The **property** tag is used to get the property of a value, which will default to the top of the stack if none is specified. This example shows you the usage of three simple data tags - namely **set**, **push** and **property**.

Create action classes

For this exercise, let us reuse examples given in "Data Type Conversion" chapter but with little modifications. So let us start with creating classes. Consider the following POJO class **Environment.java**.

```java
package com.tutorialspoint.struts2;

public class Environment {
   private String name;
   public  Environment(String name)
   {
      this.name = name;
   }
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }
}
```

Let us have following action class:

```java
package com.tutorialspoint.struts2;
import com.opensymphony.xwork2.ActionSupport;

public class SystemDetails extends ActionSupport {
   private Environment environment = new Environment("Development");
   private String operatingSystem = "Windows XP SP3";

   public String execute()
   {
      return SUCCESS;
   }
   public Environment getEnvironment() {
      return environment;
   }
   public void setEnvironment(Environment environment) {
      this.environment = environment;
   }
   public String getOperatingSystem() {
      return operatingSystem;
   }
   public void setOperatingSystem(String operatingSystem) {
      this.operatingSystem = operatingSystem;
   }
}
```

Create views

Let us have **System.jsp** with the following content:

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
 pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>System Details</title>
</head>
<body>

   <p>The environment name property can be accessed in three ways:</p>

   (Method 1) Environment Name:
   <s:property value="environment.name"/><br/>

   (Method 2) Environment Name:
   <s:push value="environment">
      <s:property value="name"/><br/>
   </s:push>

   (Method 3) Environment Name:
   <s:set name="myenv" value="environment.name"/>
   <s:property value="myenv"/>

</body>
</html>
```

Let us now go through the three options one by one:

- In the first method, we use the property tag to get the value of the environment's name. Since the environment variable is in the action class, it is automatically available in the value stack. We can directly refer to it using the property **environment.name**. Method 1 works fine, when you have limited number of properties in a class. Imagine if you have 20 properties in the Environment class. Every time you need to refer to these variables you need to add "environment." as the prefix. This is where the push tag comes in handly.

- In the second method, we push the "environment" property to the stack. Therefore now within the body of the push tag, the environment property is available at the root of the stack. So you now refer to the property quite easily as shown in the example.

- In the final method, we use the set tag to create a new variable called myenv. This variable's value is set to environment.name. So, now we can use this variable wherever we refer to the environment's name.

Configuration Files

Your **struts.xml** should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
   <constant name="struts.devMode" value="true" />
   <package name="helloworld" extends="struts-default">
      <action name="system"

            method="execute">
         <result name="success">/System.jsp</result>
      </action>
   </package>
</struts>
```

Your **web.xml** should look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://java.sun.com/xml/ns/javaee"
   xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
```

```xml
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
    >

    <display-name>Struts 2</display-name>
    <welcome-file-list>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
    <filter>
        <filter-name>struts2</filter-name>
        <filter-class>
            org.apache.struts2.dispatcher.FilterDispatcher
        </filter-class>
    </filter>

    <filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
 </web-app>
```

Right click on the project name and click **Export > WAR File** to create a War file. Then deploy this WAR in the Tomcat's webapps directory. Finally, start Tomcat server and try to access URL http://localhost:8080/HelloWorldStruts2/system.action. This will give you following screen: