Struts

Student Workbook

itcourseware
Training Materials For IT Professionals

STRUTS

Suzanne Werle

Published by ITCourseware, LLC., 7245 South Havana Street, Suite 100, Centennial, CO 80112

Contributing Author: Jamie Romero

Editor: Rick Sussenbach

Special thanks to: Many instructors whose ideas and careful review have contributed to the quality of this workbook, including Brandon Caldwell, John Crabtree, Julie Johnson, and Mike Naseef, and the many students who have offered comments, suggestions, criticisms, and insights.

CONTENTS

CHAPTER 1 - COURSE INTRODUCTION

COURSE OBJECTIVES

- Describe how Struts fit in a Java application server environment.
- Use Struts to implement an MVC web application design.
- Configure a Struts application using *struts-config.xml* and *web.xml*.
- Build a JSP view using Struts and JSTL tags.
- Handle form validation, error processing, and logging in a Struts environment.
- Use the Struts Tiles facilities to make the look and feel of a web application flexible and easy to maintain.

COURSE OVERVIEW

- **Audience**: Experienced Java Servlet and JSP developers who need to use Struts as a framework for MVC web application development.

- **Prerequisites**: *Java Programming* and *Java Web Programming*.

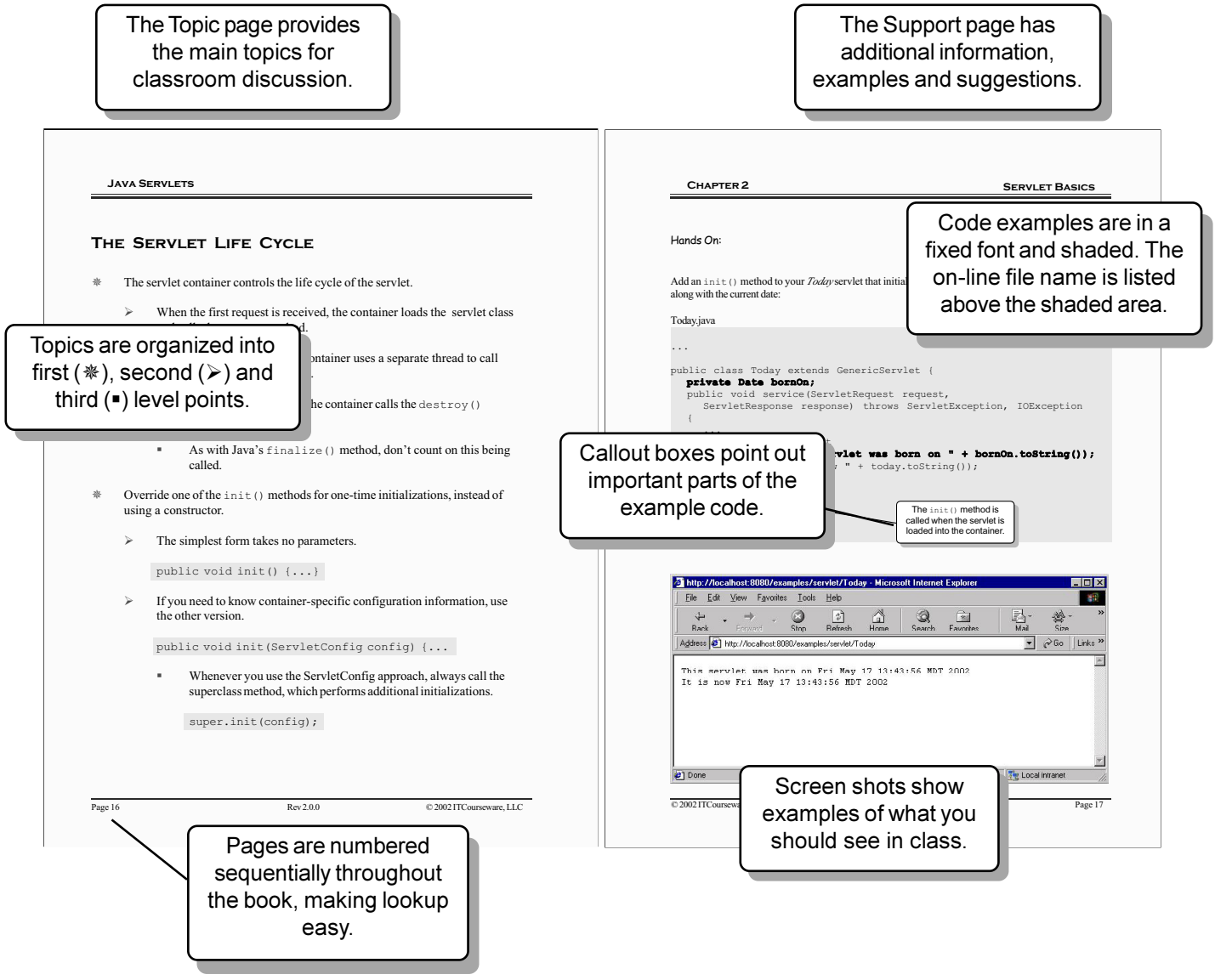
- **Classroom Environment:**

  - A workstation per student.

USING THE WORKBOOK

This workbook design is based on a page-pair, consisting of a Topic page and a Support page. When you lay the workbook open flat, the Topic page is on the left and the Support page is on the right. The Topic page contains the points to be discussed in class. The Support page has code examples, diagrams, screen shots and additional information. **Hands On** sections provide opportunities for practical application of key concepts. **Try It** and **Investigate** sections help direct individual discovery.

In addition, there is an index for quick look-up. Printed lab solutions are in the back of the book as well as on-line if you need a little help.



