

1. Record Nr.	UNINA9910809963203321
Autore	Moore Bill
Titolo	WebSphere Application Server-Express : a development example for new developers // [Bill Moore ... et al.]
Pubbl/distr/stampa	[S.I.], : IBM, International Technical Support Organization, c2003
Edizione	[1st ed.]
Descrizione fisica	1 online resource (706 p.)
Collana	IBM redbooks
Altri autori (Persone)	MooreBill
Disciplina	005.2/762
Soggetti	Web site development Application software - Development Internet programming
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"November 2003."
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Front cover -- Contents -- Notices -- Trademarks -- Preface -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Part 1 The development process -- Chapter 1. Introduction -- 1.1 Our objectives -- 1.1.1 WebSphere Application Server - Express defined -- 1.2 The focus of this redbook -- 1.2.1 WebSphere Application Server - Express users -- 1.2.2 Application stages -- 1.3 How to use this redbook -- Chapter 2. Getting started -- 2.1 The WebSphere platform -- 2.1.1 Application servers -- 2.1.2 Development environment -- 2.1.3 Which studio do I use for which server? -- 2.1.4 What if I have an iSeries? -- 2.2 What is WebSphere Application Server - Express -- 2.2.1 Simple to install and administer -- 2.2.2 Simplifies enterprise-class tools and technologies -- 2.2.3 Easy migration path to other servers in the family -- 2.2.4 Summary of features -- 2.2.5 What can I do with WebSphere Application Server - Express? -- 2.3 Architecture and components -- 2.3.1 Web container -- 2.4 Key technologies -- 2.4.1 Java 2 Platform, Enterprise Edition -- 2.4.2 Java Database Connectivity -- 2.4.3 XML -- 2.4.4 Jakarta -- Chapter 3. Development tools -- 3.1 WebSphere Application Server - Express, V5.0 -- 3.1.1 WebSphere Application Server - Express components -- 3.1.2 Deploying WebSphere Application Server - Express -- 3.1.3 Eclipse technology -- 3.1.4 WebSphere Application Server - Express tooling -- 3.1.5 WebSphere Studio -- 3.1.6 Express

Application Server: Overview -- 3.2 Information road map -- 3.2.1 Installation -- 3.2.2 Migration -- 3.2.3 Application development -- 3.3 Using WebSphere Studio -- 3.3.1 Perspectives -- 3.3.2 Editors -- 3.3.3 Customizing perspectives -- 3.3.4 Using fast views -- 3.3.5 Setting perspective preferences -- 3.3.6 The workspace -- 3.3.7 Recovering files from their local history -- 3.3.8 Finding help.

3.4 Struts -- 3.4.1 Struts overview -- 3.4.2 Model-View-Controller architecture -- 3.4.3 WebSphere Studio and Struts -- 3.4.4 Web applications -- 3.4.5 Struts details -- 3.4.6 Why we use Struts -- 3.4.7 Components of a Struts-based application -- 3.4.8 Configuration of a Struts-based application -- 3.5 Developing in a team environment -- 3.5.1 Team environment using WebSphere Studio -- 3.5.2 What is Concurrent Versions System -- 3.5.3 Using Concurrent Versions System with WebSphere Studio -- 3.5.4 Working in a team using CVS -- 3.6 Common development tasks -- 3.6.1 Creating projects in Studio -- 3.6.2 Create a Java package -- 3.6.3 Create a folder for Web content -- 3.6.4 Create a Java class -- 3.6.5 Required common code -- 3.6.6 Create a Struts Web diagram -- 3.6.7 Create an SQL statement -- 3.6.8 Generate a JavaBean from an SQL statement -- Chapter 4. The development process -- 4.1 Development process basics -- 4.1.1 Definition of a development process -- 4.1.2 Importance of a development process -- 4.1.3 Realization of a development process -- 4.1.4 Development process principles -- 4.2 Starting a project -- 4.2.1 Understanding your business today -- 4.2.2 Where do you want to go -- 4.2.3 An initial roadmap of how to get there -- 4.3 Understanding and planning a project -- 4.4 Building a solution -- 4.5 Project hand-over -- 4.6 RealEstate application architecture -- 4.6.1 Component-based architecture -- 4.6.2 Layered design -- 4.6.3 Package structure -- 4.6.4 Naming conventions -- Part 2 The sample solution -- Chapter 5. Requirements -- 5.1 The business problem -- 5.1.1 Business background -- 5.1.2 Business problems -- 5.2 Business objectives of the solution -- 5.2.1 Put company information online -- 5.2.2 Make all property information available in one place -- 5.2.3 Communicate with customers online -- 5.2.4 Benefit internal management.

