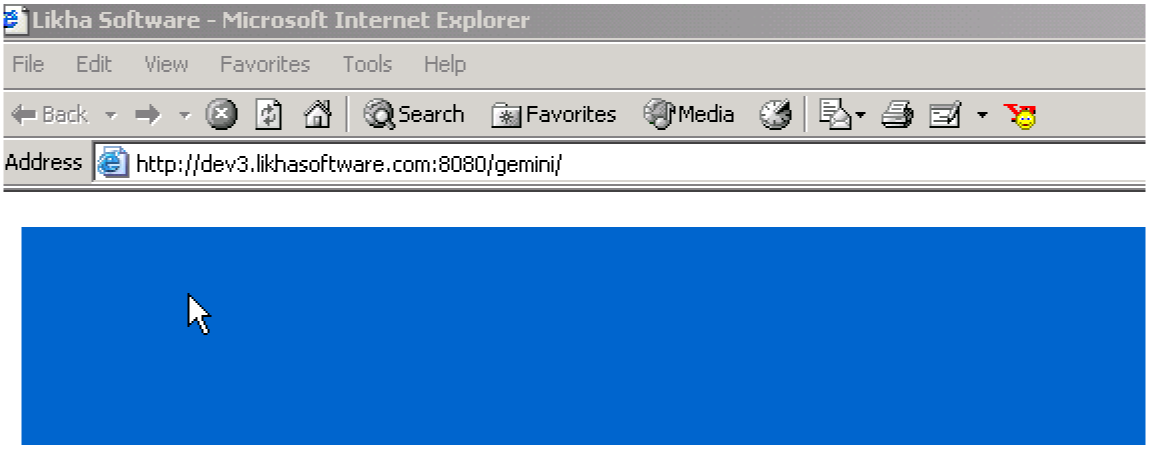
In this case, the web archive name is “gemini.war”. Make sure that “gemini” does not exist in the web application listing.



```
OK - Listed applications for virtual host localhost
/admin:running:0:../server/webapps/admin
/webdav:running:0:/products/tomcat4/webapps/webdav
/examples:running:0:examples
/project2:running:0:/products/tomcat4/webapps/project2
/tomcat-docs:running:0:/products/tomcat4/webapps/tomcat-docs
/project3:running:0:/products/tomcat4/webapps/project3
/andrei:running:0:/products/tomcat4/webapps/andrei
/:running:0:/products/tomcat4/webapps/ROOT
/manager:running:0:../server/webapps/manager
```

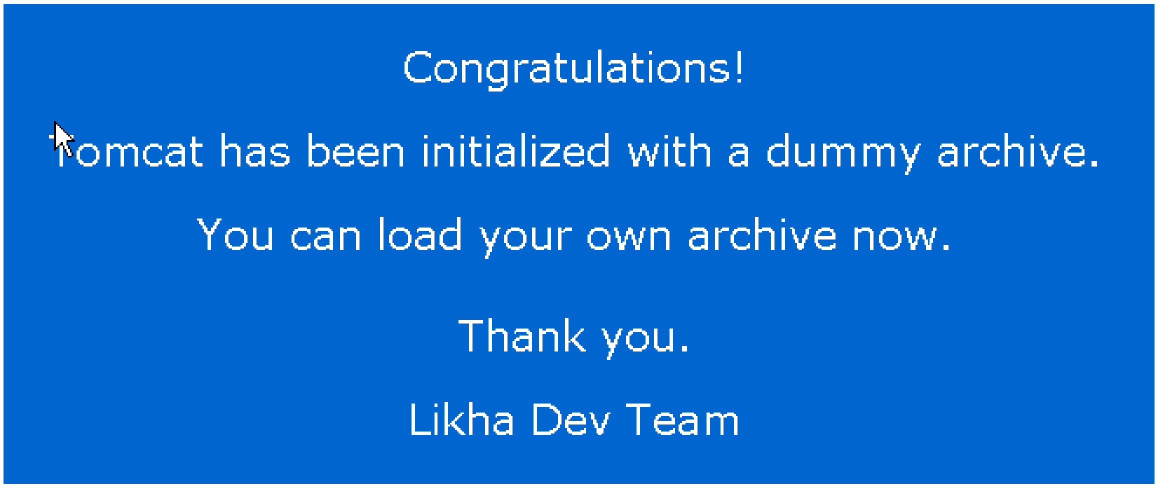
App Name: gemini .war

Enter “gemini” in the App Name text field, then, click “Deploy”. This will create a dummy archive in Tomcat. Verify that is has been deployed by opening the URL pointing to the “gemini” web application on your web browser.



Address http://dev3.likhasoftware.com:8080/gemini/

The message on the web browser might look similar to this.



Congratulations!

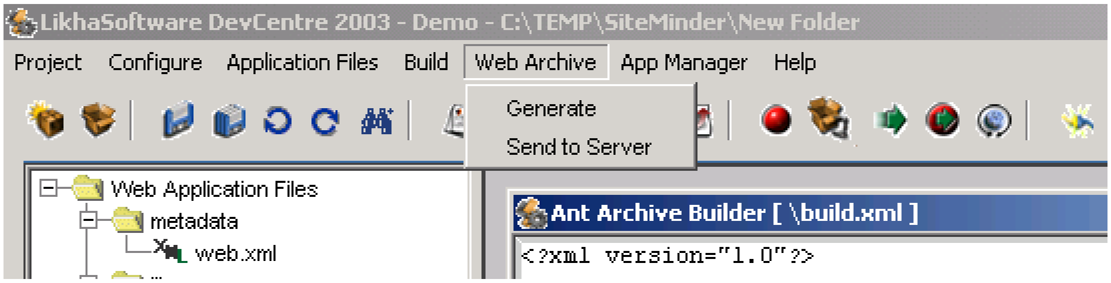
Tomcat has been initialized with a dummy archive.

You can load your own archive now.

Thank you.

Likha Dev Team

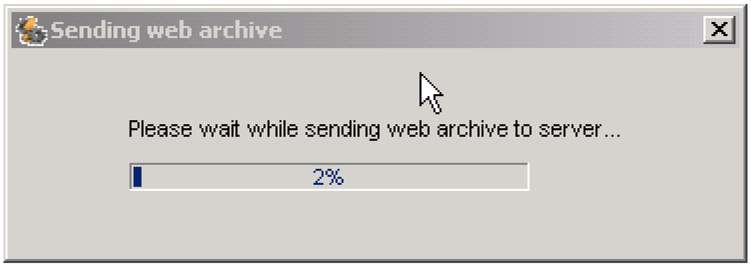
At this point, you can now remotely load the default web application to Tomcat. First, generate "gemini.war", then, send it to the server.



Generating the web archive will display a similar message below:

```
Removing current web archives.
C:\j2sdk1.4.0\bin\java -cp "C:\Program Files\DevCentre\ant\lib\ant.jar;C:\Program
Files\DevCentre\ant\lib\optional.jar;C:\Program Files\DevCentre\ant\lib\xercesImpl.jar;C:\Program
Files\DevCentre\ant\lib\xml-apis.jar"  -Dant.home="C:\Program Files\DevCentre\ant" org.apache.tools.ant.Main -f
"C:\TEMP\SiteMinder\New Folder\build.xml"
Buildfile: C:\TEMP\SiteMinder\New Folder\build.xml
main:
[war] Building war: C:\TEMP\SiteMinder\New Folder\gemini.war
BUILD SUCCESSFUL
Total time: 10 seconds
Ant build done
```

Sending the web archive will display a progress window.

