Expert One-on-One
J2EE™ Design and Development

Rod Johnson
# Table of Contents

## Introduction

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>J2EE Myths</td>
<td>2</td>
</tr>
<tr>
<td>How is this Book Different?</td>
<td>5</td>
</tr>
<tr>
<td>My Approach</td>
<td>6</td>
</tr>
<tr>
<td>Who this Book Is for</td>
<td>7</td>
</tr>
<tr>
<td>Alims of this Book</td>
<td>7</td>
</tr>
<tr>
<td>What this Book Covers</td>
<td>7</td>
</tr>
<tr>
<td>Assumed Knowledge</td>
<td>8</td>
</tr>
<tr>
<td>Recommended Reading</td>
<td>9</td>
</tr>
<tr>
<td>What You Need to Use this Book</td>
<td>9</td>
</tr>
</tbody>
</table>

## Chapter 1: J2EE Architectures

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals of an Enterprise Architecture</td>
<td>16</td>
</tr>
<tr>
<td>Deciding Whether to Use a Distributed Architecture</td>
<td>18</td>
</tr>
<tr>
<td>New Considerations in J2EE Design</td>
<td>19</td>
</tr>
<tr>
<td>When to Use EJB</td>
<td>20</td>
</tr>
<tr>
<td>Implications of Using EJB</td>
<td>20</td>
</tr>
<tr>
<td>Questionable Arguments for Using EJB</td>
<td>22</td>
</tr>
<tr>
<td>Compelling Arguments for Using EJB</td>
<td>22</td>
</tr>
<tr>
<td>Arguments for Using EJB to Consider on a Case-by-Case Basis</td>
<td>23</td>
</tr>
<tr>
<td>Accessing Data</td>
<td>24</td>
</tr>
<tr>
<td>J2EE Data Access Shibboleths</td>
<td>24</td>
</tr>
<tr>
<td>Entity Beans</td>
<td>25</td>
</tr>
<tr>
<td>Java Data Objects (JDO)</td>
<td>25</td>
</tr>
<tr>
<td>Other O/R Mapping Solutions</td>
<td>25</td>
</tr>
<tr>
<td>JDBC</td>
<td>26</td>
</tr>
<tr>
<td>State Management</td>
<td>26</td>
</tr>
<tr>
<td>J2EE Architectures</td>
<td>26</td>
</tr>
<tr>
<td>Common Concepts</td>
<td>27</td>
</tr>
<tr>
<td>Architectural Tiers in J2EE Applications</td>
<td>27</td>
</tr>
<tr>
<td>The Importance of Business Interfaces</td>
<td>28</td>
</tr>
</tbody>
</table>
Table of Contents

Non-distributed Architectures 28
  Web Application with Business Component Interfaces 28
  Web Application that Accesses Local EJBs 30
Distributed Architectures 32
  Distributed Application with Remote EJBs 32
  Web Application Exposing Web Services Interface 34

Web Tier Design 36
  The Model View Controller (MVC) Architectural Pattern 36
  Connectivity Between the Web Tier and Business Objects 38

Designing Applications for Portability 38

Summary 40

Chapter 2 J2EE Projects: Choices and Risks 43

Developing a Policy on Specification Versions 44

Choosing an Application Server 45
  When to Choose an Application Server 46
  Defining the Requirements 47
  Evaluation Criteria 48
    Supported Specifications 48
    Sun Resources 48
    Cost 50
    Vendor Relationship 50
    Vendor Viability 51
    Development and Deployment 51
    Value-added Features 52
    Quality of Documentation 52
    Availability of Skills 53
    User Experience 53
  Choice Process 54
  Top Mistakes in Choosing an Application Server 54

The "Neat Technology" Trap 55

When to Use Alternative Technologies to Supplement J2EE 56

Portability Issues 56
  What does Portability Mean? 57
  A Pragmatic Approach to Portability 58

Staging Environments and Release Management 58

Building a Team 60
  Team Structure 60
    Who Owns the Architecture? 60
    Vertical or Horizontal Responsibility 61
# Table of Contents

## Choosing Development Tools
- Visual Modeling Tools  63
- IDEs  64
- Build Utilities  64
- Code Generators  66
- Version Control  67