Suggested References

Carnell, John and Rob Harrop. 2004. ***Pro Jakarta Struts, Second Edition***. APress L.P., Berkeley, CA. ISBN 159059228X

Cavaness, Chuck and Brian Keeton. 2003. ***Jakarta Struts Pocket Reference***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596005199

Cavaness, Chuck. 2004. ***Programming Jakarta Struts***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596006519

Franciscus, George and Danilo Gurovich. 2004. ***Struts Recipes***. Manning Publications, Greenwich, CT. ISBN 1932394249

Friedl, Jeffrey. 2002. ***Mastering Regular Expressions, Second Edition***. O'Reilly & Associates, Sebastopol, CA. ISBN 0596002890

Goodwill, James and Richard Hightower. 2003. ***Professional Jakarta Struts***. Wrox Press/Wiley Publishing, Inc., Indianapolis, IN. ISBN 0764544373

Holmes, James and Herbert Schildt. 2004. ***Struts: The Complete Reference***. Osborne/McGraw-Hill, Emeryville, CA. ISBN 0072231319

Husted, Ted, Cedric Dumoulin, George Franciscus, and David Winterfeldt. 2002. ***Struts in Action: Building Web Applications with the Leading Java Framework***. Manning Publications, Greenwich, CT. ISBN 1930110502

Kurniawan, Budi. 2005. ***Struts Design and Programming: A Tutorial***. BrainySoftware.com, Vancouver, BC, Canada. ISBN 0975212818

Shenoy, Srikanth and Nithin Mallya. 2004. ***Struts Survival Guide: Basics to Best Practices***. Objectsource LLC, Austin, TX. ISBN 0974848808

Siggelkow, Bill. 2005. ***Jakarta Struts Cookbook***. O'Reilly & Associates, Sebastopol, CA. ISBN 059600771X

Spielman, Sue. 2002. ***The Struts Framework: Practical Guide for Java Programmers***. Morgan Kaufman Publishers, San Francisco, CA. ISBN 1558608621

http://struts.apache.org
http://java.sun.com

CHAPTER 2 - STRUTS OVERVIEW

OBJECTIVES

- Describe how Struts fits into a Java web application.
- Identify the major components of the MVC Model 2 application design.
- Outline the basic flow through a web application using Struts.

What is Struts?

- Struts is an open source Java web application framework based on the Model-View-Controller design pattern.
  - It is based upon proven design patterns and published Java-based standards.
  - The name is a reference to the concept of struts in the architectural sense — struts form the skeleton of a building or bridge.
- Struts provides a solid basis for a framework that is easily extensible.
  - Struts framework classes can be easily extended to add or modify functionality.
- Struts facilitates change in web applications and specialization of development team members.
- Struts is an Apache Jakarta project sponsored by the Apache Software Foundation.
  - For more information, see: ***http://struts.apache.org***
- Struts uses Jakarta Commons projects, see: ***http://jakarta.apache.org/commons/***.
  - BeanUtils — JavaBean utility classes
  - Validator — Java and JavaScript form validation
  - Digester — An XML parser that facilitates the conversion of XML to objects
  - Collections — Some simple, fast collection utility classes
  - Logging — An interface to several prominent logging technologies

The Apache Struts project has been broken into two frameworks: Stuts Action Framework and Struts Shale Framework.

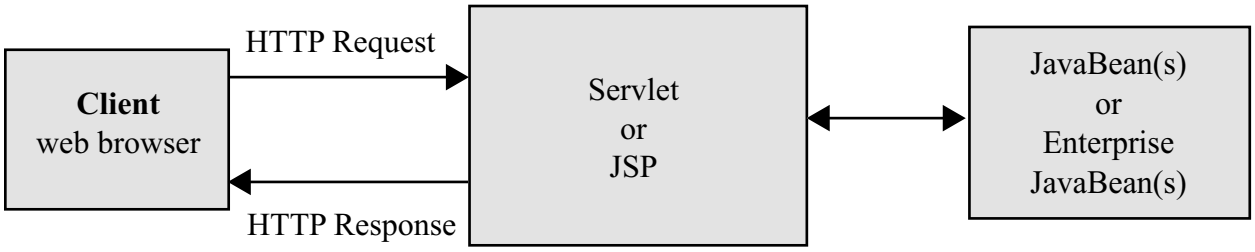
The Struts Action Framework is the new name for what has traditionally been called Struts. It continues support and development on the request-based JSP framework that has been used in production since 2001. In this course, we will be studying the Struts Action Framework.

The Struts Shale Framework is a new project that builds on five years of learning and technology advances. It embraces the JavaServer Faces (JSF) component model and allows for integration with other popular frameworks, such as Spring.

MODEL 1 DESIGN PATTERN

- Servlet and JSP development typically involves several technologies and skills:
  - Java
  - JSP
  - HTML, XHTML, XML, XSLT
  - CSS
  - JavaScript
- Co-mingling Java and JavaScript can make pages difficult to follow.
- Coupling business logic and presentation logic can also cause problems.
  - It is difficult to modify the website appearance.
  - Changes in business rules can necessitate retesting of the web application.
  - Embedded flow of control can be difficult to follow.
  - A web application is a more complex debugging environment.
- Servlets and JSPs make it easy to build simple inquiry sites.