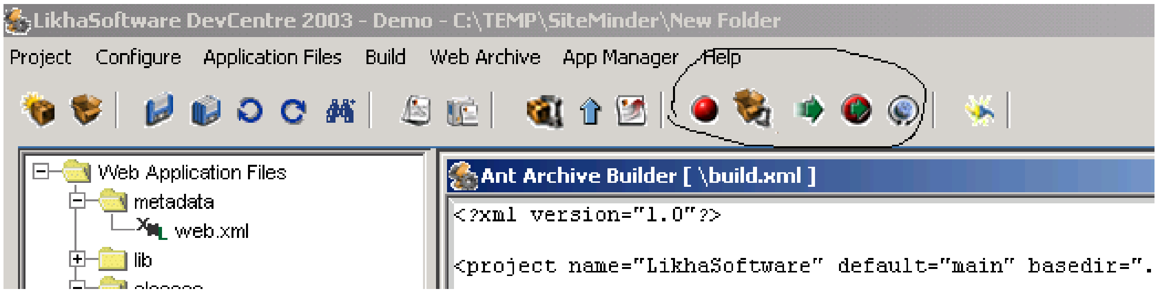


Please wait while sending web archive to server...

The message will look similar below.

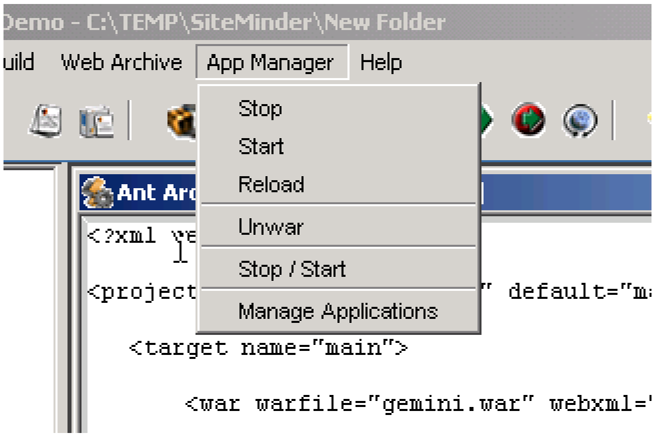
```
Sending C:\TEMP\SiteMinder\New Folder\gemini.war to dev3.likhasoftware.com
gemini.war        |      36 kB |  36.0 kB/s | ETA: 00:01:03 |   1%
gemini.war        |     272 kB |  27.2 kB/s | ETA: 00:01:15 |  11%
gemini.war        |    1792 kB |  26.7 kB/s | ETA: 00:00:19 |  77%
gemini.war        |    2124 kB |  26.5 kB/s | ETA: 00:00:07 |  91%
gemini.war        |    2316 kB |  26.6 kB/s | ETA: 00:00:00 | 100%
Sending done
```

The next steps are to “Stop” the web application on the server, “Unwar” the archive, and then “Start” it back. This will overwrite the dummy web application on the server. Use the toolbar shortcuts.



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

<project name="LikhaSoftware" default="main" basedir=".
```

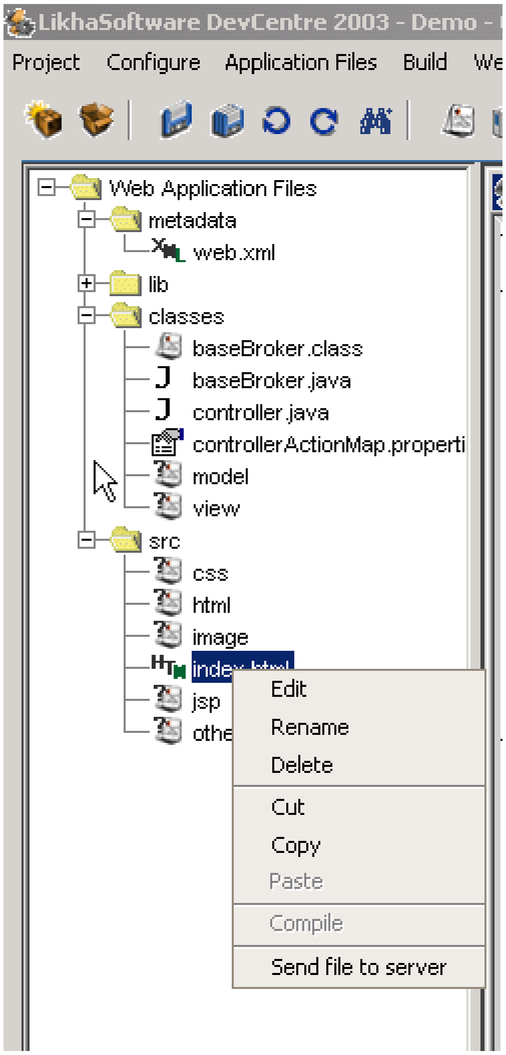
Or use the menu items …



```
<?xml ve

<project                " default="m

   <target name="main">

       <war warfile="gemini.war" webxml='
```

VI. Adding Applications files.

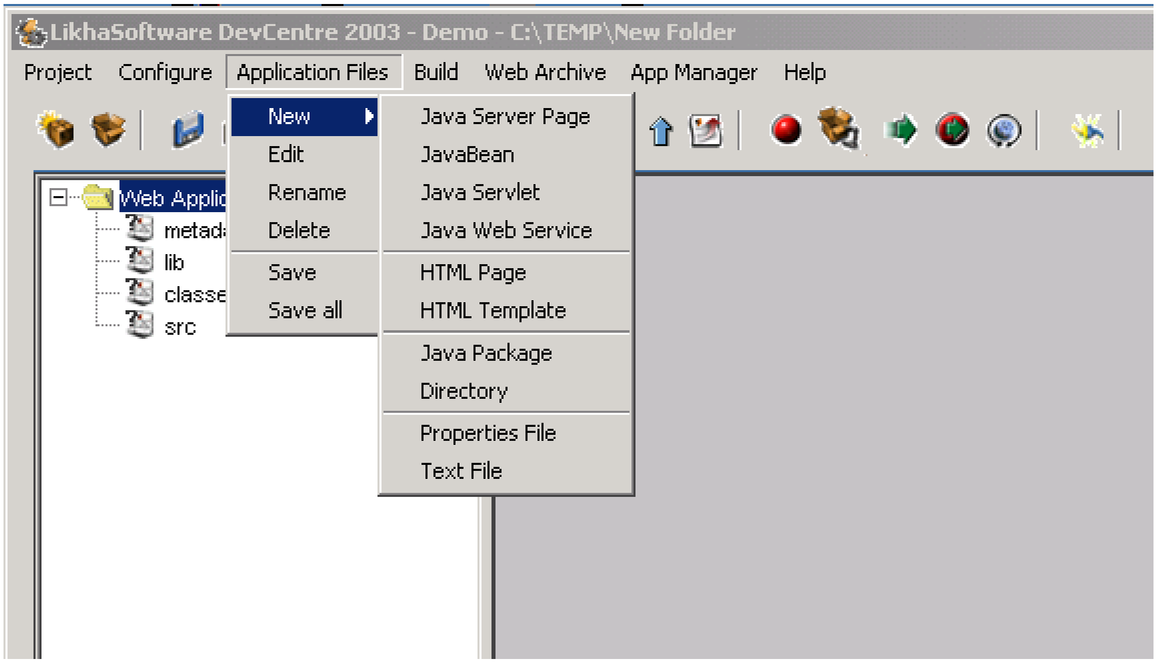
The file system tree reflects the file structure required by the initial Ant web archive **“build.XML”** file



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

<project name="LikhaSoftware" default="main" basedir=".">

   <target name="main">
       <war warfile="gemini.war" webxml="metadata/web.xml" >
           <fileset dir="src" />
           <classes dir="classes" />
           <lib dir="lib" />
       </war>
   </target>

</project>
```