## Identifying and Mitigating Risks

## Summary

## Chapter 3: Testing J2EE Applications

### What Can Testing Achieve?

### Definitions

### Testing Correctness
- The XP Approach to Testing  76
- Writing Test Cases  78
  - What Makes a Good Test Case?
  - Recognizing Test Case Authoring and Maintenance as a Core Task  78
- Unit Testing  78
  - main() Methods  79
  - Using JUnit  79
  - Test Practices  86
- Should Testing Strategy Affect How We Write Code?
- Integration and Acceptance Testing
- Testing Business Objects  92
  - Testing Business Objects Implemented Without Using EJB  92
  - Testing EJBs  92
  - Testing Database Interaction  99
- Testing Web Interfaces  100
  - Unit Testing Web-Tier Components  101
  - Acceptance Testing Web Interfaces  104
- Design Implications

### Testing Performance and Scalability
- Load Testing EJBs and Other Business Objects  105
- Load Testing Web Interfaces  105

### Automating Tests

### Complementary Approaches to Testing

### Summary
# Table of Contents

## Chapter 4: Design Techniques and Coding Standards for J2EE Projects

### OO Design Recommendations for J2EE Applications

- Achieving Loose Coupling with Interfaces
- Prefer Object Composition to Concrete Inheritance
- The Template Method Design Pattern
- The Strategy Design Pattern
- Using Callbacks to Achieve Extensibility
- The Observer Design Pattern
- Consider Consolidating Method Parameters
- Exception Handling – Checked or Unchecked Exceptions
  - Good Exception Handling Practices
  - Exceptions in J2EE
  - Making Exceptions Informative
- Using Reflection
  - Reflection Idioms
- Using JavaBeans to Achieve Flexibility
- Avoid a Proliferation of Singletons by Using an Application Registry
- Refactoring

### Coding Standards

- Start from the Standard
- Allocation of Responsibilities
- Avoid Code Duplication
- Avoid Literal Constants
- Visibility and Scoping
  - Public Instance Variables
  - Protected and Package Protected Instance Variables
  - Method Visibility
  - Variable Scoping
  - Inner Classes and Interfaces
- Using the final Keyword
  - Method Overriding and Final Methods
  - Final Classes
  - Final Instance Variables
- Implementing toString() Methods Useful for Diagnostics
- Defensive Coding Practices
  - Handle Nulls Correctly
  - Consider the Ordering of Object Comparisons
  - Use Short-circuit Evaluation
  - Distinguish Whitespace in Debug Statements and Error Messages
- Prefer Arrays to Collections in Public Method Signatures
- Documenting Code
- Logging
  - Choosing a Logging API
  - Logging in the EJB Tier
<table>
<thead>
<tr>
<th>Chapter 5: Requirements for the Sample Application</th>
<th>179</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
<td>180</td>
</tr>
<tr>
<td><strong>User Populations</strong></td>
<td>180</td>
</tr>
<tr>
<td>Public Internet Users</td>
<td>180</td>
</tr>
<tr>
<td>Box Office Users</td>
<td>181</td>
</tr>
<tr>
<td>Administrators</td>
<td>182</td>
</tr>
<tr>
<td><strong>Assumptions</strong></td>
<td>182</td>
</tr>
<tr>
<td><strong>Scope Limitations</strong></td>
<td>183</td>
</tr>
<tr>
<td><strong>Delivery Schedule</strong></td>
<td>183</td>
</tr>
<tr>
<td><strong>Internet User Interface</strong></td>
<td>183</td>
</tr>
<tr>
<td>Basic Workflow</td>
<td>184</td>
</tr>
<tr>
<td>Error Handling</td>
<td>185</td>
</tr>
<tr>
<td>Application Screens</td>
<td>185</td>
</tr>
<tr>
<td>Welcome Screen</td>
<td>188</td>
</tr>
<tr>
<td>Display Show Screen</td>
<td>189</td>
</tr>
<tr>
<td>Book Seats Screen</td>
<td>191</td>
</tr>
<tr>
<td>Show Reservation Screen</td>
<td>192</td>
</tr>
<tr>
<td>Payment Details Screen</td>
<td>194</td>
</tr>
<tr>
<td>Confirm Reservation Screen</td>
<td>196</td>
</tr>
<tr>
<td><strong>Box Office User Interface</strong></td>
<td>198</td>
</tr>
<tr>
<td><strong>Non-Functional Requirements</strong></td>
<td>198</td>
</tr>
<tr>
<td><strong>Hardware and Software Environment</strong></td>
<td>199</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>200</td>
</tr>
</tbody>
</table>

| Chapter 6: Applying J2EE Technologies        | 203 |
|------------------------------------------------|
| **When is a Distributed Architecture Appropriate?** | 204 |
# Table of Contents

