

1. Record Nr.	UNINA9910824525503321
Autore	Endrei Mark
Titolo	Patterns : self-service application solutions using WebSphere V5.0 / / Mark Endrei, Min Luo, Margo Pulles
Pubbl/distr/stampa	Austin, TX, : IBM, International Technical Support Organization, 2003
Edizione	[1st ed.]
Descrizione fisica	1 online resource (476 p.)
Collana	Redbooks
Altri autori (Persone)	LuoMin PullesMargo
Soggetti	Client/server computing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes index.
Nota di contenuto	Front cover -- Contents -- Notices -- Trademarks -- Preface -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Part 1 Self-Service patterns -- Chapter 1. Patterns for e-business -- 1.1 The Patterns for e-business layered asset model -- 1.2 How to use the Patterns for e-business -- 1.2.1 Business, Integration, or Composite pattern, or a custom design -- 1.2.2 Selecting Application patterns -- 1.2.3 Review Runtime patterns -- 1.2.4 Review Product mappings -- 1.2.5 Review guidelines and related links -- 1.3 Summary -- Chapter 2. The Self-Service business pattern -- 2.1 Self-Service applications -- 2.2 Self-Service application patterns -- 2.3 Application patterns used in this book -- Chapter 3. Runtime pattern -- 3.1 An introduction to the node types -- 3.1.1 Web application server node -- 3.1.2 User node -- 3.1.3 Public Key Infrastructure (PKI) -- 3.1.4 Domain Name System (DNS) node -- 3.1.5 Protocol firewall node -- 3.1.6 Domain firewall node -- 3.1.7 Directory and security services node -- 3.1.8 Database server node -- 3.1.9 Existing applications and data node -- 3.1.10 Web server redirector node -- 3.1.11 Application server node -- 3.2 Runtime patterns for Stand-Alone Single Channel -- 3.2.1 Basic Runtime pattern -- 3.2.2 Runtime pattern: Variation 1 -- 3.2.3 Runtime pattern: Other variations -- 3.3 Runtime patterns for Directly Integrated Single Channel -- 3.3.1 Basic Runtime pattern -- 3.3.2 Runtime pattern: Variation 1 -- 3.3.3 Runtime pattern: Other variations -- Chapter 4. Product mappings --

4.1 IBM WebSphere Application Server -- 4.1.1 IBM WebSphere Application Server Express V5.0 -- 4.1.2 IBM WebSphere Application Server base V5.0 -- 4.1.3 IBM WebSphere Application Server Network Deployment V5.0 -- 4.1.4 IBM WebSphere Application Server Enterprise V5.0 -- 4.2 IBM CICS.

