

## ภาคผนวก ข

### Java Web Applications

Struts Application ทั้งหมดจะถูกรวมกันโดยใช้รูปแบบของ Java Web Application ซึ่งต่อไปนี้จะ เป็นรายละเอียดเกี่ยวกับพื้นฐานเบื้องต้นของ Java Web Applications

Java Web applications อธิบายได้โดย Java Servlet Specification 2.2 ซึ่งได้อธิบายแนวคิดของ Java Web application ไว้ว่า Web Application คือ ชุดของ servlet, HTML page, class และทรัพยากรอื่น ๆ ที่ถูกรวมกันไว้และทำการประมวลผลบนหลาย ๆ ตัวบรรจุ (container) จากหลาย ๆ ผู้ค้า ถ้าจะให้คำจำกัดความอย่างง่าย ๆ ก็คือ Java Web application เป็นชุดของ Web component หนึ่งชิ้นส่วนหรือมากกว่าที่ถูกรวมเข้าไว้ด้วยกันเพื่อสร้างโปรแกรมประยุกต์ที่สมบูรณ์หรือการประมวลผลจาก Web layer ของ enterprise application โดยชิ้นส่วนพื้นฐานของ Web application ที่ถูกรวมเข้าไว้ด้วยกันมีดังต่อไปนี้

- Servlets
- JavaServer Pages (JSPs)
- JSP custom tag libraries
- Utility classes และ Application classes
- Static documents, including HTML, images, JavaScript และทรัพยากรอื่น ๆ
- Metainformation describing the Web application

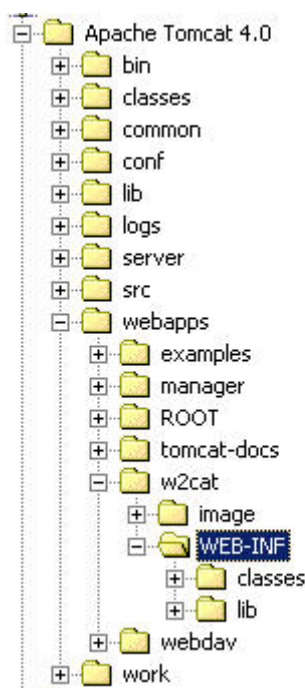
#### ข.1 โครงสร้างของไดเร็คทอรี

Web application ทั้งหมดถูกรวมกันไว้ในโครงสร้างที่เป็นไดเร็คทอรีและ โครงสร้างของไดเร็คทอรีนี้คือ container ซึ่งบรรจุชิ้นส่วนต่าง ๆ ของ Web application โดยขั้นตอนแรกในการสร้าง Web application คือ การสร้างโครงสร้างของไดเร็คทอรีดังกล่าวนี้ ตารางที่ ข.2 อธิบายถึงตัวอย่างของ Web application ที่ชื่อ w2cat และเนื้อหาที่บรรจุอยู่ในแต่ละไดเร็คทอรีนั้น ๆ โดยแต่ละไดเร็คทอรีสร้างจาก <SERVER\_ROOT> ของ Servlet/JSP container

DIRECTORY	CONTAINS
/w2cat	ไดเรกทอรีรากของ Web application ส่วนของ JSP และ HTML file จะถูกจัดเก็บไว้ที่นี้ทั้งหมด
/w2cat/WEB-INF	เป็นไดเรกทอรีที่บรรจุทรัพยากรทั้งหมดที่เกี่ยวข้องกับแอปพลิเคชัน ที่มีไม่มีใน document root ของแอปพลิเคชัน และเป็นไดเรกทอรีที่ใช้เก็บ Web application deployment descriptor จะสังเกตได้ว่าไดเรกทอรี WEB-INF ไม่ได้เป็นส่วนหนึ่งของ public document ไม่มีแฟ้มข้อมูลใดจาก client ถูกบันทึกอยู่ในไดเรกทอรีนี้
/w2cat/WEB-INF/classes	เป็นไดเรกทอรี ที่ใช้บรรจุ servlet และ utility classes
/w2cat/WEB-INF/lib	เป็นไดเรกทอรีที่บรรจุแฟ้มข้อมูล Java Archive (JAR) ที่เกี่ยวข้องกับ Web application