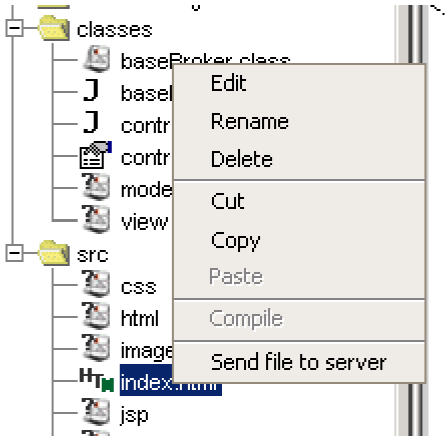
You may add additional Ant tasks but do not modify target name = “main” components because the IDE will look for those to generate the web archive. See the Ant user guide at http://jakarta.apache.org.

You can add new files using the menu “Application Files”.



You may send single files to the server after you have unwarred the original web archive because the local file structure has been reflected in the remote web application file structure.

Any changes or updates to the local file structure will require generating the web archives, sending it to the server and unwarring it to have the remote file structure updated.

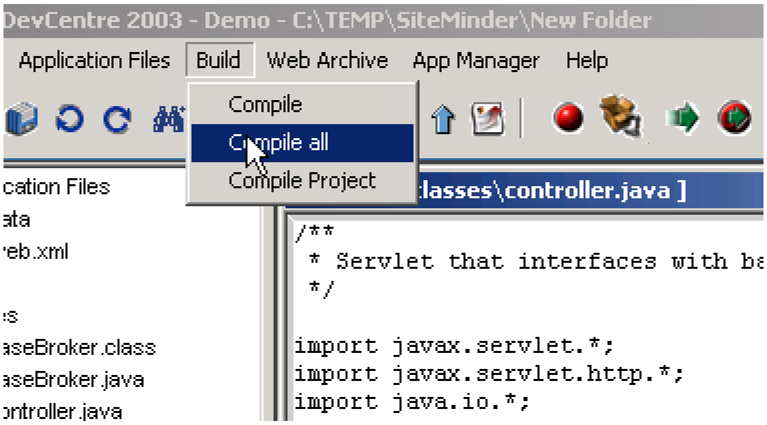


VII. Compiling Source Codes

You may compile the following file type:

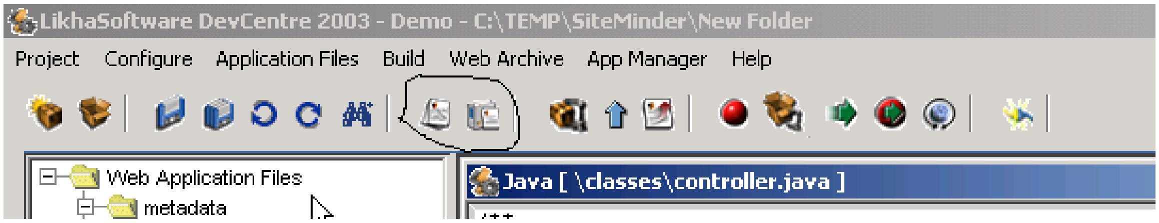
<filename>.java	- Java source codes
<filename>.jsp	- Java Server Pages
<filename>.jws	- Jakarta Axis Java Web Service

You can compile using the "Build" menu items if you have a selected compilable source code.

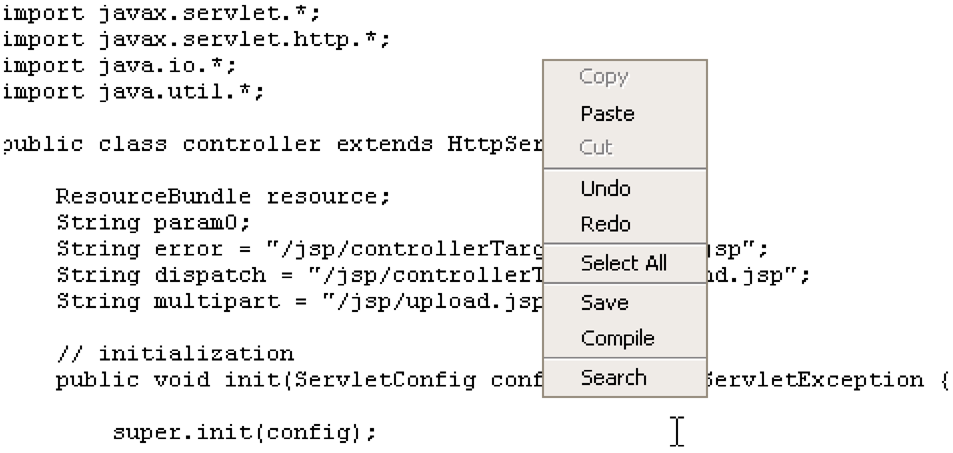


```
/**
 * Servlet that interfaces with be
 */

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
```

You can also compile using the toolbar "Compile" and "Compile All" icons below.


LikhaSoftware DevCentre 2003 - Demo - C:\TEMP\SiteMinder\New Folder
Project
Configure
Application Files
Build
Web Archive
App Manager
Help
Web Application Files
metadata
Java [\classes\controller.java]

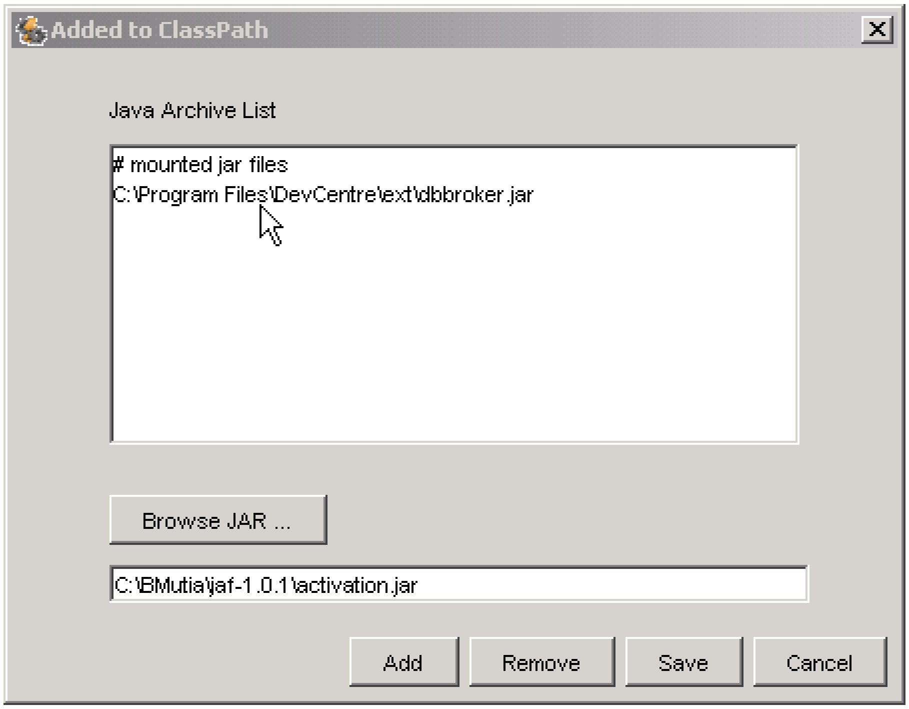
Or you can right click on the source code, and click "Compile".



```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class controller extends HttpSer

    ResourceBundle resource;
    String param0;
    String error = "/jsp/controllerTarg          sp";
    String dispatch = "/jsp/controllerT          d.jsp";
    String multipart = "/jsp/upload.jsp

    // initialization
    public void init(ServletConfig conf          ServletException {

        super.init(config);
```

VIII. Adding Jar file to the Class Path for Proper Compilation

Select “Project” > “Add Jar to ClassPath”. The window below will display.



```
# mounted jar files
C:\Program Files\DevCentre\ext\dbbroker.jar
```

```
C:\BMutia\jaf-1.0.1\activation.jar
```

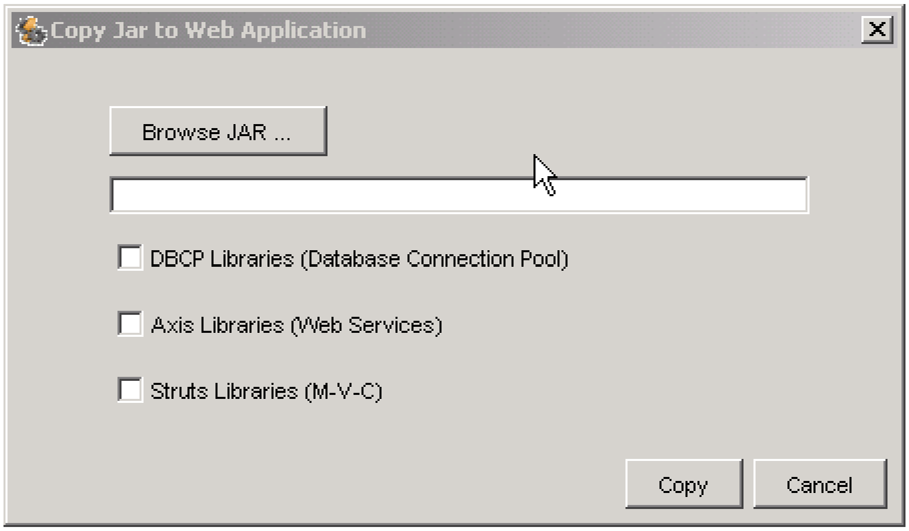
Then, browse a Jar file and the click “Add”. Click “Save” when you are done.

You may remove Jar file by pasting the Jar file name in the text field below and clicking “Remove”. Then click “Save”.

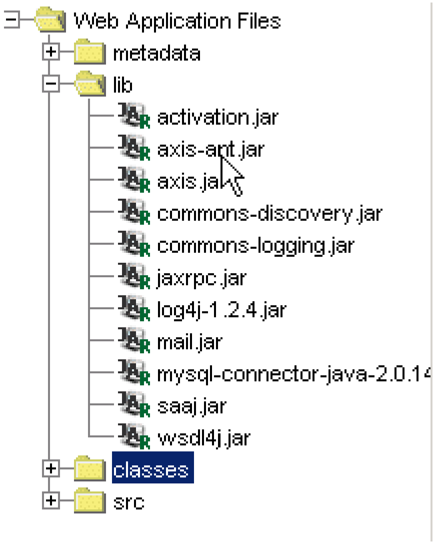
IX. Copying Jar files into the Web Application

You may browse for a Jar file, and then click “Copy”.

Or, select any of the check boxes to copy pre-selected libraries.


Copy Jar to Web Application
Browse JAR ...
☐ DBCP Libraries (Database Connection Pool)
☐ Axis Libraries (Web Services)
☐ Struts Libraries (M-V-C)
Copy
Cancel

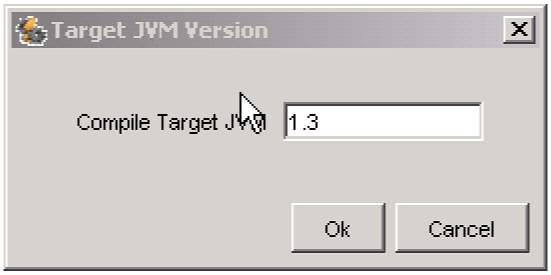
Copied Jar files are added to the web archive when it is generated. Also, these will appear in the “lib” system files.



X. Generating Class Files that Match the Web Server JVM

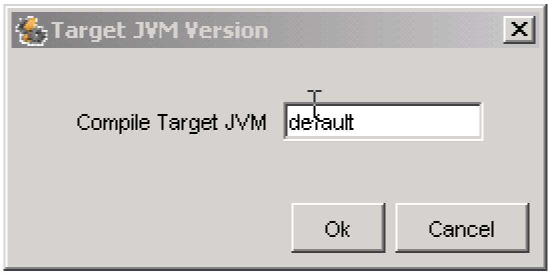
Sometimes the JVM version in the web server does not match the JVM in your workstation. So, when compiling you java source files, you want them compiled like the web server JVM version.

To do this, select “Project” > “Target JVM Version”. If your web server JVM is verion 1.3, then you would want the compile target to look like below.



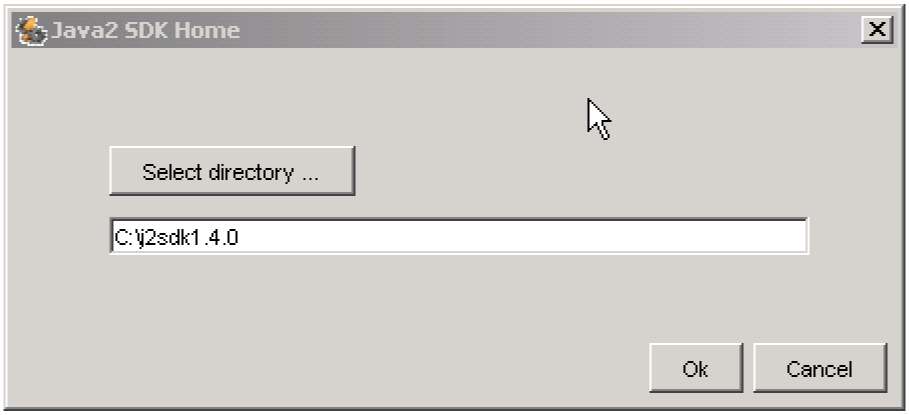
Compile Target JVM: 1.3

If you are sure that the web server JVM and your workstation JVM is the same, then, set it to “default”.



