



北京希望电子出版社
www.bhp.com.cn

21 世纪新工具软件开发指南丛书

5

交互式 Web 应用程序开发指南

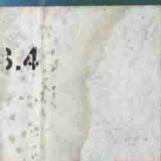
德博拉·科廷翰 著



本光盘内容是：
本版电子书

Creating an Interactive B2B Web Application

英文版



Creating an Interactive B2B Web Application



本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制

交互式 Web 应用程序开发指南

德博拉·科廷翰 著



本光盘内容是：
本版电子书

Creating an Interactive B2B Web Application

英文版

内 容 简 介

本书是一本关于“商务过程管理”的英文版最新专著，主旨是指导用户使用 IBM 公司的战略性电子开发软件（包括 VisualAge for Java、WebSphere Studio 和 WebSphere 应用程序服务器）来创建交互式 Web 应用程序。用户通过使用 S / 390 服务器环境和现实的场景，可以学会如何在最新的 Web 应用程序中访问 CICS、DB2 数据库系统、IMS 和 MQSeries 的企业资源。

全书由 3 部分和 1 个附录组成。第 1 部分是场景概述，分 3 章，描述了总的场景，包括服务器环境的配置、创建 Web 应用程序的步骤，以及场景与现实世界中的企业到企业场景之间的关系。第 2 部分是“创建小服务程序”，共有 4 章，详述了如何创建作为 Web 应用程序基础的 4 个小服务程序（即订单小服务程序、请求小服务程序、接受小服务程序和电子邮件小服务程序）的步骤。用户先用 VisualAge for Java 和 WebSphere Studio 来开发小服务程序，然后再将这些小服务程序运用到 WebSphere 应用程序服务器上。第 3 部分有 2 章，分别讲述小服务程序的连接与故障诊断，教授用户如何通过连接小服务程序来创建交互式 Web 应用程序，与用户进行三次交互。

对开发 Web 应用程序缺少经验的普通应用程序编程人员来说，本书是他们从事现实 Web 应用程序开发的利器；对经验比较丰富的 Web 应用程序编程高手来说，本书可帮助他们进一步熟悉如何将 S / 390 用作 Web 应用程序环境的一部分。本书也是高等院校有关专业师生教学、自学参考书，及科研院所图书馆必备的馆藏读物。

本光盘内容包括本版电子书。

- 系 列 书： 21 世纪新工具软件开发指南丛书（5）（英文版）
书 名： 交互式 Web 应用程序开发指南
总 策 划： 北京希望电子出版社
文 本 著 者： 德博拉·科廷翰
责 任 编 辑： 谢建勋
CD 制 作 者： 希望多媒体创作中心
CD 测 试 者： 希望多媒体测试部
出版、发行者： 北京希望电子出版社
地 址： 北京中关村大街 26 号，100080
网址： www.bhp.com.cn E-mail: lwm@hope.com.cn
电话： 010-62562329,62541992,62637101,62637102
010-62633308,62633309（发行）
010-62613322-215（门市） 010-62629581（编辑部）
经 销： 各地新华书店、软件连锁店
排 版： 希望图书输出中心 房利萍
CD 生 产 者： 北京中新联光盘有限责任公司
文 本 印 刷 者： 北京广益印刷有限公司
开 本 / 规 格： 787×1092 毫米 16 开本 13 印张 320 千字
版 次 / 印 次： 2001 年 7 月第 1 版 2001 年 7 月第 1 次印刷
印 数： 1-2000 册
本 版 号： ISBN 7-900071-78-4/TP•77
定 价： 42.00 元（1CD，含配套书）

说明：凡我社光盘配套图书若有缺页、倒页、脱页、自然破损，本社发行部负责调换。

Part 1

In this first part, you see a description of the overall scenario, including the configuration of the server environment, an overview of the steps you take to create the Web application, and how the scenario pertains to real world business-to-business scenarios.

Contents

Part 1 Overview of the scenario	0
CHAPTER 1 INTRODUCTION	2
1.1 Scenario overview	4
1.1.1 S/390 software prerequisites	4
1.1.2 Windows NT client software prerequisites	5
1.2 Overview of the servlets	5
1.2.1 Order servlet	5
1.2.2 Bid servlet	7
1.2.3 Accept servlet	8
1.2.4 Email servlet	9
1.3 Overview of the server environment	10
1.4 Overview of the Web application	11
1.4.1 From the user's perspective	11
1.4.2 From the programmer's perspective	14
CHAPTER 2 INSTALL THE DEVELOPMENT SOFTWARE	16
2.1 Install IBM VisualAge for Java V3.02	17

2.2	Install VisualAge for Java features	18
2.3	Install IBM WebSphere Studio 3.5	18
2.4	Install a Web browser.....	19
2.5	Install the demo files.....	19

CHAPTER 3 ABOUT THE SERVER CONFIGURATION 20

3.1	About the application flow.....	21
3.1.1	Getting started.....	22
3.1.2	Retrieving customer order information	24
3.1.3	Requesting shipping options	26
3.1.4	Accepting a bid	28
3.1.5	Emailing the customer	30
3.2	About the server applications.....	31
3.2.1	About the customer and order applications.....	31
3.2.2	About the shipper applications.....	32
3.2.3	About the hub application.....	32
3.2.4	About the IMS customer application	33
3.2.5	Check database and transaction	34
3.3	About the subsystem configuration.....	35
3.3.1	About MQSeries configuration.....	35
3.3.2	About MQ/CICS Bridge configuration.....	36
3.3.3	About CICS configuration	36
3.3.4	About DB2 configuration	37
3.3.5	About WebSphere Application Server configuration.....	37
3.3.6	About IMS Connect.....	39

Part 2 Creating the servlets 42

CHAPTER 4 CREATE THE ORDER SERVLET 44

4.1	Create Java beans that access CICS and DB2.....	46
4.1.1	Create the B2BApp project.....	46
4.1.2	Create the order package.....	46
4.1.3	Create the COBOL customer record type	46
4.1.4	Create the CustomerRecord class	47
4.1.5	Create the COBOL order record type	48
4.1.6	Create the OrderRecord class.....	48
4.1.7	Create the RetrieveCustomerCommand class	49

4.1.8	Create the RetrieveOrderCommand class	54
4.1.9	Create the CICSRetrieveNavigator class	57
4.2	Test the CICS connection.....	63
4.2.1	Create a CICSNavExecute test application	63
4.2.2	Run the CICSNavExecute test	65
4.2.3	What you have proved so far	66
4.2.4	Create a version of the order package	66
4.3	Generate the order servlet and generic user interface files	67
4.3.1	Preparing the WebSphere Studio project	67
4.3.2	Generate the order servlet, HTML, and JSP files.....	69
4.4	Customize the order servlet.....	71
4.5	Publish the order servlet on the test stage	71
4.5.1	Set up Studio publishing for local testing	72
4.5.2	Publish the order files on the test stage.....	72
4.6	Test the order servlet in the WebSphere Test Environment	73
4.6.1	Set up the Servlet Engine classpath	73
4.6.2	Run the Servlet Engine	73
4.6.3	Run the servlet.....	74
4.7	Modify the order beans for publishing on the server	76
4.7.1	About the modifications.....	76
4.7.2	Change the code page	76
4.7.3	Change the command bean properties	77
4.7.4	Change the navigator properties	78
4.7.5	Export from VisualAge for Java to WebSphere	78
4.8	Publish the order servlet on the server	78
4.8.1	Modify the servlet configuration file	78
4.8.2	Modify the user interface files	79
4.8.3	Prepare to publish on the host.....	80
4.8.4	Publish the order files on the server.....	80
4.8.5	Check the WebSphere Application Server classpath.....	81
4.8.6	Provide permissions for the published files	82
4.9	Test the order servlet from a Web browser.....	83
CHAPTER 5 CREATE THE BID SERVLET		86
5.1	Create Java beans that access MQSeries.....	87
5.1.1	Create the bid package.....	88
5.1.2	Create the BidMessageType record type.....	88
5.1.3	Create the BidRequest record class.....	88

5.1.4	Create the AllBidsMessageType record type	89
5.1.5	Create the AllBidsReply record class	90
5.1.6	Create the BidRequestCommand class	90
5.1.7	Create the AllBidsReplyCommand class	94
5.1.8	Create the BidsNavigator class	98
5.2	Test the MQ connection	102
5.2.1	Create a TestBidsNav test application.....	102
5.2.2	Run TestBidsNav to test the MQ connection	104
5.2.3	Create a version of the bid package	105
5.3	Generate the bid servlet and generic user interface files.....	105
5.3.1	Prepare to generate.....	105
5.3.2	Generate the bid servlet, HTML, and JSP files.....	106
5.4	Customize the bid servlet.....	107
5.5	Publish the bid servlet on the test stage	108
5.6	Test the bid servlet in the WebSphere Test Environment.....	108
5.6.1	Start the WebSphere Test Environment.....	108
5.6.2	Run the Servlet Engine	109
5.6.3	Run the servlet.....	109
5.7	Publish the bid servlet on the server	110
5.7.1	Modify the servlet configuration file	111
5.7.2	Modify the user interface files	111
5.7.3	Publish the bid files on the server	112
5.7.4	Check the WebSphere Application Server classpath.....	113
5.7.5	Provide permissions for the published files	113
5.8	Test the bid servlet from a Web browser.....	114

CHAPTER 6 CREATE THE ACCEPT SERVLET

116

6.1	Create Java beans to exchange MQSeries messages.....	118
6.1.1	Create the accept package.....	119
6.1.2	Create the AcceptMessageType record type	119
6.1.3	Create the AcceptRequestMessage record class	120
6.1.4	Create the BothConfMessage record type.....	121
6.1.5	Create the BothConfReply record class	121
6.1.6	Create the AcceptRequestCommand class.....	122
6.1.7	Create the BothConfReplyCommand class.....	124
6.1.8	Create the AcceptNavigator class	126
6.2	Test the MQ connection	131
6.2.1	Create the TestAcceptNav class.....	131

6.2.2	Run TestAcceptNav to test the MQ connection	133
6.2.3	Review what you have demonstrated.....	133
6.2.4	Create a version the accept package.....	134
6.3	Generate the accept servlet and generic user interface files.....	134
6.3.1	Getting the accept bean files into WebSphere Studio.....	134
6.3.2	Generate the accept servlet, HTML, and JSP files.....	135
6.4	Customize the accept servlet.....	136
6.5	Publish the accept servlet on the test stage	137
6.6	Test the accept servlet in the WebSphere Test Environment.....	137
6.7	Publish the accept servlet on the server	139
6.7.1	Modify the servlet configuration file	139
6.7.2	Modify the user interface files	140
6.7.3	Publish the accept files on the server	141
6.7.4	Provide permissions for the published accept files	141
6.8	Test the accept servlet from a Web browser.....	141
6.9	Summary	143

CHAPTER 7 CREATE THE EMAIL SERVLET 144

7.1	Create the IMS beans	145
7.1.1	Create the package.....	145
7.1.2	Create a RecordType for the input message.....	145
7.1.3	Create a transaction input record bean	146
7.1.4	Create a RecordType for the output message.....	147
7.1.5	Create a transaction output record bean	148
7.1.6	Create a command class for the IMS transaction	149
7.2	Test the beans in the Websphere Test Environment.....	157
7.2.1	Create the test Java Server Page	157
7.2.2	Run the Java Server Page locally	158
7.3	Create the servlet and add email functionality	159

Part 3 Creating the Web application 162

CHAPTER 8 JOIN THE SERVLETS 164

8.1	Define a Web application in WebSphere Application Server.....	165
8.2	Provide the interactive flow	167
8.2.1	Create the opening page.....	168
8.2.2	Modify the order results page	169

8.2.3	Modify the order servlet configuration file	171
8.2.4	Modify the bid results page	172
8.2.5	Modify the bid servlet configuration file	175
8.2.6	Modify the accept results page	175
8.2.7	Modify the accept servlet configuration file	177
8.3	Modify the generated servlets to pass the data	177
8.3.1	Understanding the order servlet	178
8.3.2	Understanding the bid servlet	179
8.3.3	Modifying the accept servlet	180
8.4	Optionally, enhance the user interface	182
8.5	Considerations for developing your own Web applications	182
CHAPTER 9 TROUBLESHOOTING		184
APPENDIX A SERVER APPLICATIONS REFERENCE INFORMATION		188
A.1	Message syntax	189
A.1.1	Bid messages	189
A.1.2	Accept messages	190
A.2	Server application quick reference	192

Overview of the scenario

Introduction

This book shows you how to create a Web application that lets a user expedite the delivery of a product ordered through the Internet.