5.3 Functional requirements -- 5.3.1 Who are the application users? -- 5.3.2 What can each group of users do? -- 5.3.3 Customer registration -- 5.3.4 Agents account maintenance -- 5.3.5 Property search -- 5.3.6 Customer interest list -- 5.3.7 Business reports -- 5.3.8 E-mail communication -- 5.3.9 Error handling -- 5.4 Summary -- Chapter 6. Building a prototype -- 6.1 Introduction to prototypes -- 6.1.1 Benefits of building a prototype -- 6.1.2 Before you start building -- 6.2 Tools to use when building a prototype -- 6.3 Creating a prototype to our sample application -- 6.3.1 Create the Web project -- 6.3.2 Create the menus -- 6.3.3 Update and use of the CSS file -- 6.3.4 Create the logo -- 6.3.5 Search properties -- Chapter 7. Application architecture -- 7.1 Overview of the architecture -- 7.1.1 Component-based development -- 7.1.2 Layered application design -- 7.1.3 Usage of design patterns -- 7.2 Component architecture -- 7.2.1 PropertyCatalog component -- 7.2.2 News component -- 7.2.3 E-mail component -- 7.2.4 InterestList component -- 7.2.5 Reporting component -- 7.2.6 User component -- 7.2.7 Dependencies between the components -- 7.3 Layered architecture -- 7.3.1 Presentation layer -- 7.3.2 Controller layer -- 7.3.3 Business facade layer -- 7.3.4 Domain layer -- 7.3.5 Data access layer -- Chapter 8. Design and specification -- 8.1 High level design -- 8.1.1 Screen flows -- 8.1.2 User management and security -- 8.1.3 Database operations -- 8.1.4 User input validation -- 8.1.5 E-mail -- 8.1.6 Infrastructure -- 8.1.7 Design of the managers -- 8.2 Detailed design -- 8.2.1 Core functionality -- 8.2.2 Presentation -- 8.2.3