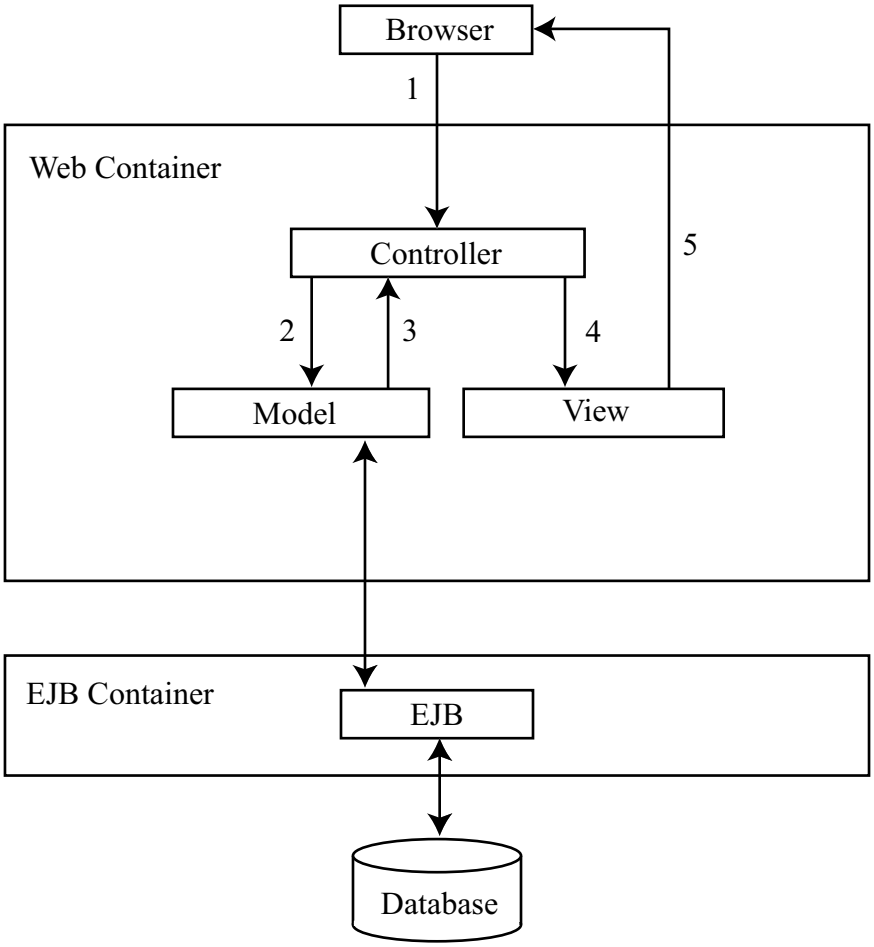
MVC Model 1


HTTP Request
Client
web browser
Servlet
or
JSP
JavaBean(s)
or
Enterprise
JavaBean(s)
HTTP Response

MODEL 2 / MVC DESIGN PATTERN

- The *model*:
  - Encapsulates the function of an application, including its internal state and actions to change its state.
  - Typically knows nothing about the data presentation (view) or the application flow (control).
  - Can use many Java-based technologies, such as JavaBeans, Enterprise JavaBeans, JDBC, etc.
  - Components are not necessarily dependent upon a web container environment.
- The *view*:
  - Is a presentation of the model.
  - Retrieves data with the accessor methods of a model, but knows nothing about the controller.
  - Should be notified when the model changes (state change).
- The *controller*:
  - Reacts to user input and delegates the request.
  - Creates and initializes the model with user information.
  - Mediates flow, determining the flow required to handle a request.
  - Is frequently directed by external configuration files, such as XML files.

A typical J2EE MVC application flow:



1. The browser makes an HTTP request. The web container forwards the request to the controller assigned to this kind of request.

2. The controller (usually a servlet), invokes business tier logic via the model. This may involve EJB and/or JDBC.

3. New or updated information is returned to the controller via the model.

4. The controller passes the new information to the appropriate view (usually a JSP).

5. The new view is returned to the browser via an HTTP response.

IMPLEMENTING MVC WITH A FRAMEWORK

- Frameworks such as Struts provide a generic implementation of the MVC design pattern so that the developer can focus on the application requirements, rather than the MVC architecture.