4.2.1 IBM CICS Transaction Server -- 4.2.2 CICS Transaction Gateway -- 4.3 IBM WebSphere MQ -- 4.4 Product mappings for Stand-Alone Single Channel -- 4.5 Product mappings for Directly Integrated Single Channel -- Part 2 Guidelines -- Chapter 5. Technology options -- 5.1 Web client -- 5.1.1 Web browser -- 5.1.2 HTML -- 5.1.3 Dynamic HTML -- 5.1.4 Cascading Style Sheets (CSS) -- 5.1.5 JavaScript -- 5.1.6 Java applets -- 5.1.7 XML (client side) -- 5.1.8 XHTML 1.1 (HTML 4.01) -- 5.1.9 VoiceXML -- 5.1.10 XForms -- 5.1.11 Mobile clients -- 5.2 Web application server -- 5.2.1 Java servlets -- 5.2.2 JavaServer Pages (JSPs) -- 5.2.3 JavaBeans -- 5.2.4 XML -- 5.2.5 Enterprise JavaBeans -- 5.2.6 Additional enterprise Java APIs -- 5.3 Integration technologies -- 5.3.1 Web services -- 5.3.2 J2EE Connector Architecture -- 5.3.3 Java Message Service -- 5.3.4 Others -- 5.4 Where to find more information -- Chapter 6. Application and system design -- 6.1 Application structure -- 6.1.1 Model-View-Controller design pattern -- 6.1.2 Result bean design pattern -- 6.1.3 View bean design pattern -- 6.1.4 Formatter beans design pattern -- 6.1.5 Command bean design pattern -- 6.1.6 Frameworks -- 6.1.7 WebSphere command framework with EJBs -- 6.1.8 Best practices for EJBs -- 6.2 Design guidelines for Web services -- 6.2.1 Web services architecture -- 6.2.2 Web services design considerations -- 6.2.3 The key challenges in Web services -- 6.2.4 Best practices for Web services -- 6.2.5 Web services and Microsoft .NET -- 6.3 Design guidelines for J2EE Connectors -- 6.3.1 Components of J2EE Connector Architecture -- 6.3.2 Managed and non-managed environments -- 6.3.3 Common Connector Interface -- 6.3.4 The CCI classes -- 6.3.5 System contracts -- 6.3.6 Migration issues -- 6.3.7 CICS resource adapters -- 6.3.8 Selecting a CICS resource adapter -- 6.3.9 Asynchronous calls. 6.3.10 CICS ECI design considerations -- 6.3.11 Best practices for J2EE Connector Architecture -- 6.4 Design guidelines for Java Message Service -- 6.4.1 Message models -- 6.4.2 JMS point-to-point model -- 6.4.3 JMS publish/subscribe model -- 6.4.4 JMS messages -- 6.4.5 Synchronous vs. asynchronous design considerations -- 6.4.6 Where to implement message producers and consumers -- 6.4.7 Message-driven beans -- 6.4.8 Managing JMS objects -- 6.4.9 JMS and JNDI -- 6.4.10 Embedded JMS Provider vs. WebSphere MQ -- 6.4.11 WebSphere to MQ connection options -- 6.4.12 Best practices for JMS and IBM WebSphere MQ -- 6.4.13 More information -- Chapter 7. Application development -- 7.1 Application development methodology -- 7.2 Development guidelines for Web services -- 7.2.1 Transmission patterns -- 7.2.2 SOAP messaging mechanism -- 7.2.3 Web service application development steps -- 7.2.4 IBM WebSphere Studio Application Developer -- 7.2.5 WebSphere V5.0 support for Web services -- 7.2.6 Other Web services packages from IBM -- 7.3 Development guidelines for J2EE Connectors -- 7.3.1 Creating a J2EE Connector application using native CCI -- 7.3.2 Enterprise Services overview -- 7.3.3 Using Enterprise Services toolkit -- 7.3.4 Data conversion -- 7.3.5 Migration to other J2EE Connector resource adapters -- 7.4 Development guidelines for JMS -- 7.4.1 Creating a JMS application -- 7.4.2 WebSphere Studio JMS development -- 7.4.3 IBM WebSphere MQ client or server -- 7.5 Development guidelines for XML messaging -- 7.5.1 Creating an XML Message -- 7.5.2 Creating Java classes -- 7.5.3 WebSphere Studio XML tools -- 7.6 Design modeling

using Rational XDE -- 7.6.1 Creating an XDE model -- 7.6.2 Modeling behavior -- 7.6.3 Modeling structure -- 7.6.4 Round-trip engineering models and code -- Chapter 8. Systems management -- 8.1 Web services system management.