## Deciding When to Use EJB
- Using EJB to Implement a Distributed Architecture 209
- Transaction Management
  - Transaction Management in J2EE Applications 210
  - Transaction Management and EJB 211
  - Transaction Management in the Sample Application 213
- EJB and Authorization 213
- EJB and Multi-threading 214
- Declarative Configuration Management 214
- The Downside of EJB
  - So Much Infrastructure 214
  - Programming Restrictions Applying to EJBs 215
  - The Singleton Problem in EJB 220
  - Timer Functionality 222
- EJBs in the Sample Application 222

## Deciding How to Use EJB
- What Should EJBs Do? 223
- When to Use Local or Remote Interfaces
  - Does it Make Sense for a Bean to Have Both Local and Remote Interfaces? 224
  - Phony Remote Interfaces 225
- EJB Interface Summary 226
- Using EJBs in the Sample Application 227

## Deciding when to Use Asynchronous Calling with JMS
- Message-Oriented Middleware (MOM) and JMS 228
- Producing Messages 228
- Consuming Messages
  - Consuming Messages without Using EJB 229
  - Consuming Messages with Message-Driven Beans (MDB) 229
- When to Use Asynchronous Calling
  - Indications for Using Messaging 232
  - Disadvantages of Using Messaging 233
  - JMS and Performance 233
  - Alternatives to JMS Messaging 234
- JMS in the Sample Application 234

## Authentication and Authorization
- The Standard Security Infrastructure 235
- The Server Implementation 237

## Deciding When to Use XML
- Using XSLT in J2EE Applications 238
- "Deep" Use of XML 239
- Converting Between JavaBeans and XML 241
- J2EE and XML in the Future 244
- XML in the Sample Application 245

## Caching to Improve Performance
- Caching Options 245
- A Caching Strategy for the Sample Application 247

## Summary 248
# Chapter 7: Data Access in J2EE Applications

## Data Access Goals

252

## Business Logic and Persistence Logic

252

## Object-Driven and Database-Driven Modeling: A Philosophical Debate

253

## O/R Mapping and the "Impedance Mismatch"

255

## The Data Access Object (DAO) Pattern

257

### Working with Relational Databases

259

- Referential Integrity
- Stored Procedures, Triggers, and Views
- RDBMS Performance Issues
  - RDBMS Performance Tuning
  - Denormalization

259

## Portability Versus Performance

263

## Exchanging Data in Distributed Applications

265

- The Value Object J2EE Pattern
- "Generic" Value Objects
- "Disconnected" Data Access Using JDBC Rowsets

265

## Common Data Access Issues

268

- Transaction Isolation
- Pessimistic and Optimistic Locking
- Primary Key Generation Strategies
  - Sequence Entity Bean
  - Unique ID Generation in Java
  - Database-Specific ID Generation
  - JDBC 3.0

268

## Where to Perform Data Access

273

- Data Access in the EJB Tier
  - Entity EJBs
  - Session EJBs and Helper Classes
- Data Access in the Middle Tier without Using EJB
- Data Access in the Web Tier
  - Servlets and Web-Specific Classes
- Data Access from JSP Pages

273

## Summary

278

## Data Modeling in the Sample Application

278
### Chapter 8: Data Access Using Entity Beans

**Entity Bean Concepts**
- Definition
- How Should We Use Entity Beans?
  - The Granularity Debate
  - The Business Logic Debate
  - Session Beans as Mediators

**CMP Versus BMP**

**Entity Beans in EJB 2.0**
- Local Interfaces
- Home Interface Business Methods
- EJB 2.0 CMP
  - Basic Concepts
  - Container-Managed Relationships (CMR)
  - EJB QL
  - Limitations of O/R Modeling with EJB 2.0 Entities
  - Custom Entity Behavior with CMP/BMP Hybrids

**Entity Bean Caching**
- Entity Bean Locking Strategies
  - Exclusive Locking
  - Database Locking
  - Read-only and "Read-mostly" Entities
  - Transactional Entity Caching