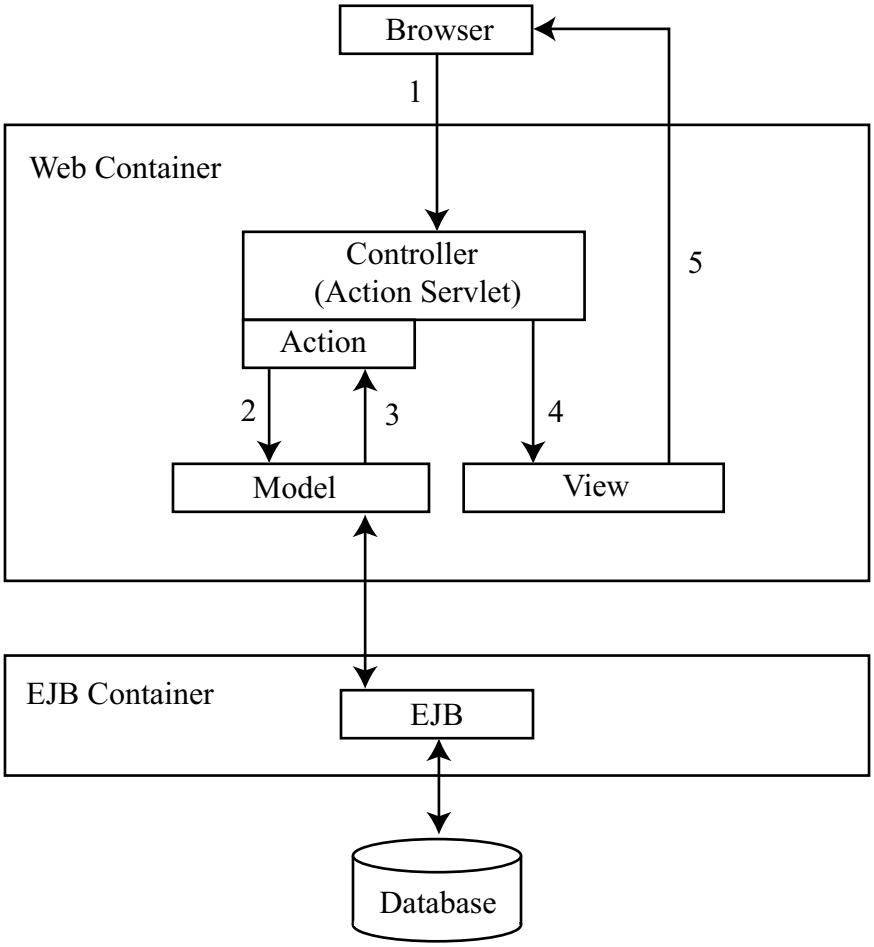
- Some functions of the framework:

  - To be declarative (XML) rather than code (Java) driven.
  - Map the incoming request to the application processing components.
  - Handle form input.
  - Provide error handling support.
  - Provide forwarding support, particularly for the view.
  - Provide for internationalization of views.
  - Assist in the development of a JSP.
  - Provide for user extensions.

THE STRUTS FRAMEWORK

- Struts is a group of configurable components designed for use in a Model 2 / MVC application design.
  - All parts of the Struts framework can be extended.
- The Struts controller is configurable.
  - The main controller is a Java servlet called **ActionServlet**.
  - It uses other controller component classes, including **RequestProcessor**, **ActionForm**, **ActionMapping**, **Action**, **ActionForward**, and **ActionError**.
- The Struts model interfaces with a user model.
  - There are no specific model components to Struts.
  - Struts can use existing Java-based technologies, such as JavaBeans, EJB, and JDBC.
  - There are typically beans that execute business logic and other beans that persist data.
- The Struts view is based on JSP technology.
  - View components are typically JSPs.
    - Struts includes extensive tag libraries and Velocity templates.
  - Struts can incorporate other web technologies, such as static template text (HTML, XML), dynamic content from other servlets and JSPs, and XSLT.

A typical J2EE Struts application flow:



1. The browser makes an HTTP request. The web container forwards the request to the Struts controller (**ActionServlet**). The **ActionServlet** passes the request to the appropriate **Action** that has been coded to deal with this request.