8.1.1 Security considerations -- 8.1.2 Creating Secure Web services -- 8.1.3 Web service security specifications -- 8.1.4 Web services component security -- 8.2 J2EE Connectors and CICS system management -- 8.2.1 Logging and tracing -- 8.2.2 Performance monitoring and tuning -- 8.2.3 Scalability and availability considerations -- 8.2.4 Security considerations -- 8.3 Java Message Service system management -- 8.3.1 JMS administration -- 8.3.2 JMS performance issues -- 8.3.3 WebSphere MQ design -- 8.3.4 WebSphere MQ administration management -- 8.3.5 WebSphere MQ monitoring -- 8.3.6 WebSphere MQ restart and recovery -- 8.3.7 Managing log files -- 8.3.8 Backing up WebSphere MQ resources -- 8.3.9 Security considerations -- Part 3 Technical scenarios -- Chapter 9. PDK sample overview -- 9.1 The Patterns Development Kit -- 9.2 PDK use cases -- 9.2.1 Display Funds -- 9.2.2 Transfer Funds -- 9.3 Architectural overview model -- 9.4 System design overview -- 9.4.1 PDK component model -- 9.4.2 PDK object model -- 9.5 Lower-level design considerations -- 9.5.1 Validation of client input -- 9.5.2 Singleton factories -- 9.5.3 Model-view-controller implementation -- 9.5.4 Using the WebSphere Command Framework -- 9.5.5 Session EJB facade -- 9.5.6 Caching of EJB home references -- 9.5.7 Using two-phase commit -- 9.6 PDK development -- 9.6.1 Tools and APIs -- 9.6.2 PDK project structure -- 9.6.3 Team development -- 9.6.4 Testing -- 9.7 PDK runtime -- 9.8 More information -- Chapter 10. Web services scenario -- 10.1 Architectural overview model -- 10.2 System design overview -- 10.2.1 Component model -- 10.2.2 Object model -- 10.3 Enterprise application design -- 10.4 XML messaging design and development -- 10.4.1 XML message structure -- 10.4.2 XML DTD and XML Schema -- 10.4.3 XML and JavaBean -- 10.5 Application development using Web services.

10.5.1 Preparing for development -- 10.5.2 Approach to developing the sample Web services components -- 10.5.3 Creating the enterprise-tier application -- 10.5.4 Creating the Web service -- 10.5.5 Using the Web Service wizard -- 10.5.6 Examining the generated files -- 10.5.7 SOAP router servlets -- 10.5.8 Viewing the deployed Web services -- 10.5.9 Web service client proxy -- 10.5.10 Web service test client -- 10.5.11 Creating the Web service requester -- Chapter 11. J2EE Connector Architecture scenario -- 11.1 Architectural overview model -- 11.2 System design overview -- 11.2.1 Component model -- 11.2.2 Object model -- 11.3 Low-level design considerations -- 11.3.1 Connection management -- 11.3.2 Creating the input and output record -- 11.3.3 Data conversion -- 11.3.4 Executing the enterprise application -- 11.3.5 Transaction management -- 11.3.6 Security -- 11.4 Application development using J2EE Connectors -- 11.4.1 Installing the J2EE Connector resource adapter -- 11.4.2 Create a session EJB -- 11.4.3 Testing the session EJB -- 11.5 Runtime configuration -- 11.5.1 Runtime configuration design -- 11.5.2 WebSphere Application Server configuration -- 11.5.3 CICS Transaction Server configuration -- Chapter 12. Java Message Service scenario -- 12.1 Architectural overview model -- 12.2 System design overview -- 12.2.1 Component model -- 12.2.2 Object model -- 12.3 Low-level design considerations -- 12.3.1 Point-to-point messaging model -- 12.3.2 Message structure -- 12.3.3 JMS resource lookups using JNDI -- 12.3.4 Managing JMS resources -- 12.3.5 Synchronous vs. asynchronous -- 12.3.6 Message selectors -- 12.3.7

12.3.6 Message selectors -- 12.3.7

12.3.6 Message selectors -- 12.3.7

12.3.6 Message selectors -- 12.3.7

12.3.6 Message selectors -- 12.3.7

12.3.6 Message selectors -- 12.3.7

Message time to live -- 12.3.8 Persistent vs. non-persistent messages
-- 12.3.9 Enterprise application design -- 12.3.10 Alternate
publish/subscribe design -- 12.4 Development environment
configuration.
12.4.1 Adding JMS resources to the server configuration.