**Entity Bean Performance**

**Tool Support for Entity Beans**

**Summary**

### Chapter 9: Practical Data Access

**Data Access Technology Choices**
- SQL-Based Technologies
  - JDBC
  - SQLJ
- O/R Mapping Technologies
  - Established Commercial Products
  - Java Data Objects (JDO)
- Choosing a Data Access Strategy for the Sample Application

**JDBC Subtleties**
- Correct Exception Handling
- Extracting Information from SQLExceptions
- The PreparedStatement Question
# Table of Contents

## A Generic JDBC Abstraction Framework

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motivation</td>
<td>325</td>
</tr>
<tr>
<td>Aims</td>
<td>326</td>
</tr>
<tr>
<td>Exception Handling</td>
<td>327</td>
</tr>
<tr>
<td>A Generic Data-Access Exception Hierarchy</td>
<td>328</td>
</tr>
<tr>
<td>Converting JDBC Exceptions to Generic Exceptions</td>
<td>331</td>
</tr>
<tr>
<td>Two Levels of Abstraction</td>
<td>333</td>
</tr>
<tr>
<td>A Framework to Control JDBC Workflow and Error Handling</td>
<td>333</td>
</tr>
<tr>
<td>&quot;Inversion of Control&quot; Revisited</td>
<td>334</td>
</tr>
<tr>
<td>The com.interface21.jdbc.core package</td>
<td>334</td>
</tr>
<tr>
<td>Using the JdbcTemplate Class</td>
<td>340</td>
</tr>
<tr>
<td>A Higher Level of Abstraction: Modeling RDBMS Operations as Java Objects</td>
<td>342</td>
</tr>
<tr>
<td>Implementation of the com.interface21.jdbc.object Package</td>
<td>342</td>
</tr>
<tr>
<td>Using the JDBC Object Abstraction</td>
<td>347</td>
</tr>
<tr>
<td>JDBC Abstraction Summary</td>
<td>351</td>
</tr>
</tbody>
</table>

## Implementing the DAO Pattern in the Sample Application

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>360</td>
</tr>
</tbody>
</table>

## Chapter 10: Session Beans

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using Stateless Session Beans</td>
<td>364</td>
</tr>
<tr>
<td>Benefits of Stateless Session Beans</td>
<td>364</td>
</tr>
<tr>
<td>Stateless Session Beans and Internal State</td>
<td>365</td>
</tr>
<tr>
<td>Implications of Stateless Session Bean Pooling</td>
<td>365</td>
</tr>
<tr>
<td>Using Stateful Session Beans</td>
<td>366</td>
</tr>
<tr>
<td>Why Not to Use Stateful Session Beans</td>
<td>366</td>
</tr>
<tr>
<td>Performance and Scalability Issues</td>
<td>366</td>
</tr>
<tr>
<td>Reliability Issues</td>
<td>368</td>
</tr>
<tr>
<td>When to Use Stateful Session Beans</td>
<td>370</td>
</tr>
<tr>
<td>Session Synchronization</td>
<td>370</td>
</tr>
<tr>
<td>Protecting Stateful Session Beans from Concurrent Calls</td>
<td>371</td>
</tr>
<tr>
<td>Patterns for Achieving Stateful Functionality with SLSBs</td>
<td>371</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>372</td>
</tr>
<tr>
<td>Using a &quot;Required Workflow Exception&quot; to Mimic an SFSB State Machine</td>
<td>372</td>
</tr>
<tr>
<td>Using a Stateful Session Bean as Controller</td>
<td>373</td>
</tr>
</tbody>
</table>

## J2EE Design Patterns Applicable to Session Beans

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Session Façade Pattern in Distributed J2EE Applications</td>
<td>374</td>
</tr>
<tr>
<td>The EJB Command Design Pattern</td>
<td>374</td>
</tr>
<tr>
<td>Implementing the EJB Command Design Pattern</td>
<td>375</td>
</tr>
<tr>
<td>Advantages and Disadvantages of the EJB Command Design Pattern</td>
<td>377</td>
</tr>
<tr>
<td>Using Commands without Adopting the Command Design Pattern</td>
<td>378</td>
</tr>
</tbody>
</table>

## Session Bean Implementation Issues

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Handling in EJBs</td>
<td>379</td>
</tr>
<tr>
<td>The EJB Container's Behavior on Exceptions</td>
<td>380</td>
</tr>
<tr>
<td>Understanding EJB API Exceptions</td>
<td>381</td>
</tr>
<tr>
<td>Transaction Attributes for EJBs using CMT</td>
<td>382</td>
</tr>
<tr>
<td>The Business Methods Interface &quot;Pattern&quot;</td>
<td>386</td>
</tr>
</tbody>
</table>
Table of Contents

