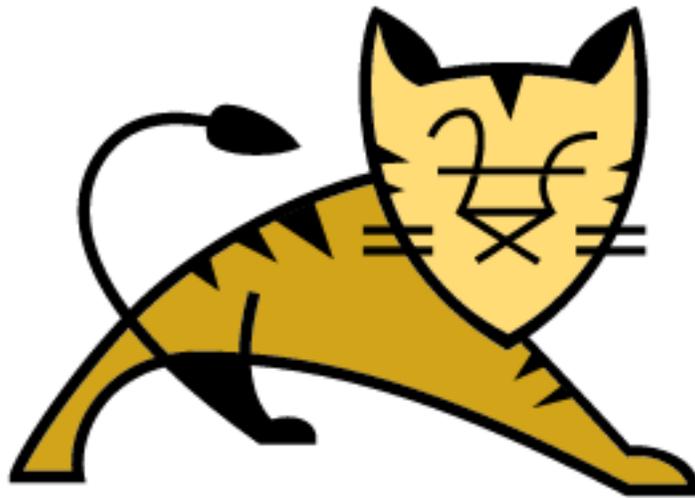


**On-campus
Tomcat Deployment Client**



User Guide

The Department of Computer Science

Southern Illinois University of Carbondale

Table of Contents

On-campus Tomcat Deployment.....	3
Working with deployment templates.....	4
JSP Template Configuration.....	6
Configuring your JSP servlet web.xml file.....	7
Java Template Configuration.....	8
Configuring your Java servlet web.xml file.....	9
Compiling and Deploying .WAR files with ANT.....	10
Quick Reference.....	11

On-campus Tomcat Deployment

The following guide will outline how end-users can utilize Apache Ant to compile java code into a packaged .WAR archive and deploy this archive to the CS Department Tomcat server. The Apache Ant package management utility will utilize the Tomcat Client Deployer to deploy the compiled .WAR file.

Please note, CS Department computers should already be configured with these packages. This guide is intended for computers located on-campus in the cs computer labs.

Outline of resources being used:

1. Ant v1.9.5 binaries (Java JAR & WAR Packaging utility)
2. Tomcat Client Deployer v8.0.23 (Tomcat WAR deployer)
[Requires Ant Version 1.6.2+]
3. Sample JSP and Servlet template deployments (.zip archives)
4. Java SE JDK 1.7.0+ is recommended. (Java SE JDK 1.8 currently appears to be supported). Java JDK binaries need to be available in your PATH environment variable.
5. Apache Ant [Version 1.6.2+, currently v1.9.5 is the latest] must be installed. Ant binaries must be located in the PATH environment variable.

Working with deployment templates

The following templates work with both windows and linux computers.

Standard Directory Layout

To facilitate creation of a Web Application Archive (.WAR) file in the required format, it is convenient to arrange the "executable" files of your web application (that is, the files that Tomcat actually uses when executing your app) in the same organization as required by the WAR format itself. To do this, you will end up with the following contents in your application's "document root" directory:

Document root path: `sample-deploy/<your_login_ID>/`

***.html, *.jsp, etc.** - The HTML and JSP pages, along with other files that must be visible to the client browser (such as JavaScript, stylesheet files, and images) for your application. In larger applications you may choose to divide these files into a subdirectory hierarchy, but for smaller apps, it is generally much simpler to maintain only a single directory for these files.

/WEB-INF/web.xml - The Web Application Deployment Descriptor for your application. This is an XML file describing the servlets and other components that make up your application, along with any initialization parameters and container-managed security constraints that you want the server to enforce for you. This file is discussed in more detail in the following subsection.

/WEB-INF/classes/ - This directory contains any Java class files (and associated resources) required for your application, including both servlet and non-servlet classes, that are not combined into JAR files. If your classes are organized into Java packages, you must reflect this in the directory hierarchy under `/WEB-INF/classes/`. For example, a Java class named `com.mycompany.mypackage.MyServlet` would need to be stored in a file named `/WEB-INF/classes/com/mycompany/mypackage/MyServlet.class`.

/WEB-INF/lib/ - This directory contains JAR files that contain Java class files (and associated resources) required for your application, such as third party class libraries or JDBC drivers.

When you install an application into Tomcat (or any other 2.2 or later Servlet container), the classes in the `WEB-INF/classes/` directory, as well as all classes in JAR files found in the `WEB-INF/lib/` directory, are made visible to other classes within your particular web application. Thus, if you include all of the required library classes in one of these places (be sure to check licenses for redistribution rights for any third party libraries you utilize),

you will simplify the installation of your web application -- no adjustment to the system class path (or installation of global library files in your server) will be necessary.