Compile Target JVM: default

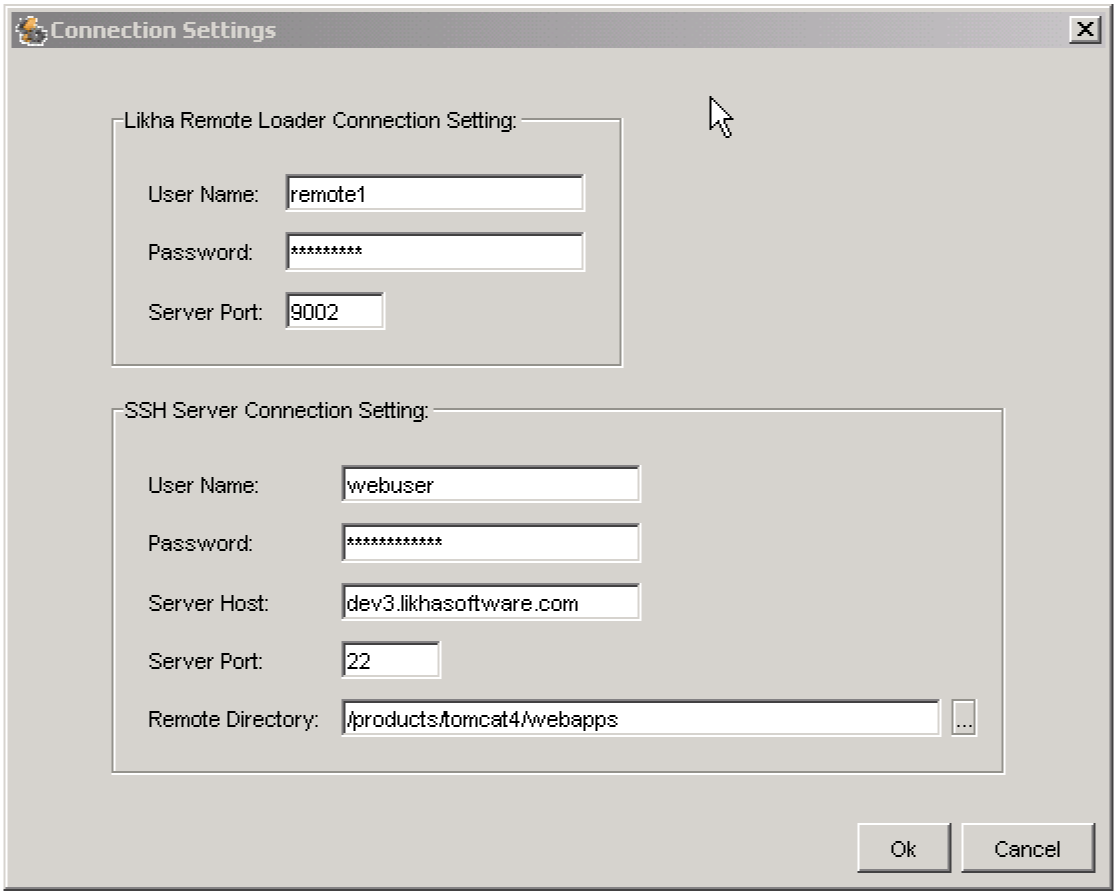
If you have installed a new Java2 SDK, your SDK home directory may have changed. To update the Java2 SDK home, select “Project” > “Java2 SDk Home”. A window will display like below.



C:\j2sdk1.4.0

XI. Changing Server Connection Settings

If you want to connect to a new server or just want to edit your server connection setting, select “Project” > “Server Connection”. A window will display like below.



Likha Remote Loader Connection Setting:

User Name: remote1
Password: *********
Server Port: 9002

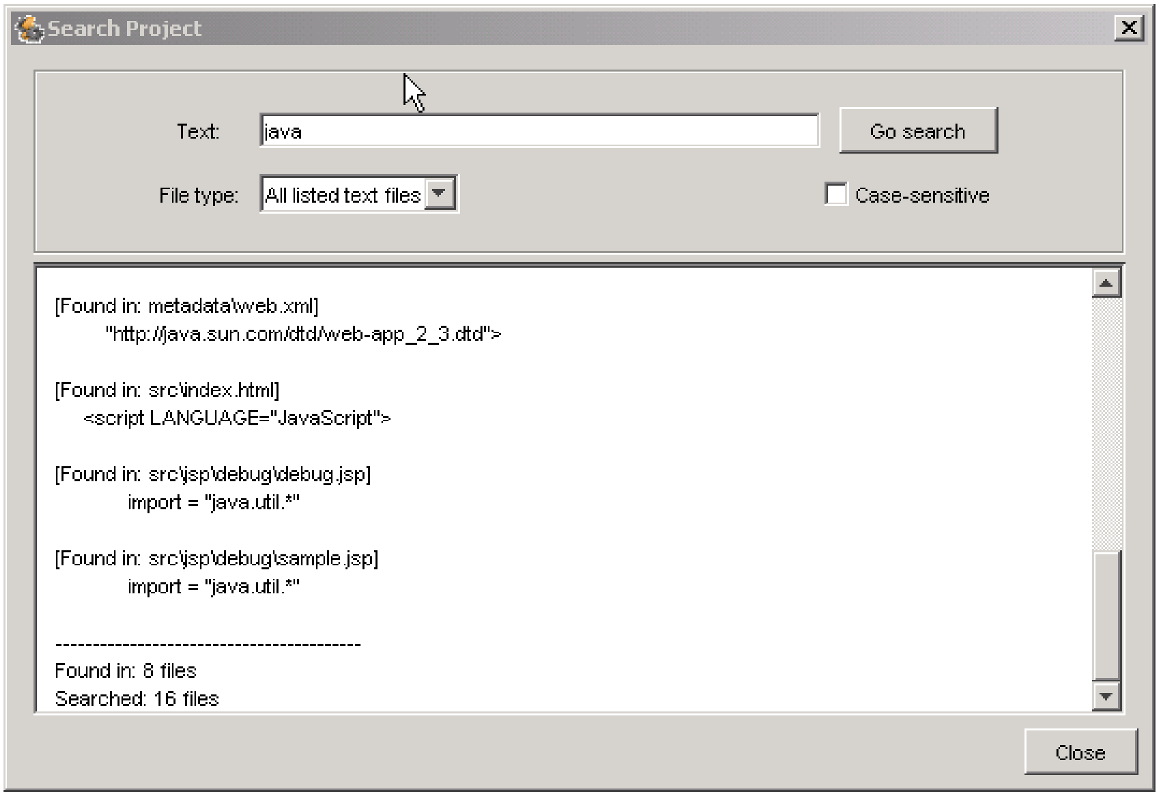
SSH Server Connection Setting:

User Name: webuser
Password: ************
Server Host: dev3.likhasoftware.com
Server Port: 22
Remote Directory: /products/tomcat4/webapps

XII. Searching Project Files

You may want to search project files for certain text. To do this, select “Project” > “Search Files”. A window like below will display.

Enter your text. Select a file type. Then click “Go search”.



Text: java

File type: All listed text files

☐ Case-sensitive

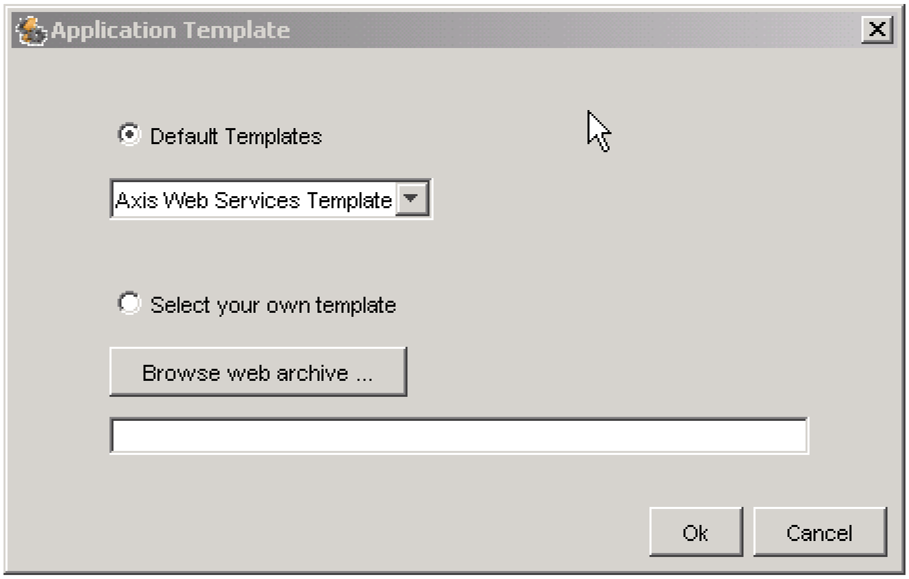
```
[Found in: metadata\web.xml]
        "http://java.sun.com/dtd/web-app_2_3.dtd">

[Found in: src\index.html]
    <script LANGUAGE="JavaScript">

[Found in: src\jsp\debug\debug.jsp]
           import = "java.util.*"

[Found in: src\jsp\debug\sample.jsp]
           import = "java.util.*"

-----------------------------------------
Found in: 8 files
Searched: 16 files
```

XIII. Creating your Project Files from a Web Archive or Template

Sometimes somebody else has a web application archive (with source files) that you want to pattern your own project.

Select "Project" > "Web Archive Template". There are some default templates that you can use. Or, select your own web archive.

Automatically, the web archive will be unwarred and copied over to your project. The previous project files will be removed and placed in the project "workarea" directory.



● Default Templates

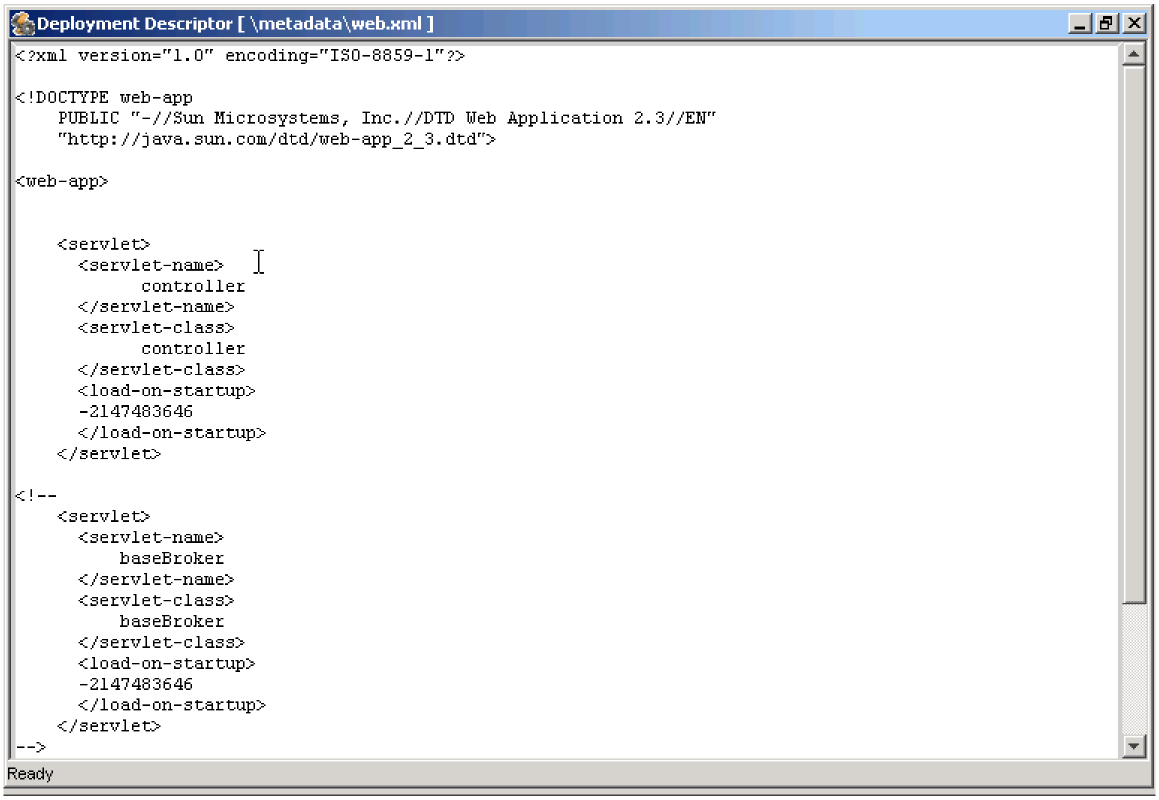
Axis Web Services Template

○ Select your own template

XIV. Editing the Web Archive Deployment Descriptor

Editing this file requires knowledge of the Jakarta Tomcat web application descriptor documentation. Please refer to http://jakarta.apache.org.

Select “Configure” > “Deployment Descriptor”.



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

<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>


    <servlet>
      <servlet-name>
            controller
      </servlet-name>
      <servlet-class>
            controller
      </servlet-class>
      <load-on-startup>
      -2147483646
      </load-on-startup>
    </servlet>