Requests and responses -- 8.2.4 Database accessors -- 8.2.5 -- 8.2.6  
Logging -- Chapter 9. User registration and security -- 9.1 User  
management basics -- 9.1.1 Underlying implementation.  
9.2 User management component specification -- 9.2.1 User  
management component: the actors -- 9.2.2 What the users can do:  
The use cases -- 9.2.3 View user details: UC.ViewUserDetails -- 9.2.4  
Request password: UC.RequestPassword -- 9.2.5 Register customer:  
UC.RegisterCustomer -- 9.2.6 Modify customer details: UC.  
ModifyCustomerDetails -- 9.2.7 Log out: UC.Logout -- 9.2.8 Log in:  
UC.login -- 9.2.9 List users: UC.ListUsers -- 9.2.10 Delete customer  
details: UC.DeleteCustomerDetails -- 9.2.11 Dependencies -- 9.2.12  
Interfaces for other components -- 9.3 Building the user management  
component -- 9.3.1 Preparing the workspace -- 9.3.2 Creating the Java  
packages -- 9.3.3 Creating the Web section -- 9.4 Strategy -- 9.5 The  
business layer (manager) -- 9.5.1 LDAP Implementation -- 9.6 The  
model layer (DAO) -- 9.7 The domain layer (DTO) -- 9.8 The  
presentation layer (Struts) -- 9.8.1 Log in/out -- 9.8.2 Registration --  
9.8.3 User details management -- 9.8.4 Administrative  
view/modify/delete user details -- 9.8.5 Password e-mail request --  
9.8.6 Security and guarding pages -- 9.9 Testing the component --  
9.10 Room for improvement -- 9.11 Summary -- Chapter 10. Property  
catalog -- 10.1 Property catalog component specification -- 10.1.1 Use  
cases -- 10.1.2 Dependencies -- 10.1.3 Interface -- 10.2 Building the  
property catalog component -- 10.2.1 Preparing the workspace --  
10.2.2 Presentation layer -- 10.2.3 Controller layer -- 10.2.4 Business  
facade layer -- 10.2.5 Domain layer -- 10.2.6 Data access layer --  
10.2.7 Putting everything together -- 10.2.8 Testing the component --  
Chapter 11. Interest lists -- 11.1 Interest list basics -- 11.1.1 What is  
implemented? -- 11.2 Interest list component specification -- 11.2.1  
Users of the interest list: the actors -- 11.2.2 What the users can do:  
the use cases -- 11.2.3 Show interest list: ILC.ShowInterestList.  
11.2.4 Add property to interest list: ILC.AddProperty -- 11.2.5 Update  
interest list: ILC.UpdateInterestList -- 11.2.6 Clear interest list: ILC.  
ClearInterestList -- 11.2.7 Return to property catalog: ILC.  
ReturnToPropertyCatalog -- 11.2.8 Checkout interest list: ILC.  
CheckoutInterestList -- 11.2.9 Dependencies -- 11.2.10 Interfaces for  
other components -- 11.3 How to work faster and write stable code --  
11.3.1 Use a component-based development process -- 11.3.2 Use a  
layered application framework -- 11.4 Building the interest list  
component -- 11.4.1 Preparing the workspace -- 11.4.2 Creating the  
Java packages -- 11.4.3 Creating the Web section -- 11.5 A quick word  
about the database -- 11.6 Strategy -- 11.7 The business layer  
(manager) -- 11.8 The model layer (DAO) -- 11.8.1 What needs to be  
implemented? -- 11.8.2 Building the DAO revealed -- 11.9 The domain  
layer (DTO) -- 11.10 The presentation layer (Struts) -- 11.10.1 What  
we need to end up with -- 11.10.2 Start with our adapter -- 11.10.3  
Create the Struts Web diagram -- 11.10.4 Create the form bean --  
11.10.5 Create the action -- 11.10.6 Create the JSP or Web page node  
-- 11.11 Testing the component -- 11.11.1 Add property -- 11.11.2  
Update property -- 11.11.3 Remove property -- 11.11.4 Clear list --  
11.11.5 Checkout list -- 11.12 Room for improvement -- 11.13  
Summary -- Chapter 12. E-mail -- 12.1 JavaMail API -- 12.1.1  
Protocols -- 12.1.2 Core classes -- 12.1.3 Example: Sending an e-mail  
-- 12.2 Build sample e-mail application -- 12.2.1 Use cases -- 12.2.2  
Business facade layer -- 12.2.3 Domain layer -- 12.3 Configuring  
JavaMail resources -- 12.4 Utility tag library mailer -- Chapter 13.  
Administering the catalog -- 13.1 Component specification -- 13.1.1  
Use cases -- 13.1.2 Dependencies -- 13.1.3 Interface -- 13.2 Building

property catalog administration -- 13.2.1 Preparing the workspace.  
13.2.2 Presentation layer.

---

Sommario/riassunto

This IBM Redbooks publication is a practical guide to developing Web applications using WebSphere Studio. WebSphere Studio is used to develop a sample Web application targeted to the WebSphere Application Server - Express runtime platform. It implements a sample scenario based on realistic requirements for small and medium customers, and provides an end-to-end guide for the development of this scenario. The book focuses on describing a simple process that allows non technical readers to understand and participate in the development of Web applications using WebSphere Studio . Part 1 of the book is an overview of the Web application development process and the tools available within WebSphere Studio. Part 2 describes the requirements and design of a sample application and details the process, architecture, and tools used to develop that sample. It is a detailed explanation of the steps needed to build the sample application. This illustrates how to achieve areas of functionality commonly found in dynamic Web applications, and provides worked examples that new developers can use in their own applications. Please note that the additional material referenced in the text is not available from IBM.

---