(Source: <http://tomcat.apache.org/tomcat-8.0-doc/appdev/deployment.html>)

```
Sample-deploy/
├── build.xml (Ant build file, don't edit)
├── lib/ (Tomcat Deployer Client .jar files)
│   ├── catalina-ant.jar
│   ├── catalina-deployer.jar
│   ├── commons-beanutils.jar
│   ├── commons-collections.jar
│   ├── commons-digester.jar
│   ├── commons-el.jar
│   ├── commons-logging.jar
│   ├── el-api.jar
│   ├── jasper-compiler.jar
│   ├── jasper-el.jar
│   ├── jasper.jar
│   ├── jasper-runtime.jar
│   ├── jsp-api.jar
│   ├── servlet-api.jar
│   ├── tomcat-juli.jar
│   └── xmlParserAPIs.jar
├── your_login_ID/ [Change this directory name to your username]
│   ├── HelloWorld.jsp [.html and .jsp files go here]
│   └── WEB-INF/
│       ├── classes/ [.java files go here]
│       ├── lib/ [External .jar libraries]
│       │   ├── mysql-connector-java-5.1.35-bin.jar
│       │   └── ojdbc6_g.jar
│       └── web.xml [Change according to your deployment]

```

5 directories, 21 files

JSP Template Configuration

Download the following file and extract the following file:

<http://cs.siu.edu/download/tomcat/sample-deploy-jsp.zip>

```
Sample-deploy-jsp/
├── build.xml (Ant build file, don't edit)
├── lib/ (Tomcat Deployer Client .jars)
│   ├── catalina-ant.jar
│   ├── catalina-deployer.jar
│   ├── commons-beanutils.jar
│   ├── commons-collections.jar
│   ├── commons-digester.jar
│   ├── commons-el.jar
│   ├── commons-logging.jar
│   ├── el-api.jar
│   ├── jasper-compiler.jar
│   ├── jasper-el.jar
│   ├── jasper.jar
│   ├── jasper-runtime.jar
│   ├── jsp-api.jar
│   ├── servlet-api.jar
│   ├── tomcat-juli.jar
│   └── xmlParserAPIs.jar
├── your_login_ID/ [Change this directory name to your username]
│   ├── HelloWorld.jsp [.html and .jsp files go here]
│   └── WEB-INF/
│       ├── classes/
│       ├── lib/ [External .jar libraries]
│       │   ├── mysql-connector-java-5.1.35-bin.jar
│       │   └── ojdbc6_g.jar
│       └── web.xml [Change according to your deployment]
5 directories, 21 files
```

Important:

You will later be using ant to build this template into a .WAR file. This will require that you upload your code to the above directories, configure the web.xml file for your code. You'll also need to remember the absolute path for the sample-deploy directory extracted in this step for later compilation purposes.

Configuring your JSP servlet web.xml file

You'll need to modify the "web.xml" file for different jsp & servlet files. web.xml is located at sample-deploy\your_login_ID\WEB-INF\web.xml.

JSP example, subsection from WEB-INF/web.xml:

```
<servlet>
  <!-- Tomcat internal naming convention -->
  <servlet-name>HelloWorld</servlet-name>
  <!-- Servlet Class can be defined many ways -->
  <!-- <servlet-class>org.apache.jsp.HelloWorld_jsp</servlet-class> -->
  <!-- <servlet-class>org.apache.jsp.HelloWorld</servlet-class>
  <servlet-class>HelloWorld</servlet-class>
  <load-on-startup>3</load-on-startup>
</servlet>
<servlet-mapping>
  <!-- Tomcat internal naming convention -->
  <!-- Must match above servlet-name -->
  <servlet-name>HelloWorld</servlet-name>
  <url-pattern>*.jsp</url-pattern>
</servlet-mapping>
```

In the example above `<servlet-class>HelloWorld</servlet-class>` needs to correspond to your `HelloWorld.jsp` file being deployed.

Both fields `<servlet-name></servlet-name>` must have matching values. i.e., `<servlet-name>HelloWorld</servlet-name>`

After deploying this example it would be viewable at:
`http://web.cs.siu.edu:8080/your_login_ID/HelloWorld.jsp`

Additionally, <http://www.cs.siu.edu/download/tomcat/web.xml> contains a more complete sample web.xml file for reference.