**Session Beans in the Sample Application**

Summary  389

**Chapter 11: Infrastructure and Application Implementation**  393

**Infrastructure**

Goals of a Strong Infrastructure  394
Using a Framework to Configure Application Components  395

- The Problem  396
- Using JavaBeans  396
- Using a "Bean Factory"  397
- The Application Context  401
- Testing Implications  406
- Summary of Application Configuration Infrastructure  409

Managing API Complexity  410

- Implementing EJBs  410
- Accessing EJBs  417
- Using JMS  425

**Implementing Business Logic**

Implementing the Sample Application  428

- Defining Business Interfaces  428
- Determining Implementation Strategy  429
- Implementing the BoxOffice  433
- Using JMS to Propagate Data Updates  435
- Pulling It All Together  436

Summary  437

**Chapter 12: Web-Tier MVC Design**  441

**The Challenges of Web Development**  442

**Lessons Learned in Java Web Development**

- The Shortcomings of Servlet-only Solutions  443
- JSP: Promise and Temptation  444
- "JSP Model 1" Architecture  444
- The Temptation of the JSP Standard Infrastructure  445

Striking a Balance  446

**Web-Tier Design Goals**

- A Clean Web Tier  447
- A Thin Web Tier  447

**MVC Concepts and the Front Controller J2EE Pattern**

- Concepts  448
- The MVC Triad  449
- Control Flow  452
<table>
<thead>
<tr>
<th>Pattern Variants</th>
<th>453</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Selection Servlet</td>
<td>453</td>
</tr>
<tr>
<td>How Many Controller Servlets?</td>
<td>454</td>
</tr>
<tr>
<td>JSP or Servlet Controller?</td>
<td>454</td>
</tr>
<tr>
<td>Should a Request Cause the Creation of a Command?</td>
<td>455</td>
</tr>
<tr>
<td>Implementation Goals</td>
<td>455</td>
</tr>
<tr>
<td><strong>Web Application Frameworks</strong></td>
<td>456</td>
</tr>
<tr>
<td>Common Concepts</td>
<td>456</td>
</tr>
<tr>
<td>Available Frameworks</td>
<td>457</td>
</tr>
<tr>
<td>Struts</td>
<td>457</td>
</tr>
<tr>
<td>Maverick</td>
<td>461</td>
</tr>
<tr>
<td>WebWork</td>
<td>463</td>
</tr>
<tr>
<td><strong>Integrating a Web Application Framework into Overall Application Architecture</strong></td>
<td>465</td>
</tr>
<tr>
<td><strong>The Web Application Framework Used in the Sample Application</strong></td>
<td>467</td>
</tr>
<tr>
<td>Design Goals</td>
<td>468</td>
</tr>
<tr>
<td>Basic MVC Control Flow</td>
<td>469</td>
</tr>
<tr>
<td>Controller Servlet</td>
<td>471</td>
</tr>
<tr>
<td>Request Controller (com.interface21.web.servlet.mvc.Controller)</td>
<td>474</td>
</tr>
<tr>
<td>Models</td>
<td>475</td>
</tr>
<tr>
<td>Views</td>
<td>476</td>
</tr>
<tr>
<td>ViewResolver</td>
<td>477</td>
</tr>
<tr>
<td>ContextLoaderServlet</td>
<td>478</td>
</tr>
<tr>
<td>Custom Tags</td>
<td>479</td>
</tr>
<tr>
<td>Workflow Refinements</td>
<td>479</td>
</tr>
<tr>
<td>Examples</td>
<td>480</td>
</tr>
<tr>
<td>A Basic Controller Implementation</td>
<td>481</td>
</tr>
<tr>
<td>A Controller Exposing Bean Properties</td>
<td>482</td>
</tr>
<tr>
<td>A Multi-Action Controller</td>
<td>484</td>
</tr>
<tr>
<td><strong>Web-Tier Session Management</strong></td>
<td>488</td>
</tr>
<tr>
<td>Session State Managed by the J2EE Server</td>
<td>488</td>
</tr>
<tr>
<td>Clustering and Replication</td>
<td>488</td>
</tr>
<tr>
<td>Simple Optimizations</td>
<td>489</td>
</tr>
<tr>
<td>Session State Held in the Browser</td>
<td>490</td>
</tr>
<tr>
<td>Session State Management with Cookies</td>
<td>490</td>
</tr>
<tr>
<td>Session State Management with Hidden Form Fields</td>
<td>491</td>
</tr>
<tr>
<td><strong>Processing User Input</strong></td>
<td>492</td>
</tr>
<tr>
<td>Data Binding and Displaying Input Errors for Resubmission</td>
<td>492</td>
</tr>
<tr>
<td>Approaches to Data Binding in MVC Frameworks</td>
<td>493</td>
</tr>
<tr>
<td>JSP Custom Tags</td>
<td>494</td>
</tr>
<tr>
<td>Data Validation</td>
<td>496</td>
</tr>
<tr>
<td>Where Should Data Validation be Performed?</td>
<td>496</td>
</tr>
<tr>
<td>Data Validation in the Framework Described in this Chapter</td>
<td>498</td>
</tr>
<tr>
<td><strong>Implementing the Web Tier in the Sample Application</strong></td>
<td>506</td>
</tr>
<tr>
<td>Overview</td>
<td>506</td>
</tr>
<tr>
<td>Handling a Seat Reservation Request</td>
<td>508</td>
</tr>
<tr>
<td>Implementation Review</td>
<td>511</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>512</td>
</tr>
</tbody>
</table>
# Table of Contents