By working through the scenario described in this book, you can see:

- How to create an interactive Web application - that is, one involving multiple interactions with the user
- How to use connectors to access IBM subsystems on OS/390, including CICS, MQ-Series, DB2, and IMS
- How to save the data acquired by one servlet and reuse it in another servlet
- How the server was configured to support the application

In the scenario described here, all the server applications are running on a single S/390 processor. In a real world environment, the server applications and subsystems which support this application could be spread across several processors. In fact, some of the server applications would reside on processors belonging to a set of Business Partners, and those systems may or may not be S/390. But these are primarily deployment choices; the overall architecture is still the same.

We call the application described here, with which users interact via the Internet and which use server applications that would ultimately belong to a set of Business Partners, *an interactive business-to-business (B2B) Web application*.

This book was adapted from a tutorial provided with the *IBM WebSphere Application Development Solution (ADS) for OS/390*. For a complete list of tutorials, see <http://www.ibm.com/s390/ads>. You do not need access to an ADS system to use this book; however, if you do have access, you can actually run the completed application.

1.1 | Scenario overview

When you work through the scenario described in this book, you create a series of servlets and then you connect and deploy them as a complete Web application. You follow steps similar to those you would take to create your own servlets to access your own CICS, MQSeries, DB2, or IMS applications or transactions on OS/390. This scenario entails these major tasks:

1. Create the order servlet to access CICS and DB2, retrieving customer and order information.
2. Create the bid servlet to access MQSeries, requesting shipping options from business partners.
3. Create the accept servlet to access MQSeries, CICS, and DB2, updating order information.
4. Create the email servlet to access IMS, retrieving the customer's email address.
5. Customize the servlets to provide interactive application flow.
6. Deploy and test the Web application.

1.1.1 S/390 software prerequisites

You don't have to have a server environment to create the servlets. You could just read about the server configuration to get an idea of how you might configure your own server for your own Web application. However, to learn the most from this exercise (and to actually deploy and run the servlets), you need access to an IBM WebSphere Application Development Solution (ADS) for OS/390 Version 2, or to an S/390 with the following software installed and configured with:

- OS/390 V2R9 or higher
- CICS Transaction Server for OS/390 V1.3
- DB2 V6.1
- IMS V6.1
- MQSeries V2.1
- WebSphere Application Server, V3.02, Service Level 3
- IBM HTTP Server V5.1
- The server applications

It is possible that other levels of these software components may also work, but the application was tested with the levels listed here.

You need a userid and password on the system and you need to know its URL and TCP/IP

host name.

See also 1.3, "Overview of the server environment" and Chapter 3, "About the server configuration".

If you are not using ADS and you actually want to run the completed Web application, you will also need to install the back-end server applications on your 390 system. See Appendix A, "Server applications reference information".

1.1.2 Windows NT client software prerequisites

To create the servlets, you need a Windows NT 4.0 client (Service Pak 4 or higher), with the following software installed and configured:

- IBM VisualAge for Java V3.02
- IBM WebSphere Studio V3.5
- A recent Java-enabled Web browser, such as Netscape 4.7

Before attempting this scenario, you should have some experience using VisualAge for Java.

If you do not, you might want to go through the VisualAge for Java tutorial.

To get to the tutorial, after you install VisualAge for Java:

1. Click **Start -> Programs -> IBM VisualAge for Java for Windows -> IBM VisualAge for Java**.
2. Click **Help -> Help Home Page**.
3. In the navigation menu on the left, select **Concepts -> Visual composition -> Visual programming fundamentals**.

1.2 | Overview of the servlets

This scenario involves four servlets that are joined together to form a fictitious company Web application, described in 1.4, "Overview of the Web application". Each servlet manages some interaction with a user and with a server application.

As you create each servlet, you also create an HTML page to provide an initial user interface to it. When the user clicks a button on the page, one of the corresponding servlets gets control. That servlet processes a request and generates a reply page using a Java Server Page (JSP) file and dynamically acquired data, which is based on the user's input.

Let's look at each stand-alone servlet briefly.

1.2.1 Order servlet

The order servlet gets control when a user enters a customer number on the order input page and clicks the **get order** button, as shown in Figure 1.



