

Understanding IBM SOA Foundation Suite

Learning Visually with Examples

Tinny Ng, Jane Fung, Laura Chan, Vivian Mak

The bottom half of the cover features a photograph of a construction site at sunset. The sky is a gradient of orange and yellow. In the foreground, several construction workers are silhouetted against the bright light, working on a structure. A large tower crane is the central focus, with its long jib extending across the frame. To the right, a steel framework is visible. The overall scene conveys a sense of active development and building.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

© Copyright 2010 by International Business Machines Corporation. All rights reserved.

Note to U.S. Government Users: Documentation related to restricted right. Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

IBM Press Program Managers: Tara Woodman, Ellice Uffer

Cover Design: IBM Corporation

Associate Publisher: Greg Wiegand

Marketing Manager: Kournaye Sturgeon

Publicist: Heather Fox

Acquisitions Editor: Bernard Goodwin

Managing Editor: Kristy Hart

Designer: Alan Clements

Project Editor: Andy Beaster

Copy Editor: Karen A. Gill

Indexer: Cheryl Lenser

Compositor: Jake McFarland

Proofreader: Jennifer Gallant

Manufacturing Buyer: Dan Uhrig

Published by Pearson plc

Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales

1-800-382-3419

corpsales@pearsontechgroup.com

For sales outside the U.S., please contact:

International Sales

international@pearson.com

Preface

Without change, there is no innovation.

We all want to try new things, open the door, and step into the next exciting new space. But that first step is so overwhelming that it requires a huge amount of startup time and research. It is not easy. We often give ourselves excuses and resistance not to try.

There is no better time than the present. This book is intended to make it easy and effortless for you, as a software developer who wants to ramp up your skills for Service-Oriented Architecture (SOA), to try something new: the IBM SOA Foundation Suite.

The objective of *Understanding IBM SOA Foundation Suite: Learning Visually with Examples* is to build the skills you need for today and position you for the future. This book introduces you to ten IBM products in the IBM SOA Foundation (www-01.ibm.com/software/solutions/soa/offerings.html) and gets you started with the basics quickly using a collection of hands-on tutorials. The target audience is assumed to have little or no skill for these products.

Each chapter has a product overview that talks about the significance of the product and some of the basic concepts related to the product space. Every chapter has a mini scenario that you will follow in the hands-on tutorial. These scenarios are designed to guide you through the basic navigations and usage of the products. Furthermore, the tutorials are accompanied by videos that allow you to watch and learn if you are under a deadline and do not have enough time to do the tutorials or set up the products.

IBM SOA Foundation is scalable. Companies who are interested in SOA can choose to begin with certain focus areas and progress through SOA gradually as requirements come. Being able to cover the fundamental concepts of ten different IBM SOA products in one book, *Understanding IBM SOA Foundation Suite: Learning Visually with Examples*, serves as a good reference for you. You, as a software developer for the company, can always come back to this book

anytime to refresh your skills for various IBM SOA products when the company is ready to pull in different subsets. This book gives you a head start and positions you well in your company and in the job market. Being able to broaden your foundation across a spectrum of products will be an invaluable experience.

What Will Be Covered in the Tutorials?

This book contains eight chapters that represent a wide selection of products across the IBM SOA Foundation. The products can roughly be categorized into the following:

- Service Design
- Service Creation
- Service Governance
- Service Integration
- Service Connectivity
- Collaboration
- Service Security

Chapter 1: Introduction

This chapter expands above categories with more details and gives a brief overview of the IBM SOA Foundation.

Chapter 2: Service Design with IBM Rational Software Architect

The exercises in this chapter step you through how to design your service using Unified Modeling Language (UML) diagrams, share your design with peers as Hypertext Markup Language (HTML) files, and transform the UML design to and from Java using IBM Rational Software Architect.

- Tutorial 2.1—Use a UML Model to Capture a Service Design
- Tutorial 2.2—Capture the Use Cases for a Service Using a Use Case Diagram
- Tutorial 2.3—Design the Blueprint for a Service Using a Class Diagram
- Tutorial 2.4—Detail the Flow of a Service Using a Sequence Diagram
- Tutorial 2.5—Share the Service Design with Others
- Tutorial 2.6—Transform the Service Design to Implementation with Round-Trip Engineering

Chapter 3: Service Creation with IBM Rational Application Developer and IBM WebSphere Application Server

The tutorials that are featured in this chapter are separated into two parts: service creation in IBM Rational Application Developer and service deployment in IBM WebSphere Application Server. In IBM Rational Application Developer, you will create and invoke a Web service and access a database using Java Persistence API (JPA). In IBM WebSphere Application Server, you will deploy and configure the application after development is completed.

- Tutorial 3.1—Create, Deploy, and Test a Web Service
- Tutorial 3.2—Create a Database Table
- Tutorial 3.3—Invoke a Web Service and Persist the Data Using Java Persistence API
- Tutorial 3.4—Deploy an Application into a WebSphere Application Server

Chapter 4: Service Governance with IBM WebSphere Service Registry and Repository

The hands-on exercise for this chapter tells you a story about “A Day with WSRR,” where you will explore many of the capabilities in IBM WebSphere Service Registry and Repository. The story has three users: an administrator, a service developer, and an application developer. As an administrator, you will set up the registry and repository with business model templates and a classification system. As a service developer, you will publish the developed service artifacts into the registry and repository, update them, and move them through the governance life cycle. As an application developer, you will perform a search for the right service and reuse it in your application.

- Tutorial 4.1—Set Up the Registry and Repository as an Administrator
- Tutorial 4.2—Publish a New Service as a Service Developer
- Tutorial 4.3—Reuse Services as an Application Developer
- Tutorial 4.4—Update Existing Services as a Service Developer

Chapter 5: Service Integration with IBM WebSphere Integration Developer and IBM WebSphere Process Server

In this chapter you will use existing services to create a new business process by assembling them. The scenario features an ordering process of a car manufacturing company. You will create a simple business process as well as other components in IBM WebSphere Integration Developer and then deploy them to IBM WebSphere Process Server.

- Tutorial 5.1—Create a Business Process
- Tutorial 5.2—Assemble and Execute the Module
- Tutorial 5.3—Deploy to a WebSphere Process Server

Chapter 6: Service Connectivity with IBM WebSphere Message Broker

This exercise lets you practice using IBM WebSphere Message Broker to implement a service that handles a library book search request. You will be writing Embedded Structured Query Language (ESQL) to implement a function that queries a library database for the book. You will also be using the Mapping node that is supplied with the product to create a search response message from the search request message.

- Tutorial 6.1—Configure Message Broker Toolkit with Predefined Databases and Runtime Artifacts
- Tutorial 6.2—Create the Message Flow and Message Set for the Library Book Search Service
- Tutorial 6.3—Deploy and Test the Library Book Search Service

Chapter 7: Collaboration with IBM WebSphere Portlet Factory and IBM WebSphere Portal

The tutorial in this chapter introduces you to IBM WebSphere Portlet Factory and IBM WebSphere Portal. It provides step-by-step instructions to give you a complete end-to-end experience from development to production. You will develop and test two portlets in a development environment using IBM WebSphere Portlet Factory, and then you will deploy/manage the portlets in a production environment using IBM WebSphere Portal. These portlets use the service provider and service consumer design.