<!--
    <servlet>
      <servlet-name>
          baseBroker
      </servlet-name>
      <servlet-class>
          baseBroker
      </servlet-class>
      <load-on-startup>
      -2147483646
      </load-on-startup>
    </servlet>
-->
```

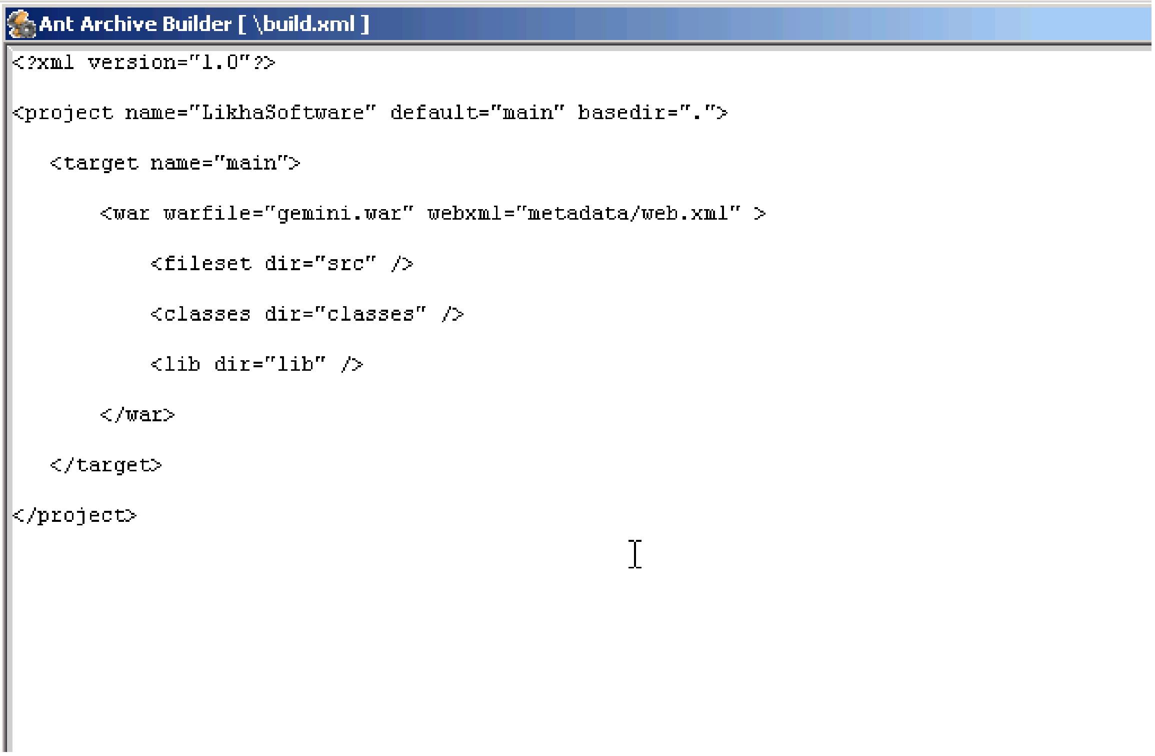
The default deployment descriptor only includes two servlets. The “controller” and the “baseBroker”. Only the controller servlet is enabled. The “baseBroker” is commented out.

The “baseBroker” is based on http://www.javaexchange.com database connection pool servlet.

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

<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

    <servlet>
      <servlet-name>
            controller
      </servlet-name>
      <servlet-class>
            controller
      </servlet-class>
      <load-on-startup>
         -2147483646
      </load-on-startup>
    </servlet>

<!--
    <servlet>
      <servlet-name>
          baseBroker
      </servlet-name>
      <servlet-class>
          baseBroker
      </servlet-class>
      <load-on-startup>
         -2147483646
      </load-on-startup>
    </servlet>
-->

    <servlet-mapping>
        <servlet-name>invoker</servlet-name>
        <url-pattern>/servlet/*</url-pattern>
    </servlet-mapping>

</web-app>
```

XV. Editing Ant's Web Archive Builder File

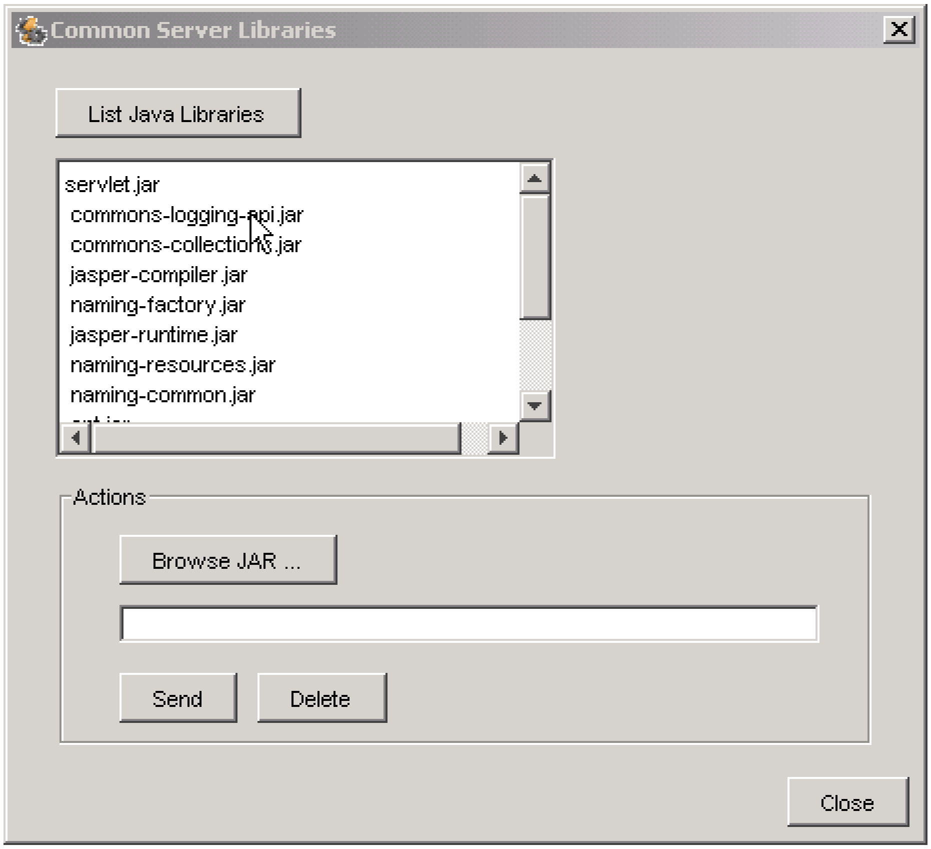
Select "Configure" > "Ant Web Archive Builder". Editing this file requires knowledge in Ant tasks. Please refer to http://jakarta.apache.org for more documentation.