Figure 1. Order servlet user input interface

The order servlet sends two requests, in the form of CICS ECI calls, to two server applications, V2CSTDB and GETORDRS, running in CICS on the server. These applications provide customer and order information from two DB2 tables. The order servlet generates a reply page, similar to the one in Figure 2.

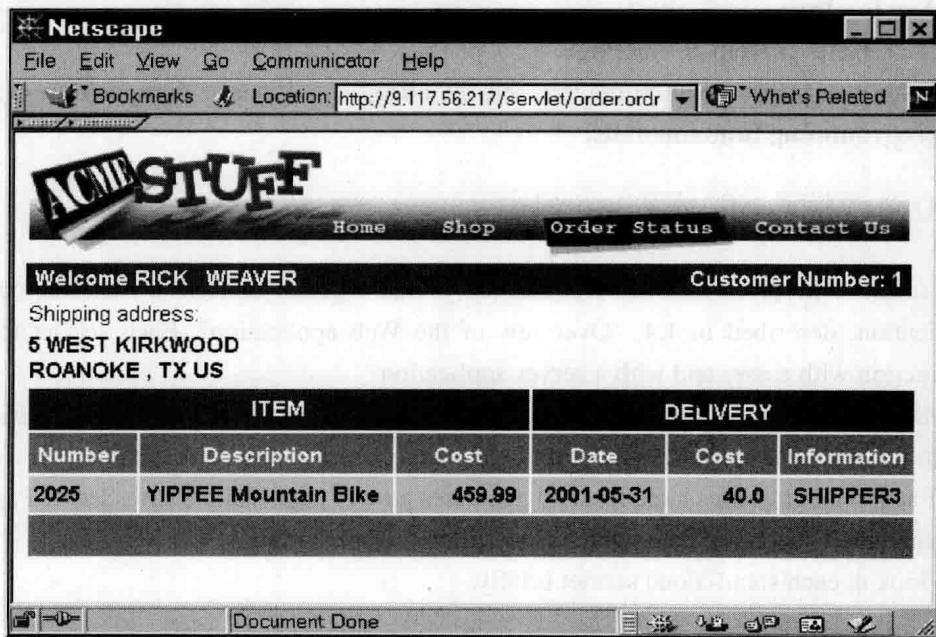


Figure 2. Order servlet results page

1.2.2 Bid servlet

The bid servlet gets control when a user enters a date on the bid input page and clicks the **expedite** button as shown in Figure 3.



Figure 3. Bid servlet input page

The bid servlet sends a request, in the form of an MQSeries message, to a server application, MQHUB, on the server. MQHUB sends additional messages to three additional applications, SHIPR1, SHIPR2, and SHIPR3. The SHIP applications provide delivery options, in the form of MQSeries messages that are consolidated and returned by MQHUB to the bid servlet. The bid servlet creates a response page, similar to Figure 4.

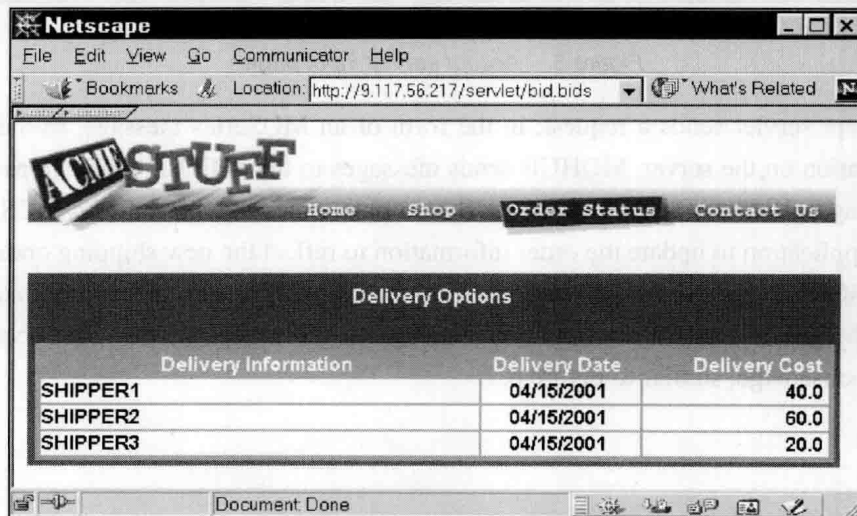


Figure 4. Bid servlet results page