- Tutorial 7.1—Create and Test a Simple Portlet
- Tutorial 7.2—Create and Test a Portlet That Accesses a Database
- Tutorial 7.3—Deploy a Portlet

Chapter 8: Service Security with IBM Tivoli Federated Identity Manager

The tutorials designed for this chapter illustrate both the token generator and token consumer functionalities of Tivoli Federated Identity Manager (TFIM). You will enable security for a service provider so that requesters must provide a valid Username security token for access. Then you will configure a service requester and enable its security so that it will generate a valid security token for accessing the service provider.

- Tutorial 8.1—Enable Security for a Service Provider
- Tutorial 8.2—Enable Security for a Service Requester
- Tutorial 8.3—Test the Service

What Is Included in the CD-ROM?

In the CD included in this book, you will find completed solutions for all the exercises discussed in this book. All of the tutorials have been recorded as videos so you can learn the operational concepts of the products.

The CD is organized as follows:

```
readme.html
chapter x
  readme.html
  /setup
  /tutorial x.y
    /solution
    /tutorial files
    /video
```

1. There is a readme file that can be opened in any browser. It provides a table of links to the readme file of each chapter.
2. Each chapter folder has the following structure:
 - a. A readme file that can be opened in any browser. It gives you a brief introduction to what to expect in the folder underneath. It also provides a table of links to the videos that can be clicked and run directly.
 - b. A setup folder that contains the files, if any, that are needed to set up the tutorials.
 - c. A set of tutorial $x.y$ folders. Each has:
 - i. A solution folder that contains a completed solution, if any, for the subject tutorial.
 - ii. A tutorial files folder that contains the files, if any, that are needed when exercising the subject tutorial.
 - iii. A video folder that contains the video files for the subject tutorial allowing readers to watch the step-by-step instructions. Open the HTML file in any browser to watch the video.

Collaboration with IBM WebSphere Portlet Factory and IBM WebSphere Portal

Product Overview

This chapter introduces you to two powerful software products that are suited for implementing a portal. The IBM WebSphere Portlet Factory provides a development environment for creating the applications for a portal. The IBM WebSphere Portal presents the tools for building and managing a portal and offers the hosting/runtime environment for a portal.

What Is a Portal?

A *portal* is a website that offers users a single point of access to a company's resources and services, requiring only one login to the website. It gives users personalized interaction with the resources and services, provides role-based access, and allows users to choose their experience and customize their own views of the website.

Users include employees, business partners, suppliers, and customers. "Company's resources and services" refer to applications, content, business processes, and people.

As an example, the following are some characteristics of a portal for a computer manufacturer.

- Using one login, employees can access applications from all departments, including human resources and technical support.
- An example of a personalized application is giving a different discount rate to customers based on their profiles. An example of personalized content is showing different news to business partners than suppliers.
- An example of role-based access is only allowing employees who are managers to see and to use the approved expense application.

- Users can add applications to their views, arrange applications, and customize the appearance of applications (for example, color).

Basic Portal Terminology

Concepts	Definitions
Portlet	To an end user, a portlet is a window on a portal website that displays information or provides a function. To a developer, it is an application.
Portlet application	A portlet application is a collection of related portlets that share the same resources. For example, these are images, properties, files, and classes.
Portal page	A portal page contains one or more portlets.
Portal website	A portal website is built with portal pages.

IBM WebSphere Portlet Factory

IBM WebSphere Portlet Factory provides a rapid development tool, WebSphere Portlet Factory Designer, to simplify and accelerate the creation of portlet applications. WebSphere Portlet Factory Designer, hereafter called Portlet Factory Designer, is a graphical tool that is a plug-in to Eclipse-based integrated development environments (IDEs).

With Portlet Factory Designer, you create projects, under which you develop models using builders and generate the resulting portlet applications from those models. Typically, no coding is required.

It is worth mentioning that IBM WebSphere Portlet Factory can be used as a development tool for creating Java 2 Platform, Enterprise Edition (J2EE) Web applications, although Web applications are not the focus of this chapter. The benefit is that you would create one model and use the same model to generate both a Web and a portlet application.

IBM WebSphere Portlet Factory Basic Concepts and Definitions

- Builder
 - A WebSphere Portlet Factory application is made up of builders. A *builder* is a software automation component that generates necessary application code. It has a simple wizard-like user interface for a developer to provide inputs. Based on the inputs, the builder generates code, including JavaServer Pages (JSPs), Java classes, and Extensible Markup Language (XML) documents. Each builder offers the function of an application design pattern, such as displaying a list of data from a backend data store. The IBM WebSphere Portlet Factory product ships with more than 160 ready-to-use builders at the time of writing this chapter and supports the creation of new builders. Builders are implemented using Java and XML.

- Model
 - Builders are assembled into models. A *model* is made up of a number of builder calls. Web and portlet applications are generated from models. A well-designed application uses a model to provide data (a service provider model) and a different model to display the data (a service consumer/presentation model).

IBM WebSphere Portal

IBM WebSphere Portal has a complete set of portal capabilities to deliver business-to-business (B2B), business-to-consumer (B2C), and business-to-employee (B2E) portals. The capabilities include framework services, integration services, content services, and collaboration services.

Portals built using IBM WebSphere Portal are secure, personalized, and role-based with a unified user experience. These portals connect people, applications, business processes, and content (document and other types) so users can work productively and be more satisfied.

IBM WebSphere Portal software helps make it easy to create and maintain a portal. It is also faster. With a rich, responsive user interface based on Web 2.0 features, companies can provide self-service, collaboration, business intelligence dashboards, and more for their employees, partners, suppliers, and customers. As a result, they can respond quickly to business opportunities and drive business success.

IBM WebSphere Portal Basic Concepts and Definitions

- Portal server runtime
 - The portal server runtime is the execution environment for the portlets. It is also called the *portlet container*. It is a J2EE application that runs on the IBM WebSphere Application Server.
- Portlet versus servlet
 - Portlets can be administered while the Portal Server is running. For example, you can install/remove portlet applications and create/delete portlets and portlet settings. Portlets are more dynamic than servlets. Portlets may not send errors directly to browsers, forward requests, or write arbitrary markup to the output stream.
- Portal page layout
 - The *portal page layout* defines the number of content areas within the page and the portlets displayed within each content area.
- Themes
 - *Themes* represent the overall look and feel of the portal, including colors, images, and fonts.
- Skins
 - *Skin* refers to the appearance of the area surrounding an individual portlet. Each portlet can have its own skin.

IBM WebSphere Portal Key Capabilities

IBM WebSphere Portal has a broad set of capabilities. It provides you with a runtime server, services, tools, and many other functions. It is not the intention to list all the capabilities here, but let's look at a few key ones.

- Composite application and mashup framework
 - A *composite application* is a set of related and integrated services that support a business process. Users select components and logic from two or more applications to build a completely new application. You can derive that IBM WebSphere Portal itself is a security-rich composite application/view that assembles and delivers services in the form of portlets in the context of a business process.
- Web 2.0 support
 - Live Text presents “one-click” access to relevant supporting information that “pops up” on the page while executing a business process.
 - REST (Representational State Transfer) services further open the portal platform to composite mashup applications with services feeds from other Web applications.
 - Client Side Aggregation reduces server-side processing, which dramatically improves end-user performance.
- Single sign-on (SSO)
 - The basis for seamless, secure digital-identity-based access to multiple enterprise applications, systems, and networks.
- Role-based access
 - Offers advanced control over access to information, content, and applications based on users' roles and responsibilities in the organization.
- Personalization
 - Allows a portal or website to choose which content should appear for a particular user.
- Content management
 - Gives business users the ability to create and manage portal content without information technology (IT) intervention or support.
- Collaboration
 - Enables users to share information and collaborate within the context of an application in which they are working. Provides access to collaboration tools including instant messaging, Web conferencing, and team workspaces.
- Search
 - Functions include advanced search, search facets, search services, search scopes, search collections, adding custom links, summarizer, and search using different languages that the product supports. There are search services that come with IBM Web-

Sphere Portal, and you can add one or more. By adding custom links, users can do direct search using popular search engines.

How Do They Support SOA?

A *Service-Oriented Architecture* (SOA) is simply an IT architectural style that allows applications to be built to support today's business needs of flexibility and integration. *Flexibility* enables a business to change and adapt. *Integration* emphasizes the communication and interaction of business processes from end-to-end, both internally and externally with customers and suppliers.

IBM WebSphere Portlet Factory is part of the IBM SOA Foundation and supports the “assemble phase” of the SOA life cycle. Developing a SOA application basically involves creating a service provider model and a service consumer model. Additional SOA features include disconnected support via stub service models, automatic service testing, simple service documentation, dynamic service mapping, and service interface support.

IBM WebSphere Portal is part of the IBM SOA Foundation and supports the “deploy phase” of the SOA life cycle. The deploy phase includes a combination of creating the hosting environment for your applications and actually deploying those applications. IBM WebSphere Portal is a hosting environment for the user interaction logic of your SOA application. IBM WebSphere Portal gives you the user interface to SOA. Additional SOA features include accessing Web services from portlets, Web services for remote portlet (WSRP), integrating business processes, and composite applications.

By connecting/integrating a company's resources including people, applications, business processes, and content, IBM WebSphere Portlet Factory and IBM WebSphere Portal build portals that provide complete interaction and collaboration services.

Tutorial Overview

The tutorial in this chapter introduces you to IBM WebSphere Portlet Factory and IBM WebSphere Portal. It provides step-by-step instructions to give you a complete end-to-end experience from development to production. You will develop and test two portlets in a development environment using IBM WebSphere Portlet Factory, and then you will deploy/manage the portlets in a production environment using IBM WebSphere Portal.

End-to-End on Developing, Testing, and Deploying of a Portlet

The tutorial is based on a scenario about a company called PetCompleteServices building their portal Web site. PetCompleteServices is all about pets. It provides information and services for pets, including pet travel insurance, pet travel carrier, pet sitting, and lots more. Victoria is responsible for architecture and technology at PetCompleteServices. In her SOA plan for the company, the first step is to start with a portal. It is in PetCompleteServices' long-term plan to integrate with airline companies to provide travel services for pets. Claire is from the airline company HelloWorldAir and is always looking to work with new partners to provide first of a kind

services for her customers to stay on top of the competition. Victoria and Claire have started working on the integration proposal.

The tutorial is divided into three parts (Tutorials 7.1, 7.2, and 7.3):

- In Tutorial 7.1, you will develop and test a simple portlet, called `PetTravelInfo`, which is expected to have the look and feel shown in Figure 7.1. This portlet displays static text.



Figure 7.1 Look and feel of `PetTravelInfo`

The focus of this part of the tutorial is the basics of IBM WebSphere Portlet Factory. It covers projects, models, builders, portlets, testing of a model, and building the WAR file for deploying in production. If your intention is to have a hands-on end-to-end experience from development to production of a portlet, you can choose to follow only Tutorials 7.1 and 7.3.

- In Tutorial 7.2, you will develop and test a portlet, called `ListPetTravelCompany`, which is expected to have the look and feel shown in Figure 7.2. This portlet retrieves information from a database and displays the results in a simple table.

The screenshot shows a portlet titled 'ListPetTravelCompany'. It displays a table with the following data:

ID	COMPANY	DESC
1	AA	Pet travel insurance.
2	BB	Pet carrier for dogs.
3	CC	Pet friendly accomodation.
4	DD	Pet sitting while you travel.

Figure 7.2 Look and feel of `ListPetTravelCompany`

The focus of this part of the tutorial is a well-designed SOA WebSphere Portlet Factory application. It covers service provider and service consumer models.

- In Tutorial 7.3, you will put `PetTravelInfo` and `ListPetTravelCompany` in a production environment. The result is shown in Figure 7.3. You will install the portlets in IBM WebSphere Portal and then add them to portal pages. Finally, you will grant permission to a new user to use the portlets.

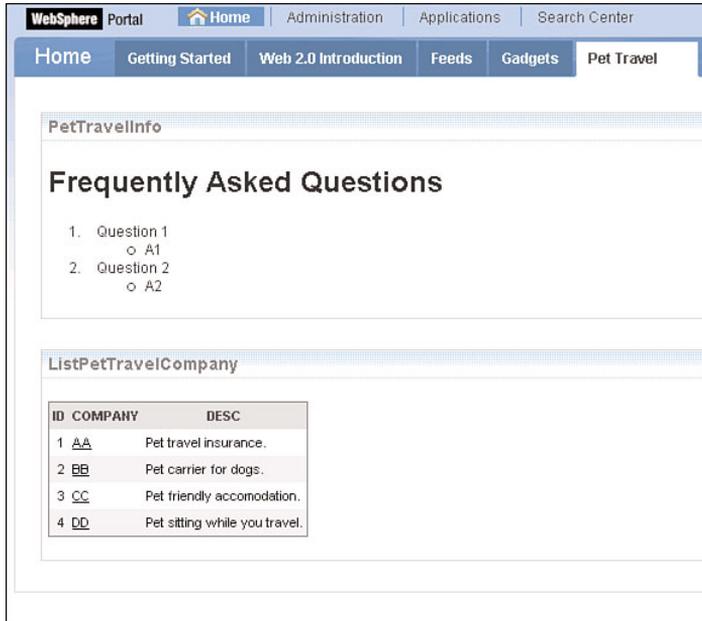


Figure 7.3 PetTravelInfo and ListPetTravelCompany in production

The focus of this part of the tutorial is on using tools to build a portal, including installing portlet application, creating portal pages, adding portlets to portal pages, and editing portal page layout; using tools to manage a portal, including giving access to users on new portlets; and single login to a portal website rather than to individual application.

Tutorial outline:

- Tutorial 7.1: Create and Test a Simple Portlet
 - Step 1: Create a project in IBM WebSphere Portlet Factory
 - Step 2: Create and test a simple portlet
- Tutorial 7.2: Create and Test a Portlet That Accesses a Database
 - Step 1: Create a service provider model
 - Step 2: Create a service consumer model
- Tutorial 7.3: Deploy a Portlet
 - Step 1: Install a portlet in IBM WebSphere Portal
 - Step 2: Add a portlet to a WebSphere portal page
 - Step 3: Access a portlet as a new user

- Step 4: Set access permissions for a portlet

System Requirements

The tutorial has been developed with the following products and environment.

- IBM WebSphere Portlet Factory version 6.1
 - An application server is needed for testing the model. The WebSphere Application Server instance `WebSphere_Portal` in the WebSphere Portal installation can be used.
- IBM WebSphere Portal version 6.1
- IBM DB2 Enterprise Server Edition version 9.1.4
 - This product is required for Tutorial 7.2. Your WebSphere Portal installation should be configured with DB2.

What Is Included in the CD-ROM?

In the CD included in this book, you will find the WebSphere Portlet Factory application and the tutorial recorded as videos.

1. **chapter 7/tutorial 7.2/solution/PetTravel.zip**—A WebSphere Portlet Factory archive file that contains the completed application for your reference. To import it to WebSphere Portlet Factory Designer, click File, Import from the menu bar. Then expand Other and click WebSphere Portlet Factory Archive. Click on Next and follow the wizard's instructions.
2. **chapter 7/tutorial 7.x/video**—Contains the video files for all the tutorials discussed in this chapter. Open the Hypertext Markup Language (HTML) file in any browser to watch the video.

Tutorial Setup

To prepare the environment for the tutorial, you will create a deployment configuration in WebSphere Portlet Factory, create a database and a table in DB2, and create a datasource in WebSphere Application Server.

WebSphere Portlet Factory—Create a Deployment Configuration

A *deployment configuration* is a set of properties that define your deployment information. An application server is needed for testing the models in the tutorial and you will create a deployment configuration for the application server instance. Your WebSphere Portal installation has two WebSphere Applications Server instances: `server1` and `WebSphere_Portal`. In this section, you will create a deployment configuration for the WebSphere Application Server instance `WebSphere_Portal`.

1. Start WebSphere Portlet Factory Designer.
 - a. Click Start, All Programs, IBM WebSphere, Portlet Factory, Designer.
 - b. Click OK to accept the default workspace folder as shown in Figure 7.4.



Figure 7.4 Select a workspace

2. Create a deployment configuration.
 - a. Click Window, Preferences as shown in Figure 7.5.

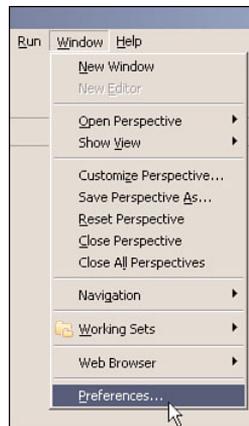


Figure 7.5 Set preferences

- b. Expand the WebSphere Portlet Factory Designer section and click Deployment as shown in Figure 7.6.



Figure 7.6 Select deployment preferences

- c. Click Add to create the deployment environment for deploying your projects as shown in Figure 7.7.

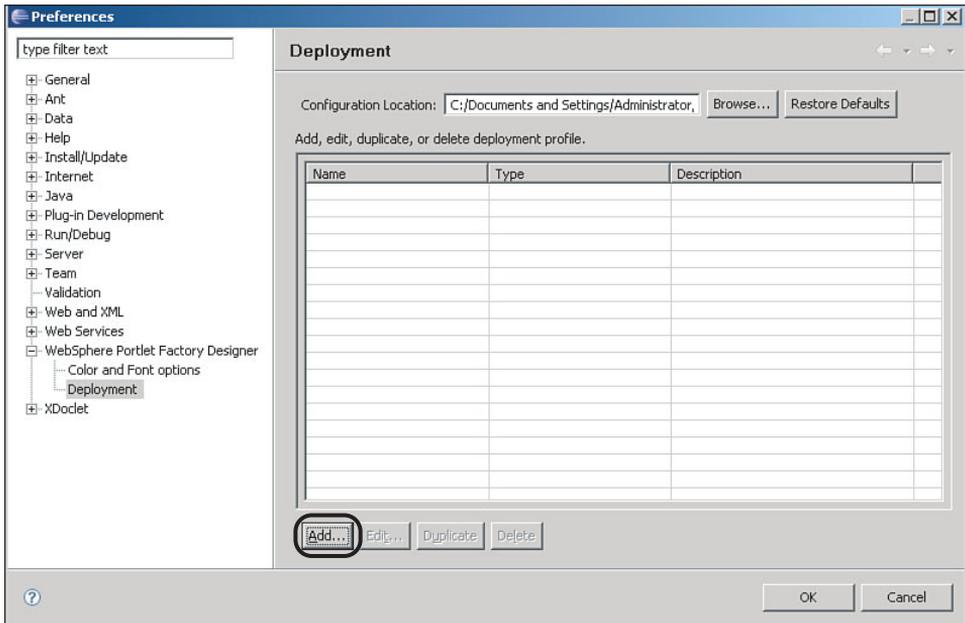


Figure 7.7 Add a deployment configuration

- d. Table 7.1 shows you how to provide inputs for the New Deployment Configuration dialog. Click OK when you are finished with the dialog as shown in Figure 7.8. Make sure the WAS server `WebSphere_Portal` has been started.

Table 7.1 Input values for Deployment Configuration dialog

Parameters	Values
Configuration Name	Enter <code>WAS61</code> or provide a different name of your choice.
Description	Enter <code>WAS 6.1</code> deployment configuration or provide your own description.
Server Type	Select WebSphere Application Server 6.x from the drop-down.
Installed Applications Dir	Enter your WebSphere Portal installed directory. (Example: <code>C:\IBM\WebSphere\wp_profile\installedApps\portalhost</code>).
Server Host	Enter your WebSphere Portal host name. (Example: <code>portalhost.ibm.com</code>).
Server Port	10040.
Specify Deployment Credentials	Select this check box.
WAS Server for deployment	Select <code>WebSphere_Portal</code> from the drop-down.
Admin User	Enter your WebSphere Portal Administrator user ID.
Admin Password	Enter your WebSphere Portal Administrator password.

- e. A message indicating that the test to connect to the WAS server was successful is shown as in Figure 7.9. Click OK to close the window.
- f. The new deployment configuration has been created successfully as shown in Figure 7.10. Click OK to close the Preferences window.

In this setup step, you have created a deployment configuration for deploying your projects.

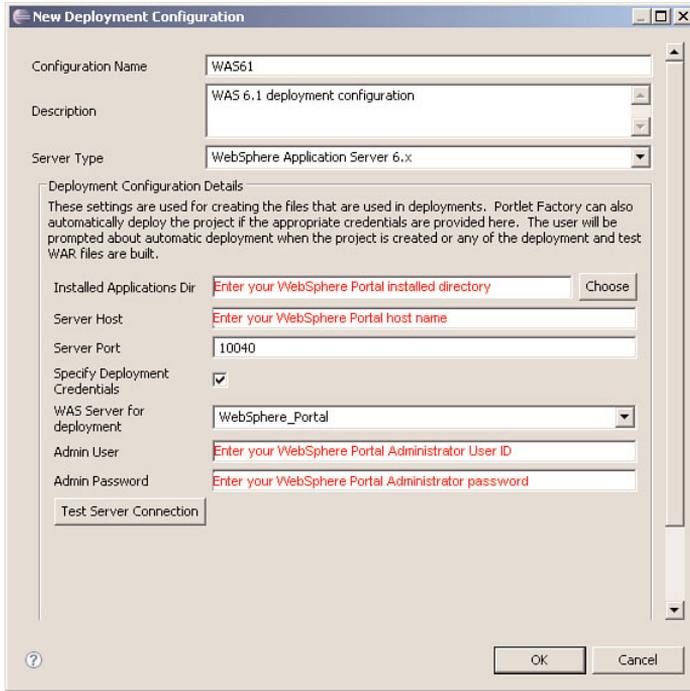


Figure 7.8 New Deployment Configuration dialog



Figure 7.9 Test connection message



Figure 7.11 Specify Control Center view

- 2. Create a database using the Create Database Wizard.
 - a. Right-click All Databases and click Create Database, Standard as shown in Figure 7.12.

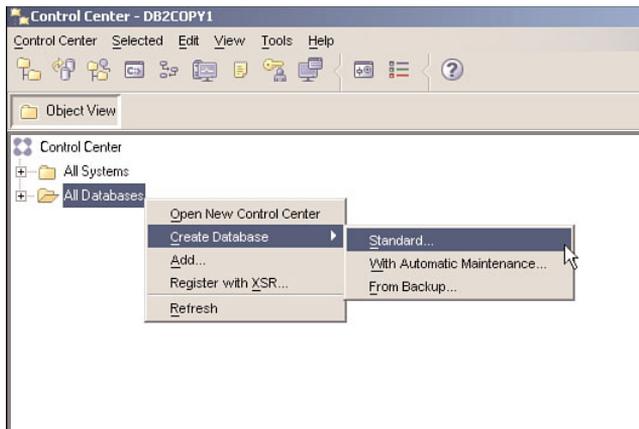


Figure 7.12 Invoke Create Database Wizard

- b. Enter `PetDB` as the database name, and keep all other defaults as shown in Figure 7.13. Click Finish.

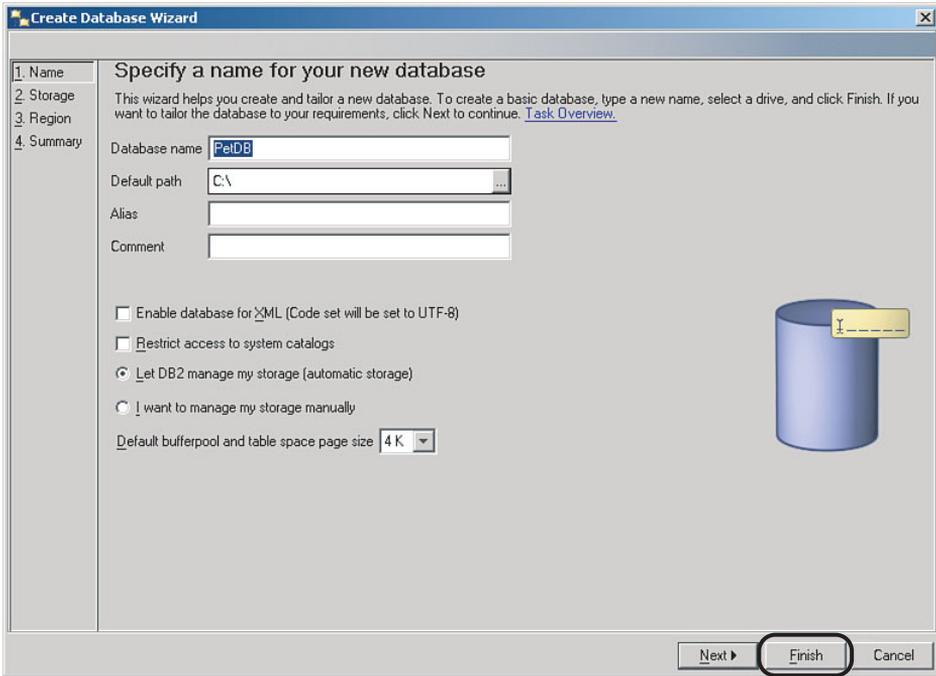


Figure 7.13 Specify database name

- 3. Create a database table using the Create Table Wizard.
 - a. After database `PetDB` is created successfully, you are returned to the Object view. You should be able to see `PetDB` by expanding All Databases. To invoke the Create Table Wizard, expand `PetDB`. Then right-click Tables and click Create as shown in Figure 7.14.

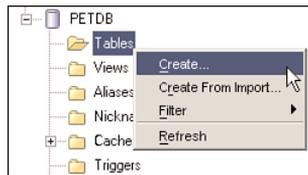


Figure 7.14 Invoke Create Table Wizard

- b. Enter `PetTravelCompany` as the table name as shown in Figure 7.15. Click Next.

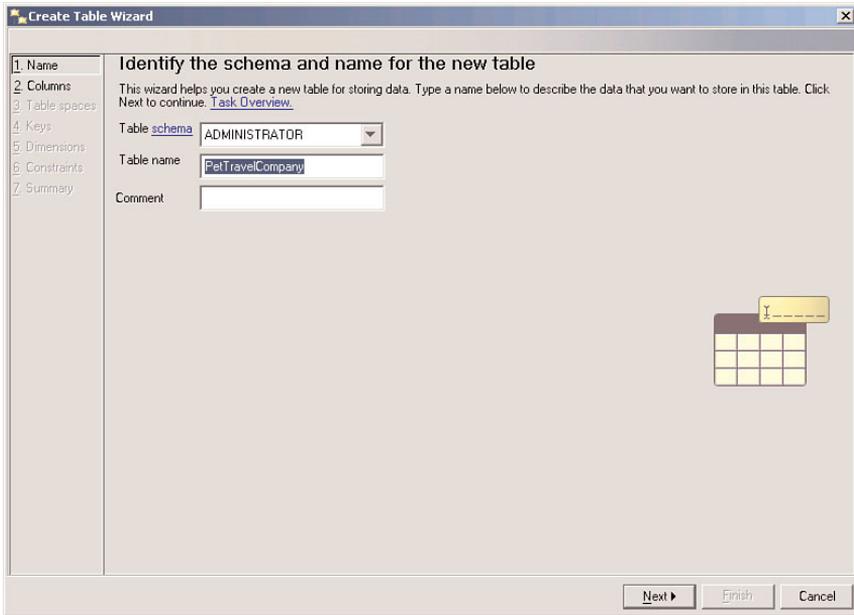


Figure 7.15 Specify table name

- c. Click Add to add a column to the PetTravelCompany table as shown in Figure 7.16.

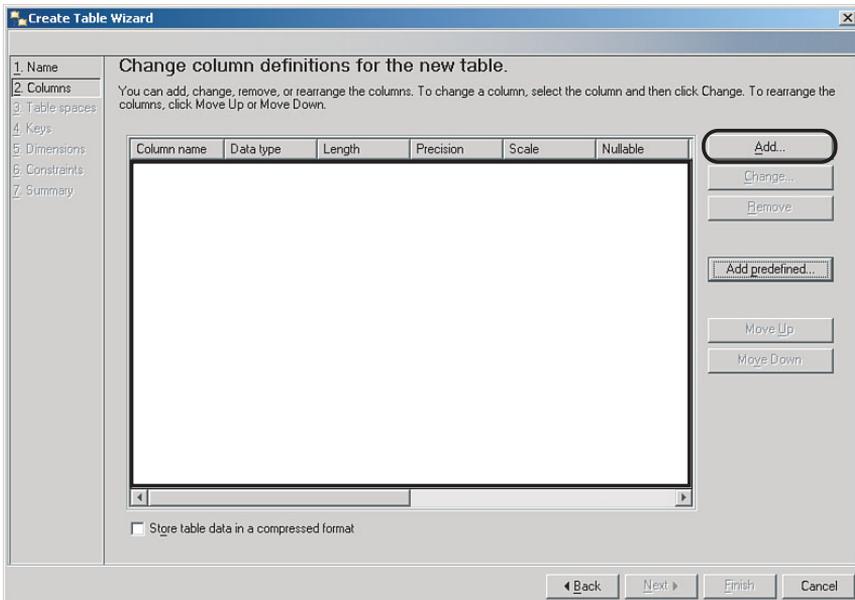


Figure 7.16 Add a column

- d. Enter `ID` as the column name and select `INTEGER` from the drop-down for data type as shown in Figure 7.17. Click OK.

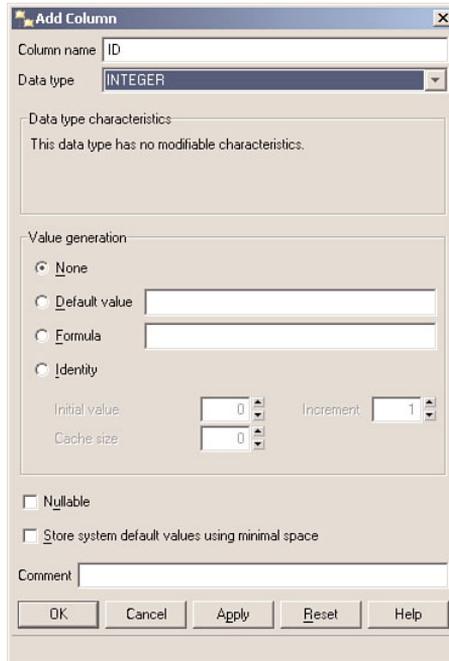


Figure 7.17 Add Column dialog

- e. Use the Add button to add two more columns using the Table 7.2 as inputs.

Table 7.2 Inputs values for add column dialog

Column Name	Data Type	Length
Company	VARCHAR	50
Desc	VARCHAR	100

- f. `PetTravelCompany` has three columns added as shown in Figure 7.18. Click Finish.
- g. A message indicating that the table creation was successful is shown as in Figure 7.19. Click Close.

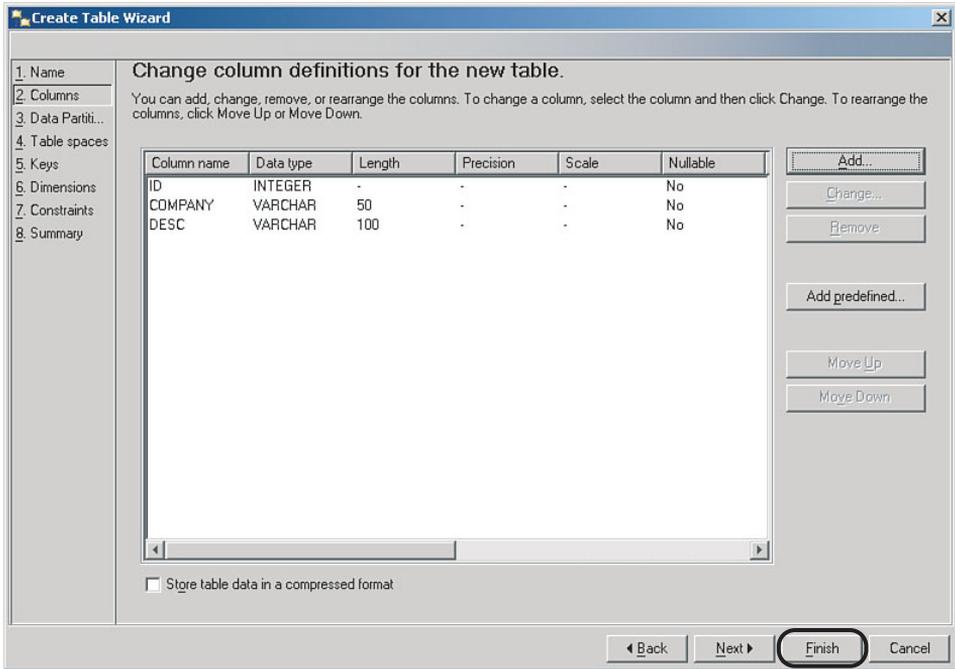


Figure 7.18 Column definition

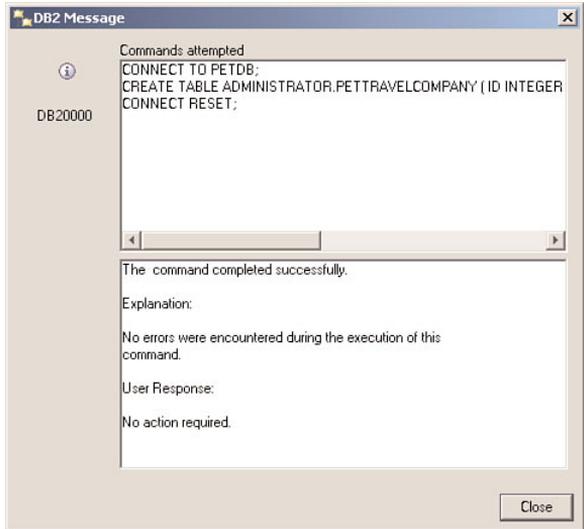


Figure 7.19 Table successfully created

- h. To display the list of tables in the Contents pane, right-click Tables and click Refresh as shown in Figure 7.20. Do you see your newly created table `PetTravelCompany` in the Contents pane on the right side?

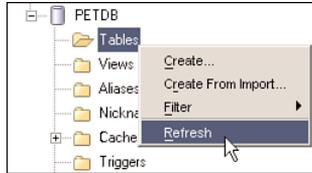


Figure 7.20 Refresh the list of tables

4. Add data to a database table.

- a. To add data to the table `PetTravelCompany`, right-click `PETTRAVELCOMPANY` and click Open. as shown in Figure 7.21



Figure 7.21 Open a table

- b. Click Add Row to add a row to the table as shown in Figure 7.22. You will be doing this action four times to add four rows.

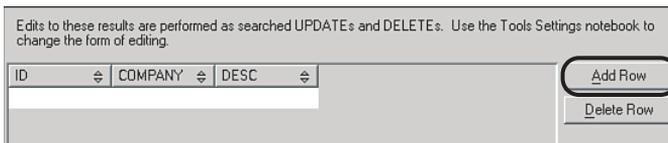


Figure 7.22 Add a row

- c. Use table 7.3 as input when adding the four rows to the table.

Table 7.3 Input values for Add Rows

ID	COMPANY	DESC
1	AA	Pet travel insurance
2	BB	Pet carrier for dogs
3	CC	Pet friendly accommodation
4	DD	Pet sitting while you travel

- d. When all the rows have been added to the table, click Commit and then Close as shown in Figure 7.23.

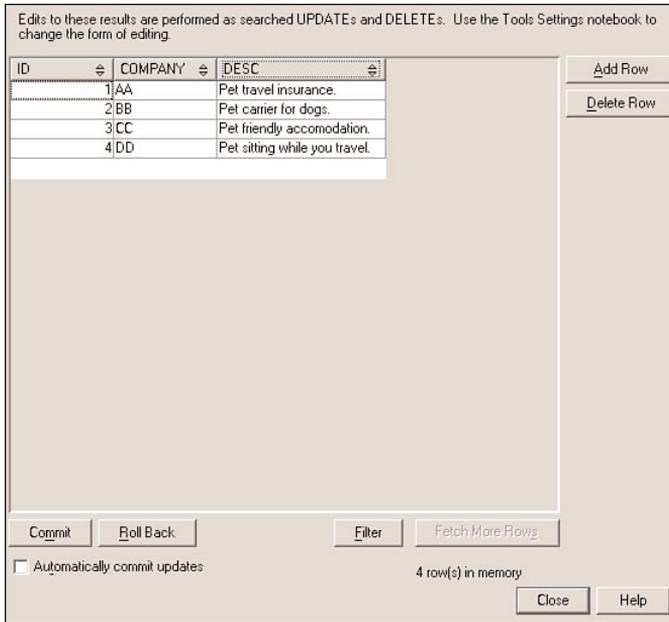


Figure 7.23 Table content

In this setup step, you have created a database and a table for creating a portlet that accesses a database.

WebSphere Application Server—Create a Datasource

This is a setup step required by Tutorial 7.2.

1. Log in to WebSphere Application Server administration console.
 - a. Start the WebSphere Application Server administration console by opening a browser and entering the uniform resource locator (URL) `http://<hostname>:<port_number>/ibm/console`, where `<hostname>` is the fully qualified host name of

the machine that is running your WebSphere Portal and `<port_number>` is the administration console port number.

For example, enter `http://portalhost.ibm.com:10027/ibm/console`

- b. Log in using your administrator user ID and password.
2. Select the JDBC (Java Database Connectivity) provider.
- a. Expand Resources, JDBC and click JDBC Providers as shown in Figure 7.24.
 - b. Click `wpdbJDBC_db2` as shown in Figure 7.25.



Figure 7.24 Go to the list of JDBC providers



Figure 7.25 Select the JDBC provider for the portal

- 3. Create a datasource.
 - a. Under Additional Properties, click Data Sources as shown in Figure 7.26.

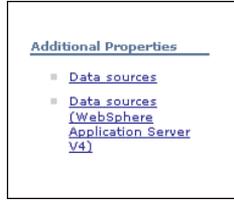


Figure 7.26 Go to the list of datasources

- b. To invoke the Create a Data Source Wizard, click New as shown in Figure 7.27. You will be creating a datasource for your database `PetDB`.

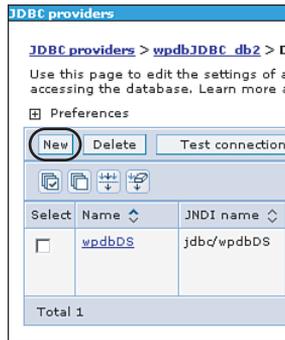


Figure 7.27 Invoke the Create a New Data Source Wizard

- c. In Step 1 of the Create a Data Source Wizard as shown in Figure 7.28, enter `PetDB` for the datasource name, `jdbc/PetDB` for the JNDI name, and select `wpdbDSJAASAuth` for the authentication alias. Click Next.
 - d. In Step 2 of the Create a Data Source Wizard as shown in Figure 7.29, enter `PetDB` for the database name and your WebSphere Portal host name for the server name. An example of a host name is `portalhost.ibm.com`. Click Next.

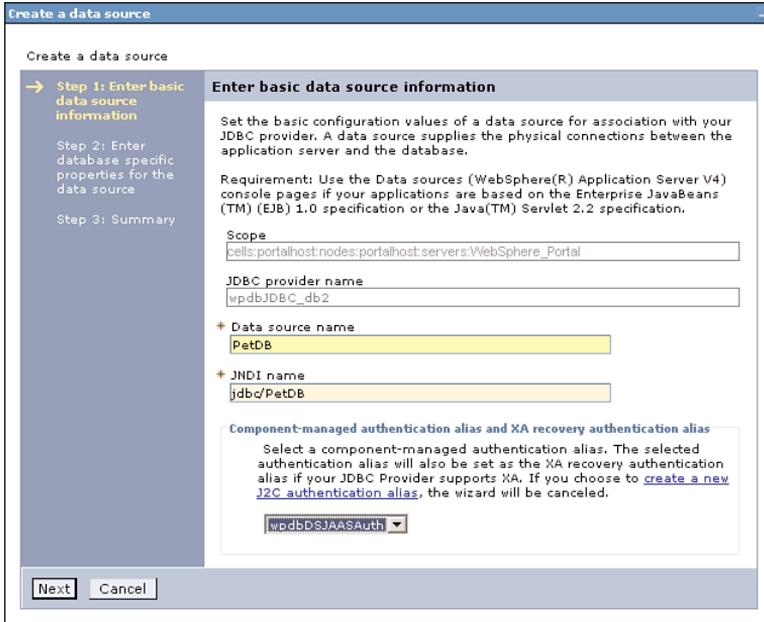


Figure 7.28 Step 1 of the Create a Data Source Wizard

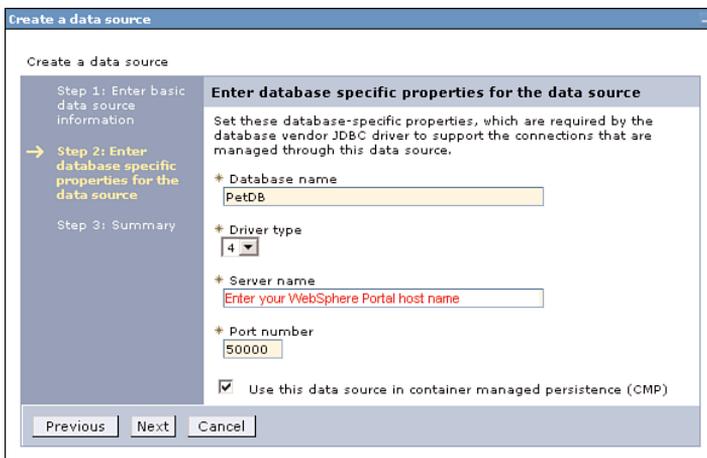


Figure 7.29 Step 2 of the Create a Data Source Wizard

- e. In Step 3 of the Create a Data Source Wizard as shown in Figure 7.30, simply review the information. Click Finish.
- f. Click the Save Directly to the Master Configuration link as shown in Figure 7.31.