2. The **Action** invokes the business tier logic with the form bean (**ActionForm**). This may involve EJB and/or JDBC.

3. New or updated information is returned to the controller via the model (form bean).

4. The controller passes the new information (form beans and other JavaBeans) to the appropriate view (usually a JSP).

5. The new view is returned to the browser via an HTTP response.

Basic Struts Components

- An **ActionMapping** is an encapsulation of how an HTTP request path is mapped to a specific **Action**.
- An **ActionServlet** is an implementation of the command design pattern.
  - The user agent creates an HTTP event.
  - The web container creates an HTTP request and follows the mapping to the **ActionServlet**.
  - All of the processing for a web context goes through one **ActionServlet**.
- **ActionForm** beans, often called "form beans," mediate between the model and the view.
  - They are created and set by the controller.
  - They can contain validation.
  - They represent the request and response state, not the persistent business data.
- The controller passes the request, response, mapping information, and the form bean to the **execute()** method of your **Action** subclass.
  - Your **execute()** method should implement the business logic as a thin wrapper, or better still, act as a delegate to business beans.
  - Your implementation may act as an adapter that converts the form data and request into data that the business components expect.

BASIC STRUTS COMPONENTS (CONT'D)

- An **ActionForward** maps a logical name to a destination.
  - The **Action** determines what view or subsequent **Action** should get control next by requesting a symbolically-referenced **ActionForward**.
  - **ActionForward**s are configured in *struts-config.xml* and, thus, loosely coupled to the application.
- **ActionErrors** and **ActionMessages** contain lists of messages that can be passed to a view to be displayed with Struts tags.
- *struts-config.xml* contains configuration information for Struts, form beans, actions, mappings, forwarding, and exception handling.
- *web.xml* configures the **ActionServlet** and its servlet mapping, and locates the *struts-config.xml* file.
- Struts tag libraries assist in the preparation of a JSP view.

STRUTS DOCUMENTATION

- The struts documentation can be found online at ***http://struts.apache.org/struts-action/index.html***.

- The User Guide contains the basic specifications for Struts.

  - General overview information on each of the components, and how to build the *struts-config.xml* file, can be found at ***userGuide/index.html***.

  - More detailed information is available in the JavaDoc API documentation.

- Struts FAQs and How to Guides contain useful links, samples, and answers to basic questions.

- There is separate documentation for the tag libraries and utilities.

  - It can be found at ***http://struts.apache.org/struts-taglib/index.html***.

- Documentation for the Apache Foundation Commons Components, such as Logging and Validator, can be found at ***http://jakarta.apache.org/commons/index.html***.

- You'll probably want documentation for Java and the web container, as well.


Apache Struts - Struts Action Framework - Welcome - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://struts.apache.org/struts-action/index.html
Search
Home Bookmarks mail craigslist Freecycle CNN Denver Channel ITC Home BHI OPS Littleton High
Apache Struts - Struts Action Frame...
The Apache Software Foundation
http://www.apache.org/
Struts
Last published: 11 March 2006 | Doc for 1.3.1
Struts Action Framework
Welcome
Learning
Roadmap
Milestones
Download
Quick Links
User Guide
FAQs and HOWTOs
JavaDocs
Release Notes
Source Repository
Apache Struts Home
Project Documentation
About Struts Action Framework
Project Info
Project Reports
Development Process
built by maven
Welcome to Struts Action Framework
Struts Action Framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, ResourceBundles, and XML, as well as various Jakarta Commons packages, like BeanUtils and Chain of Responsibility. Struts Action Framework helps you create an extensible development environment for your application, based on published standards and proven design patterns.
Action Framework in a Nutshell
The framework provides its own web Controller component and integrates with other technologies to provide the Model and the View. For the Model , the framework can interact with standard data access technologies, like JDBC and EJB , as well as most any third-party packages, like Hibernate , iBATIS , or Object Relational Bridge . For the View , the framework works well with JavaServer Pages , including JSTL and JSF , as well as Velocity Templates , XSLT , and other presentation systems.
The framework's Controller acts as a bridge between the application's Model and the web View. When a request is received, the Controller invokes an Action class. The Action class consults with the Model (or, preferably, a Facade representing your Model) to examine or update the application's state. The framework provides an ActionForm class to help transfer data between Model and View.
Most often, the Model is represented as a set of JavaBeans . Typically, developers will use the Commons BeanUtils to transfer data between ActionForms and the Model objects (or a Facade). Preferably, the Model will do the "heavy lifting", and the Action will act as a "traffic cop" or adapter.
Struts Config in a Nutshell
A web application uses a deployment descriptor to initialize resources like servlets and taglibs . The deployment descriptor is formatted as a XML document and named "web.xml". Likewise, the framework uses a configuration file to initialize its own resources.