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

<project name="LikhaSoftware" default="main" basedir=".">

   <target name="main">

       <war warfile="gemini.war" webxml="metadata/web.xml" >

           <fileset dir="src" />

           <classes dir="classes" />

           <lib dir="lib" />

       </war>

   </target>

</project>
```

XVI. Viewing, Uploading Deleting Common Server Jar Libraries

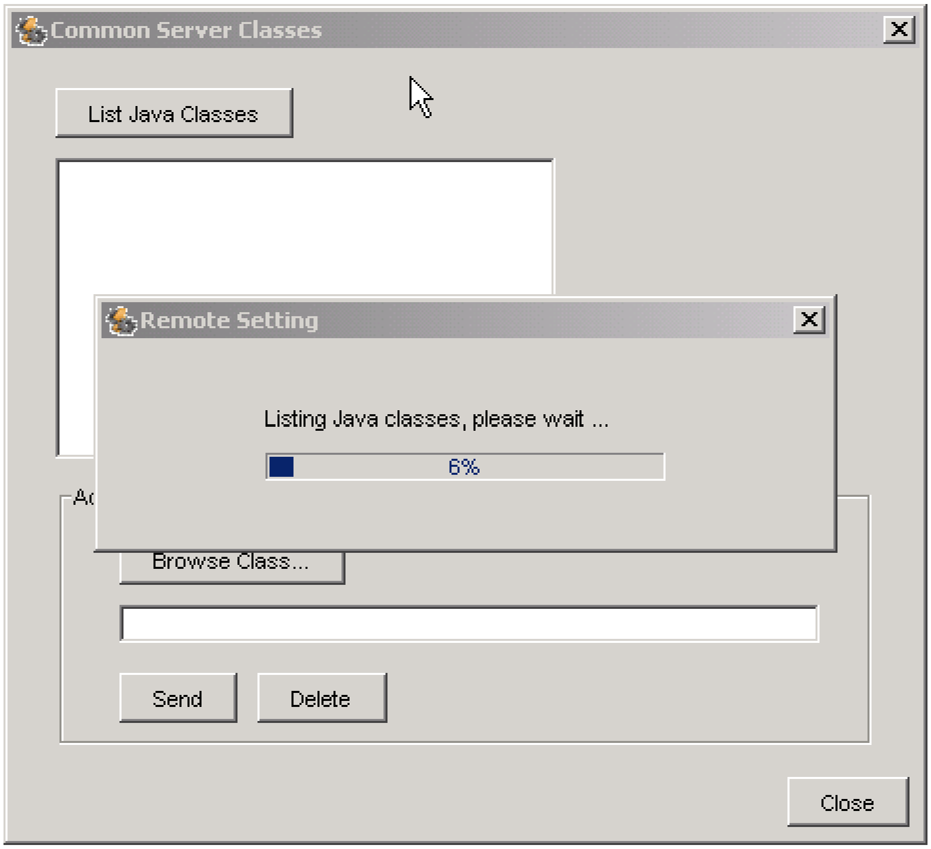
Select “Configure” > “Server Common Libraries”. You can select a Jar file to upload or delete remotely. Any change to this list requires a restart of the Tomcat server.



servlet.jar
commons-logging-api.jar
commons-collections.jar
jasper-compiler.jar
naming-factory.jar
jasper-runtime.jar
naming-resources.jar
naming-common.jar

XVII. Viewing, Uploading Deleting Common Server Classes

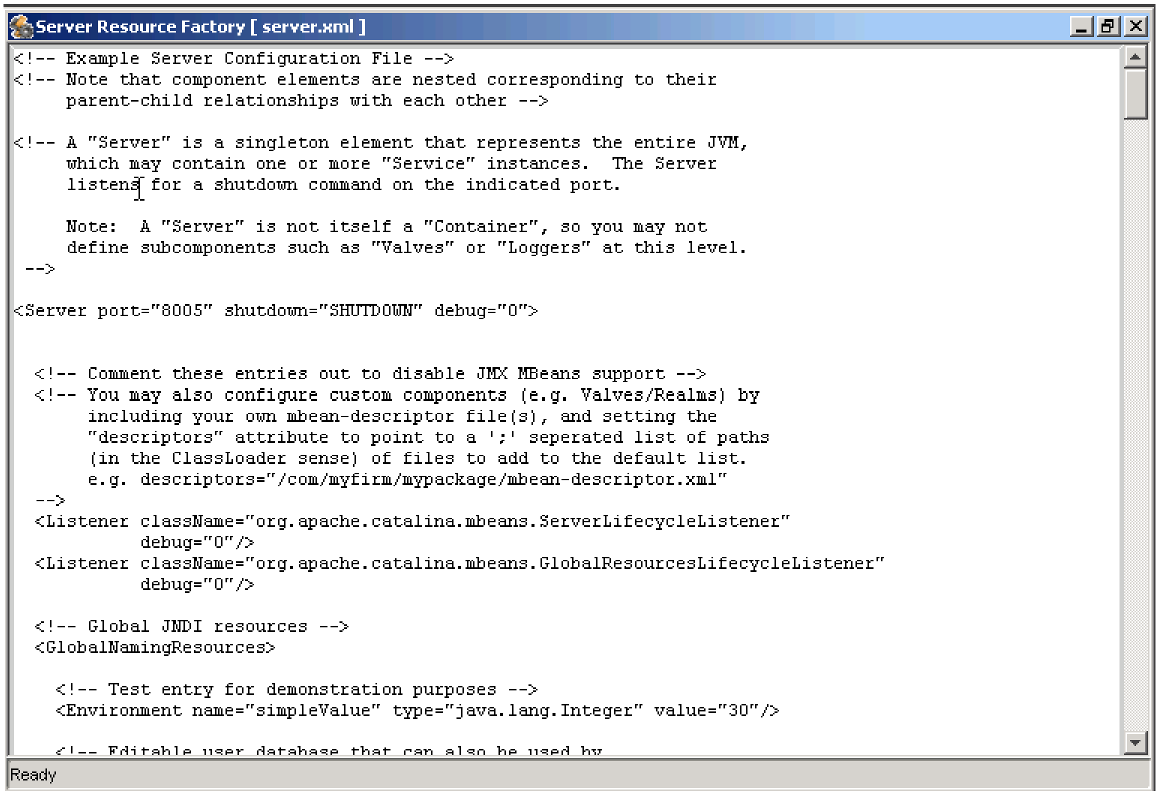
Select "Configure" > "Server Common Classes". You can select a class file to upload or delete remotely. Any change to this list requires a restart of the Tomcat server.



Listing Java classes, please wait ...

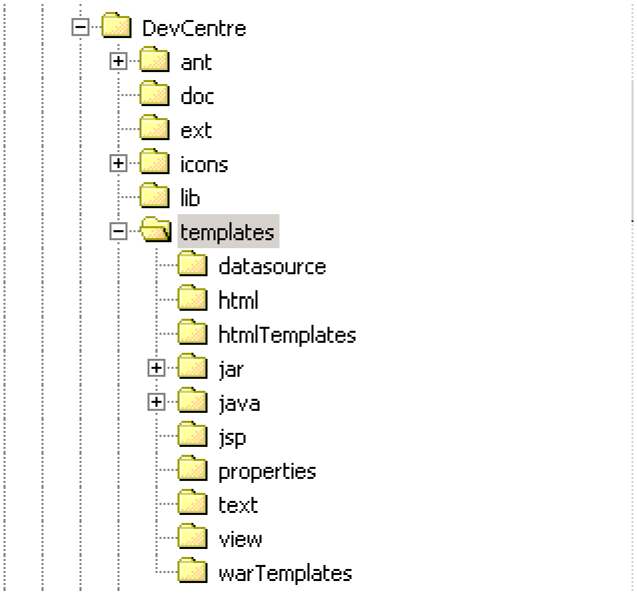
XVIII. Viewing the Server Resource Factory

Select “Configure” > “Server Resource Factory”. Understanding this file requires knowledge in Jakarta Tomcat configuration. Please refer to http://jakarta.apache.org for more documentation.



```
<!-- Example Server Configuration File -->
<!-- Note that component elements are nested corresponding to their
     parent-child relationships with each other -->

<!-- A "Server" is a singleton element that represents the entire JVM,
     which may contain one or more "Service" instances.  The Server
     listens for a shutdown command on the indicated port.

     Note:  A "Server" is not itself a "Container", so you may not
     define subcomponents such as "Valves" or "Loggers" at this level.
 -->

<Server port="8005" shutdown="SHUTDOWN" debug="0">


  <!-- Comment these entries out to disable JMX MBeans support -->
  <!-- You may also configure custom components (e.g. Valves/Realms) by
       including your own mbean-descriptor file(s), and setting the
       "descriptors" attribute to point to a ';' seperated list of paths
       (in the ClassLoader sense) of files to add to the default list.
       e.g. descriptors="/com/myfirm/mypackage/mbean-descriptor.xml"
  -->
  <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener"
            debug="0"/>
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"
            debug="0"/>

  <!-- Global JNDI resources -->
  <GlobalNamingResources>

    <!-- Test entry for demonstration purposes -->
    <Environment name="simpleValue" type="java.lang.Integer" value="30"/>

    <!-- Editable user database that can also be used by
```

XIX. Creating Your Own Code Templates used by DevCentre



You can add, edit, delete files in the “templates” subdirectories where you installed Likha DevCentre. Do not change the directory names. All changes will be reflected in the “Application Files” > “New” selections.