ตารางที่ ข.1 ตารางแสดงโครงสร้างของไดเรกทอรี Web Application

ถ้าใช้ Tomcat เป็นตัว container จะมีการกำหนดไดเรกทอรีรากเป็น <CATALINA\_HOME>/webapps/ ภาพประกอบ ข.1 แสดง w2cat ที่ใช้ Tomcat เป็น container



ภาพประกอบ ข.1 แสดง w2cat Web application ที่มี Tomcat เป็น container

## ข.2 Web Application Deployment Descriptor

ส่วนที่สำคัญที่สุดของทุก ๆ Web application คือ deployment descriptor ของมัน Web application deployment description คือ XML file ที่ชื่อ web.xml ที่บรรจุอยู่ในไดเรกทอรี `/<SERVER_ROOT>/applicationname/WEB-INF/` เพิ่มข้อมูล web.xml จะเป็นตัวอธิบายถึงชิ้นส่วนต่าง ๆ ใน Web application ตัวอย่างเช่น ถ้าเราใช้แอปพลิเคชันชื่อ wiley ดังนั้น web.xml จะถูกเก็บไว้ที่ `/<SERVER_ROOT>/wileyapp/WEB-INF/` ค่าต่าง ๆ ที่อธิบายใน deployment descriptor มี element ต่าง ๆ ดังต่อไปนี้

- ServletContext init parameters
- Localized content
- Session configuration
- Servlet/JSP definitions
- Servlet/JSP mappings
- Tag library references
- MIME type mappings

- Welcome file list
- Error pages
- Security information

ตัวอย่างต่อไปนี้เป็นารแสดง code snippet ที่บรรจุตัวอย่างของ deployment descriptor ที่กำหนดไว้ใน single 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'>

<servlet>
    <servlet-name>SimpleServlet</servlet-name>
    <servlet-class>com.wiley.SimpleServlet</servlet-class>
</servlet>

</web-app>
```

### ข.3 Packaging a Web Application

รูปแบบมาตรฐานของการทำ packaging สำหรับ Web application คือ Web Archive file (WAR) WAR file ก็คือ JAR file ที่มีนามสกุลเป็น .war แทนที่จะเป็น .jar เราสามารถสร้าง WAR file โดยใช้คำสั่ง jar ซึ่งเป็น Java's archiving tool ขั้นตอนอย่างง่ายในการสร้าง WAR file คือการใช้คำสั่ง