## Chapter 12: Views in the Web Tier

### Decoupling Controllers and Views

- 517

### Constructing the View for the Reservations Page

- Information Presented and Required Formatting
- 519
- The Model Behind this View
- 521
- Model Principles
- 524

### JSP Views

- What We Want to Avoid
- 527
- How to Use JavaBeans in JSP Pages
- 531
- JSP Custom Tags
- 532
  - The Java Standard Tag Library
  - Other Third-Party Tag Libraries
  - Implementing Your Own Tag Libraries
  - Guidelines for Custom Tag Use
- 536
- Guidelines for JSP Use
- 536
- Looking Ahead: Implications of JSP 2.0
- 538
- A JSP View for the Example
- 538
- JSP Summary
- 543

### Dedicated Template Languages

- 544
  - Common Concepts
  - WebMacro
  - Velocity
    - Velocity Concepts
    - A Velocity Template for our Example
    - Velocity Summary
- FreeMarker
- 551

### XSLT

- 552
  - When to Use XSLT
  - What Do We Want from XSL?
  - How to Use XSLT in Views
    - Using XSLT Instead of JSP
    - Using XSLT from JSP Custom Tags
- Implementing our Example Using a "Pure" XSLT Approach
- 555

### Alternative Approaches to Markup Generation

- 561
  - HTML Generation Libraries
  - XMLC
    - An XMLC Template for Our Example
    - Compiling the Template
    - Manipulating the XMLC Object Generated from the Template
  - Further Reading on XMLC
- 570

### Generating Binary Content

- 571
  - Generating PDF with iText
- 571

### View Composition and Page Layout

- 575

### Summary

- 579
# Chapter 14: Packaging and Application Deployment

## Packaging
- Deployment Units
- Expanded Deployment Units
- Understanding J2EE Class Loading
  - Java Class Loading Concepts
  - Class Loading in J2EE
  - Server Check List
  - Recommendations
  - Further Information
- Packaging the Sample Application

## Application Deployment: Common Concepts
- Configuring a Server to Run the Application
  - Creating Connection Pools
  - Creating JMS Destinations
  - Setting up Authentication
  - Installing Libraries
- Writing Proprietary Deployment Descriptors for an Application
  - EJB-Specific Configuration
  - Web-Tier Configuration
- Deploying an Application
- Deployment Parameters for the Sample Application

## Deploying the Sample Application on JBoss 3.0
- Understanding the JBoss Directory Structure
- Configuring a JBoss Server to Run the Sample Application
  - Creating a Connection Pool
  - Creating JMS Destinations
  - Installing the Service Definition File
  - Reviewing Configuration
- Writing JBoss Deployment Descriptors for the Sample Application
- Deploying the Application

## Summary

# Chapter 15: Performance Testing and Tuning an Application

## Strategic Issues and Definitions
- Performance and Scalability
- Setting Clear Goals for Performance and Scalability
- Design Versus Code Optimization