A STRUTS-BASED APPLICATION: LOGON

- The client initiates a logon request.
- The web container creates an **HttpServletRequest** object and passes it to *logon.jsp*.
- The **HttpServletResponse** of the JSP returns a logon form.
- The client completes the logon form (user and password parameters) and submits it to **/logon.do**.
- The web container creates an **HttpServletRequest** object and follows a servlet mapping specified in *web.xml* to the **ActionServlet** (the Struts controller).
- The controller creates an **ActionMapping** indicating that the request path was "**/logon**" and that control should go to the **LogonAction**, with its parameters saved in an instance of **LogonForm**.
  - This is often referred to as a *switchboard action*.
- The **LogonAction execute()** method validates the logon.
  - If valid, control goes to a welcome page (*welcome.jsp*) logically specified in *struts-config.xml* as "success" after saving the user in the **HttpSession** object.
  - If invalid, an error message is added and control returns to the input page (*logon.jsp*) to be displayed on the logon page.
- The valid (*welcome.jsp*) or invalid (*logon.jsp*) JSP is invoked by the web container and the response it generates is returned to the client.

Struts comes as a ZIP file that includes the documentation as a WAR file. You can access it through a web server or view it offline with a browser. To compile a Java component that uses Struts, several JAR files are needed in the classpath, including *struts.jar* and your application server's JAR file that contains the Servlet API. The Apache Commons packages have JARs that may be needed, as well.

To use Struts in a web application, *struts.jar* and other Common's JARs, such as *commons-logging.jar*, *commons-beanutils.jar*, and *commons-digester.jar*, from the ***<strutsInstall>\lib*** must be available to your application.

To simplify the building and deployment process, we will use the Apache Software Foundation's Ant tool. To use Ant, you create an XML configuration file called *build.xml* and define targets that define the steps for building and deploying your application. The targets also set dependencies, so that you can compile, build, and deploy with a single command.

For this class, we have provided *build.xml* files for you. They contain the proper classpath settings and targets for compiling the Java files, building the WAR file, and deploying the WAR file.

Try It:

To test the Logon application that is described on the previous page, perform the following steps:

1. Start your application server or web container.
2. Run the Ant command in the chapter directory.
3. Open a browser and go to ***http://localhost:*port*/Logon*** to retrieve a logon form.
4. Logon with **student** and **password**.
5. Try again with **student** and any invalid password.
6. Try again with your own name and password.

Your instructor will give you the details of how to start the server, run Ant, and which port to use within the URL.

LABS

❶ Use the Struts API documentation to answer the following questions:

a. What package contains **ActionServlet**, **ActionForm**, and **Action**?
b. What is the return type of the **Action execute()** method?
c. What is the signature of the **ActionForm validate()** method?
d. Where is information regarding the Struts Validator?
e. Does the Struts HTML tag library contain a **<hidden>** tag?

(Solution: *lab1.txt*)

❷ Where is the *struts-config.xml* file located for the Logon application?
(Solution: *lab2.txt*)

❸ Look through the *struts-config.xml* file and see if you can determine where the text for the error messages used in the views of the logon application can be found.
(Solution: *lab3.txt*)

❹ Start a browser and enter ***http://localhost:*port*/Logon/***. Your instructor will tell you the correct port to use for your application server. How does it get to *logon.jsp*?
Hint: Look at *web.xml*.
(Solution: *lab4.txt*)