```
jar cvf wileyapp.war
```

ผลของคำสั่งนี้จะเป็นการสร้าง archive file ที่ชื่อ wileyapp.war ที่บรรจุชิ้นส่วนต่าง ๆ ของ wileyapp Web application ซึ่งในตอนนี้อาจสามารถนำ Web application ออกไปใช้งานได้โดยการใส่ file ที่สร้างขึ้นนี้

#### ข.4 Tomcat JSP/Servlet Container

Tomcat Server เป็น open-source Java-based Web application container ที่สร้างขึ้นเพื่อประมวลผล servlet และ JavaServer Page Web application

ในการติดตั้ง Tomcat เป็น web server นั้นต้องอาศัย component ต่าง ๆ ดังตารางที่ ข.2

COMPONENT	LOCATION
Jakarta-Tomcat 4	<a href="http://jakarta.apache.org">http://jakarta.apache.org</a>
JDK 1.3 Standard Edition	<a href="http://java.sun.com/j2se/1.3/">http://java.sun.com/j2se/1.3/</a>

ตารางที่ ข.2 ตารางแสดงรายการที่ใช้ในการติดตั้ง Tomcat

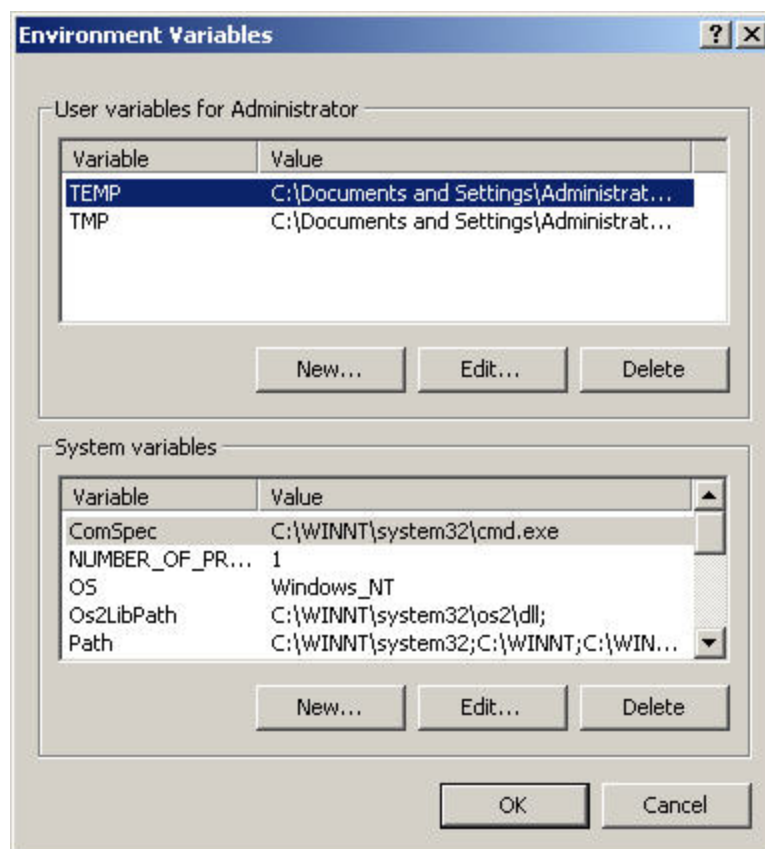
#### การติดตั้ง Tomcat

Tomcat สามารถติดตั้งเป็น stand-alone server บนระบบปฏิบัติการ Windows NT/2000 โดยการติดตั้ง JDK ซึ่ง directory ที่ทำการติดตั้ง JDK จะถูกกำหนดให้เป็น JAVA\_HOME directory เช่น ถ้าทำการติดตั้ง JDK ที่ drive D จะได้ D:\jdk1.3 เป็น JAVA\_HOME

ขั้นตอนต่อมาคือการติดตั้ง Tomcat Server ซึ่งถ้ามีการดาวน์โหลด Tomcat มาจะไม่พบโปรแกรมที่ทำหน้าที่ติดตั้ง Tomcat แต่จะทำการสร้าง TOMCAT\_HOME directory จาก path ที่มีการอ้างถึงไดเรกทอรีของ Tomcat เช่น D:\jakarta-tomcat-4.0.x. จะใช้ path นี้เป็น TOMCAT\_HOME

หลังจากติดตั้ง Tomcat แล้ว ขั้นตอนต่อมาคือการกำหนดค่าของ environment variable ต่าง ๆ ให้แก่ JAVA\_HOME และ TOMCAT\_HOME ซึ่งตัวแปรเหล่านี้จะนำมาใช้ในการแปล JSP และประมวลผล Tomcat ซึ่งถ้าใช้ระบบปฏิบัติการ Window NT/2000 ต้องทำตามขั้นตอนดังต่อไปนี้

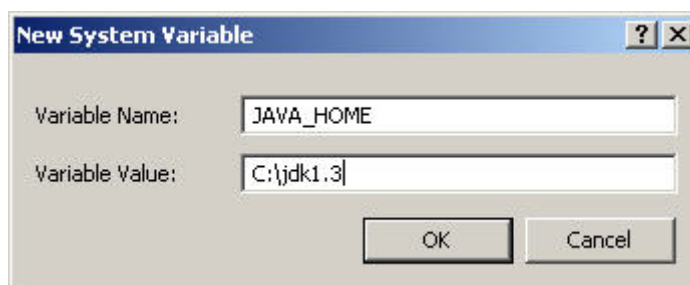
1. เปิด NT/2000 Control Panel
2. เปิด NT/2000 System Application แล้วเลือก Advanced Tab
3. คลิกที่ปุ่ม Environment Variables จะเห็นหน้าจอตั้งภาพประกอบ ข.2



ภาพประกอบ ข.2 แสดงหน้าจอ Windows NT/2000 Environment Variables dialog box

4. คลิกที่ปุ่ม New ใน System Variables section ของ Environment Variable dialog box จากนั้นเพิ่มตัวแปร JAVA\_HOME และตั้งค่าให้เป็น path ที่ทำการติดตั้ง JDK แสดงตัวอย่างการตั้งค่า ดังภาพประกอบ ข.3

5. ขั้นตอนสุดท้าย คือการทำซ้ำในขั้นตอนที่ 4 แต่เปลี่ยนเป็นการตั้งค่าให้กับ CATALINA\_HOME สำหรับค่าที่กำหนดให้คือไดเรกทอรีที่ทำการติดตั้ง Tomcat ตัวอย่างเช่น D:\jakarta-tomcat-4.0.1 เป็นต้น



ภาพประกอบ ข.3 แสดงการตั้งค่า JAVA\_HOME environment

## การทดสอบการติดตั้ง Tomcat

ขั้นตอนในการทดสอบว่า Tomcat ของเรามีการติดตั้งที่สมบูรณ์หรือไม่ เริ่มจากเรียกใช้งาน Tomcat Server โดยการใช้คำสั่งต่อไปนี้ ( แทนที่ <CATALINA\_HOME> ด้วยค่าของ location ที่มีการติดตั้ง Tomcat)

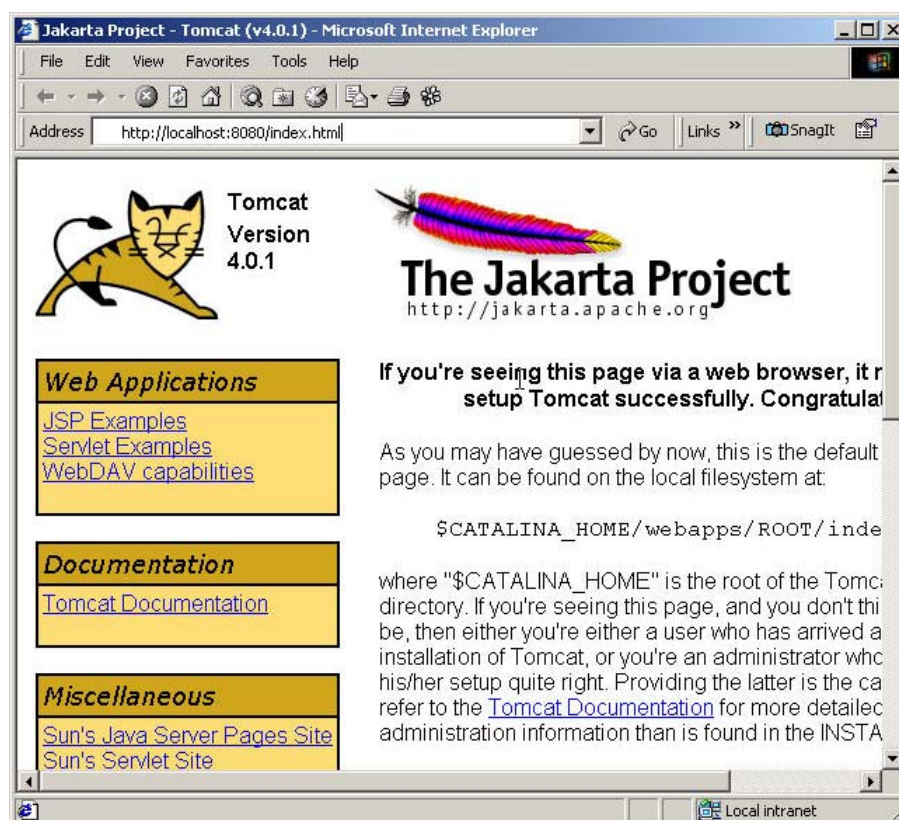
```
<CATALINA_HOME>\bin\startup.bat
```

เมื่อเรียกใช้งาน Tomcat แล้ว ให้เปิด browser ไปที่ URL :