## Tools for Testing Performance and Throughput
- Preparing to Benchmark
- Web Test Tools
  - Microsoft Web Application Stress Tool
- Non-Web Testing Tools

## Summary
# Table of Contents

## Locating Performance or Scalability Problems
- Testing in Layers ............................................. 623
- Profiling Tools .............................................. 623
  - JVM Profiling Options .................................. 624
  - The JProbe Profiler .................................... 625

## Addressing Performance or Scalability Problems
- Server Choice and Server Configuration .................. 630
- Dispensing with Redundant Container Services ......... 631
- Caching ..................................................... 632
  - When to Cache ......................................... 632
  - Where to Cache ....................................... 633
  - Third-party Caching Products for Use in J2EE Applications 637
- Code Optimization ......................................... 638

## Case Study: The "Display Show" Page in the Sample Application

## Performance in Distributed Applications
- The Overhead of Remote Method Invocation (RMI) .... 653
- Minimizing Remote Calls ................................ 655
  - Application Partitioning ............................... 655
  - Consolidating Remote Calls ......................... 658
- Moving Data Efficiently ................................ 658
  - Serialization Optimizations ......................... 659
  - Other Data Transfer Strategies ..................... 662
- Collocating Components in the Same JVM ............... 663

## Web-Tier Performance Issues
- View Performance ......................................... 663
- Web Caching Using HTTP Capabilities .................. 666
  - Cache Control HTTP Headers ......................... 666
  - Using the Servlet API to Control Caching .......... 668
  - Implications for MVC Web Applications ............. 669
  - The Welcome Page in the Sample Application ....... 670
- Edged Side Caching and ESI .............................. 671

## The Primary Causes of Performance and Scalability Problems in J2EE Applications

## Summary ................................................. 673

## Chapter 16: Conclusion: Making J2EE Work for You
- General Principles ....................................... 676
- Projects .................................................. 679

## Appendix A: Implementing View Technologies
- Decoupling Controllers from View Technologies Using a View Interface 684
- View Implementations .................................... 686
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JSP</strong></td>
<td></td>
</tr>
<tr>
<td>Configuring the JSTL</td>
<td>688</td>
</tr>
<tr>
<td>The InternalResourceView View Implementation</td>
<td>688</td>
</tr>
<tr>
<td>Defining JSP Views for Use in an Application</td>
<td>689</td>
</tr>
<tr>
<td><strong>Velocity</strong></td>
<td></td>
</tr>
<tr>
<td>Installing and Configuring Velocity</td>
<td>691</td>
</tr>
<tr>
<td>Implementing the View Interface for Velocity</td>
<td>692</td>
</tr>
<tr>
<td>Exposing Model Data to a Velocity Template</td>
<td>693</td>
</tr>
<tr>
<td>Providing Support for Date and Currency Formatting</td>
<td>694</td>
</tr>
<tr>
<td>Defining Velocity Views for Use in an Application</td>
<td>695</td>
</tr>
<tr>
<td><strong>XSLT</strong></td>
<td></td>
</tr>
<tr>
<td>Installing Dornify</td>
<td>696</td>
</tr>
<tr>
<td>Implementing the View Interface for XSLT</td>
<td>697</td>
</tr>
<tr>
<td>Performing XSLT transforms</td>
<td>697</td>
</tr>
<tr>
<td>Date and Currency Formatting Support</td>
<td>700</td>
</tr>
<tr>
<td>Defining XSLT Views for Use in an Application</td>
<td>701</td>
</tr>
<tr>
<td><strong>XMLC</strong></td>
<td></td>
</tr>
<tr>
<td>Installing and Configuring XMLC</td>
<td>703</td>
</tr>
<tr>
<td>Implementing the View Interface for XMLC</td>
<td>704</td>
</tr>
<tr>
<td>Defining XMLC Views for Use in an Application</td>
<td>705</td>
</tr>
<tr>
<td><strong>Generating PDF with iText</strong></td>
<td></td>
</tr>
<tr>
<td>Installing iText</td>
<td>705</td>
</tr>
<tr>
<td>Implementing the View Interface for PDF Generation with iText</td>
<td>705</td>
</tr>
<tr>
<td>Defining PDF Views for Use in an Application</td>
<td>707</td>
</tr>
<tr>
<td><strong>Additional Views</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Custom Views</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>711</td>
</tr>
</tbody>
</table>