Java Template Configuration

Download the following file and extract the following file:

<http://cs.siu.edu/download/tomcat/sample-deploy-servlet.zip>

```
Sample-deploy-servlet/
├── build.xml (Ant build file, don't edit)
├── lib/ (Tomcat Deployer Client .jars)
│   ├── catalina-ant.jar
│   ├── catalina-deployer.jar
│   ├── commons-beanutils.jar
│   ├── commons-collections.jar
│   ├── commons-digester.jar
│   ├── commons-el.jar
│   ├── commons-logging.jar
│   ├── el-api.jar
│   ├── jasper-compiler.jar
│   ├── jasper-el.jar
│   ├── jasper.jar
│   ├── jasper-runtime.jar
│   ├── jsp-api.jar
│   ├── servlet-api.jar
│   ├── tomcat-juli.jar
│   └── xmlParserAPIs.jar
├── your_login_ID/ [Change this directory name to your username]
│   ├── HelloWorld.html [.html and .jsp files go here]
│   └── WEB-INF /
│       ├── classes / [.java files go here]
│       │   └── HelloWorld.java
│       ├── lib / [External .jar libraries]
│       │   ├── mysql-connector-java-5.1.35-bin.jar
│       │   └── ojdbc6_g.jar
│       └── web.xml [Change according to your deployment]
5 directories, 22 files
```

Important:

You will later be using ant to build this template into a .WAR file. This will require that you upload your code to the above directories, configure the web.xml file for your code. You'll also need to remember the absolute path for the sample-deploy directory extracted in this step for later compilation purposes.

Configuring your Java servlet web.xml file

You'll need to modify the "web.xml" file for different jsp & servlet files. web.xml is located at `sample-deploy\your_login_ID\WEB-INF\web.xml`.

Java servlet example, subsection from WEB-INF/web.xml:

```
<servlet>
  <!-- Tomcat internal naming convention -->
  <servlet-name>HelloWorld</servlet-name>
  <!-- servlet-class is a reference to the HelloWorld.java file -->
  <servlet-class>HelloWorld</servlet-class>
  <load-on-startup>3</load-on-startup>
</servlet>
<servlet-mapping>
  <!-- Tomcat internal naming convention -->
  <!-- Must match above servlet-name -->
  <servlet-name>HelloWorld</servlet-name>
  <!-- http://web.cs.siu.edu:8080/your-login-id/HelloWorld -->
  <url-pattern>/HelloWorld</url-pattern>
</servlet-mapping>
```

In the example above `<servlet-class>HelloWorld</servlet-class>` needs to correspond to your `HelloWorld.java` file being deployed.

Both fields `<servlet-name></servlet-name>` must have matching values. i.e., `<servlet-name>HelloWorld</servlet-name>`

You'll also need to define the url-pattern that you would like the java servlet to be accessible from,

i.e., `<url-pattern>/HelloWorld</url-pattern>`

After deploying this example it would be viewable at:
`http://web.cs.siu.edu:8080/your_login_ID/HelloWorld`

Additionally, `http://www.cs.siu.edu/download/tomcat/web.xml` contains a more complete sample web.xml file for reference.

Compiling and Deploying .WAR files with ANT

Open a dos/bash command prompt and cd to your sample-deploy-servlet or sample-deploy-jsp directory.

To compile your servlet/jsp into a .war file, run command:

```
$ ant -buildfile build.xml -Dwebapp=your_login_ID compile
```

i.e., ant -buildfile build.xml -Dwebapp=jdoe compile

When it is finished, it will tell you whether it is successful or not.

To deploy your servlet to the departmental Tomcat server
(<http://web.cs.siu.edu:8080>), run command:

```
$ ant -buildfile build.xml -Dwebapp=your_login_ID deploy
```

i.e., ant -buildfile build.xml -Dwebapp=jdoe deploy

You will see it is running the deployment. When it is finished, you'll be informed on whether it is successful or not.

Run your deployed program in your favorite web browser by typing the following url:

```
http://lab.cs.siu.edu:8080/your_login_ID/jspfilename_or_servlet_name
```

i.e., http://lab.cs.siu.edu:8080/jdoe/jspfilename_or_servlet_name

Examples:

Servlet: **<http://web.cs.siu.edu:8080/mbarkdoll>HelloWorld>**

JSP: **<http://web.cs.siu.edu:8080/mbarkdoll>HelloWorld.jsp>**

Quick Reference

The JSP and Servlet templates work with both windows and linux computers.
(Prerequisites: Ant and Java installed and configured properly.)

JSP Template

Prerequisites: Ant and Java installed and configured

Includes: Tomcat Deployer Client
Ant build file (build.xml)
MySQL/Oracle JDBC Driver

Download:

<http://cs.siu.edu/download/tomcat/sample-deploy-jsp.zip>

Servlet Template

Prerequisites: Ant and Java installed and configured

Includes: Tomcat Deployer Client
Ant build file (build.xml)
MySQL/Oracle JDBC Driver

Download:

<http://cs.siu.edu/download/tomcat/sample-deploy-servlet.zip>