```
http://localhost:8080
```

ถ้ามีการทำงานที่ถูกต้อง จะปรากฏหน้าจอ ดังภาพประกอบ ข.4

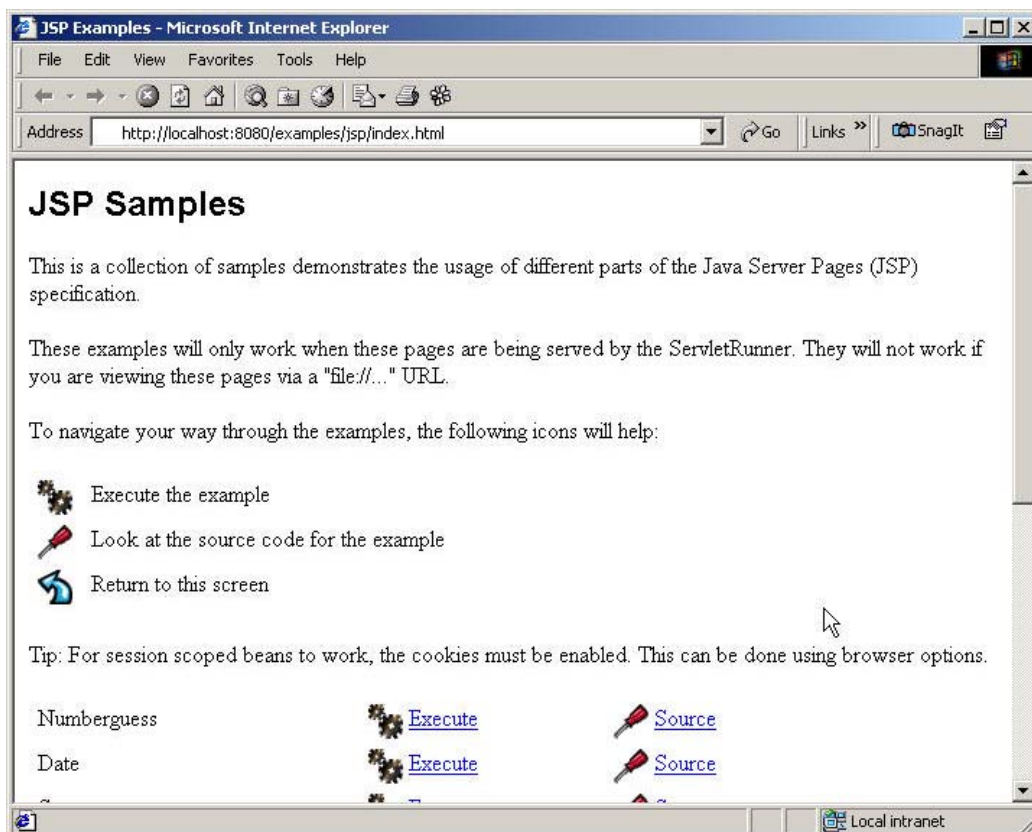
ขั้นตอนต่อไปคือการตรวจสอบการติดตั้ง JDK วิธีที่ดีที่สุดในการตรวจสอบคือการ เลือกประมวลผลตัวอย่าง JSP ที่ Tomcat Server จัดเตรียมไว้ให้ การประมวลผลตัวอย่าง JSP สามารถเลือกจาก Tomcat homepage ที่มีการกำหนดไว้ให้ ในส่วนของ JSP Examples ดังแสดงในภาพประกอบ ข.4 เมื่อคลิกแล้วจะปรากฏหน้าจอ ดังภาพประกอบ ข.5



ภาพประกอบ ข.4 แสดงหน้าจอหลักของ Tomcat Home page

จากนั้นทำการเลือกตัวอย่าง JSP ที่ต้องการ แล้วเลือก Execute link ถ้าการติดตั้งของเรามีความสมบูรณ์จะสามารถประมวลผลตัวอย่างได้

ถ้าไม่สามารถแสดงผลหน้าจอในภาพประกอบ ข.5 ได้ ให้ทำการตรวจสอบตำแหน่งของ JAVA\_HOME environment variable ว่าตรงกับตำแหน่งที่ติดตั้ง JDK หรือไม่



ภาพประกอบ ข.5 แสดง JSP Examples page