In this setup step, you created a datasource for database `PetDB`.

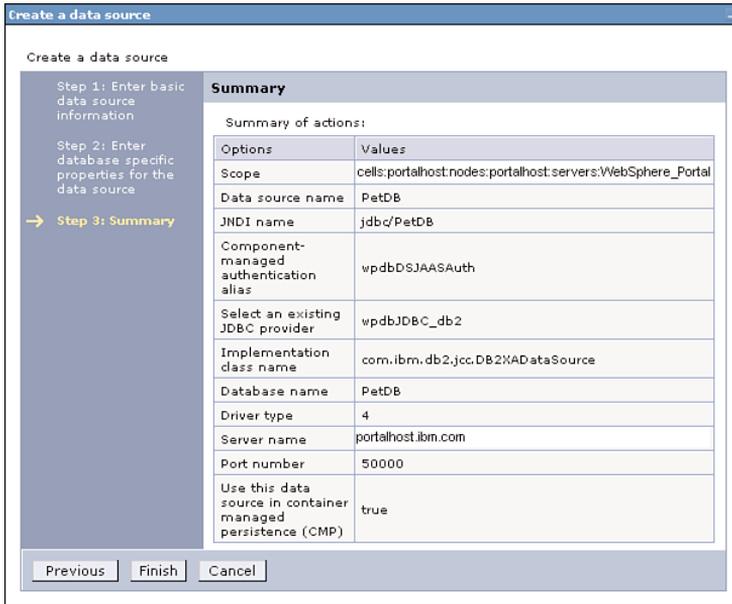


Figure 7.30 Step 3 of the Create a Data Source Wizard

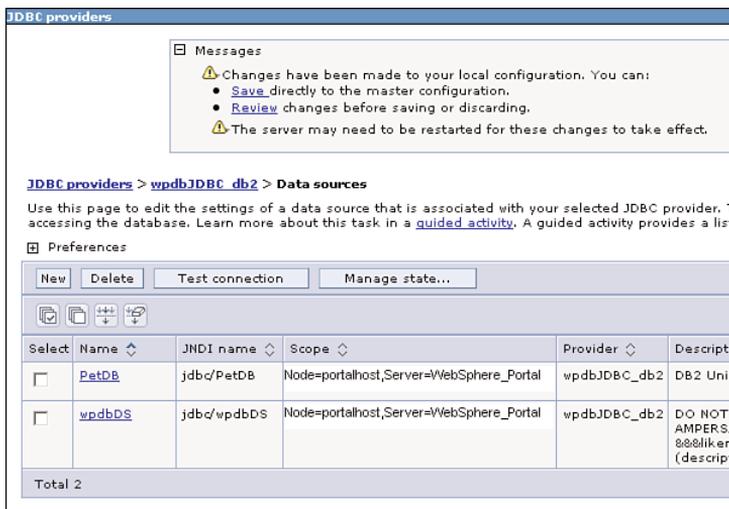


Figure 7.31 Save to master configuration

Tutorial 7.1: Create and Test a Simple Portlet

In this tutorial, you will create a project for all the models that will be built in Tutorials 7.1 and 7.2. You will then create a simple portlet.

Step 1: Create a Project in IBM WebSphere Portlet Factory

In this step, you will create a project in Portlet Factory Designer.

1. Launch the Create Portlet Factory Project Wizard to create a WebSphere Portlet Factory project.
 - a. WebSphere Portlet Factory Designer should have been started in the “Tutorial Setup” section.
 - b. Click File, New, WebSphere Portlet Factory Project to create a project as shown in Figure 7.32.

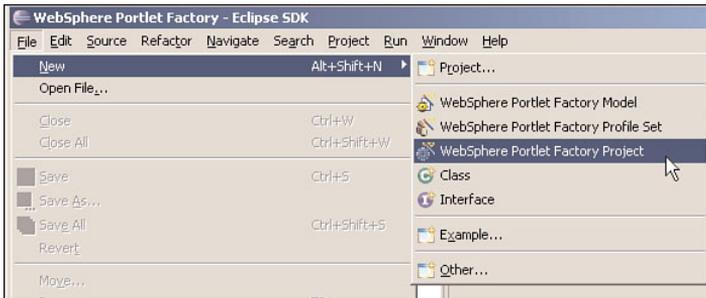


Figure 7.32 Create a WebSphere Portlet Factory project

- c. Enter `PetTravel` for the project name. Click Next as shown in Figure 7.33.
- d. There is no need to add feature sets to your project as shown in Figure 7.34. These feature sets provide additional builders that are not needed for this tutorial. Click Next.
- e. Select `WAS61` from the drop-down for application server deployment configuration as shown in Figure 7.35. Click Finish.

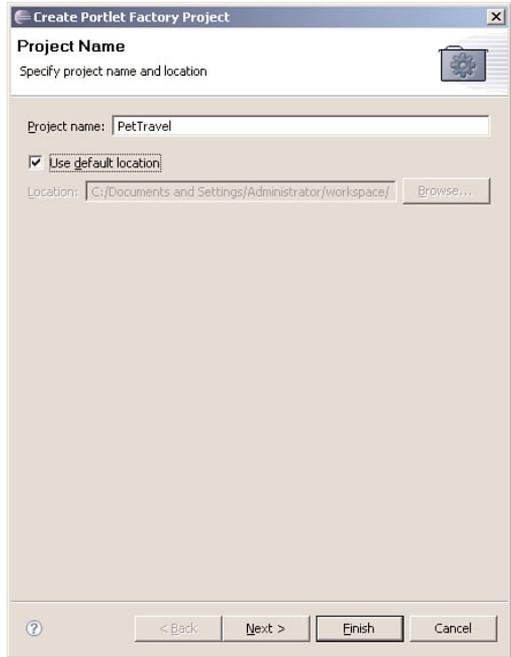


Figure 7.33 Enter the project name

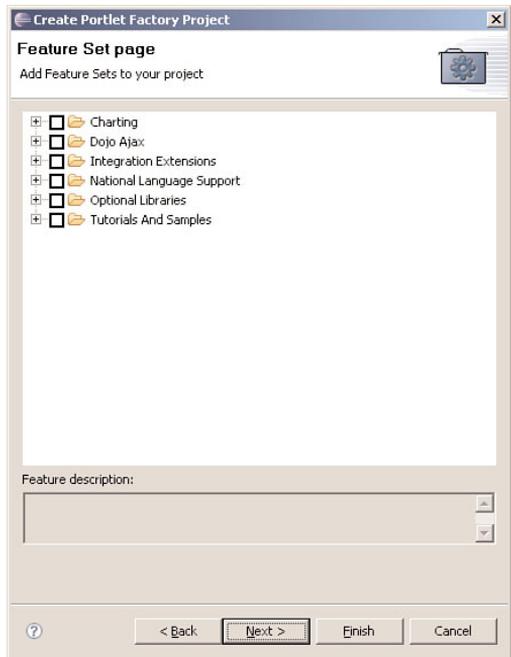


Figure 7.34 Select feature sets

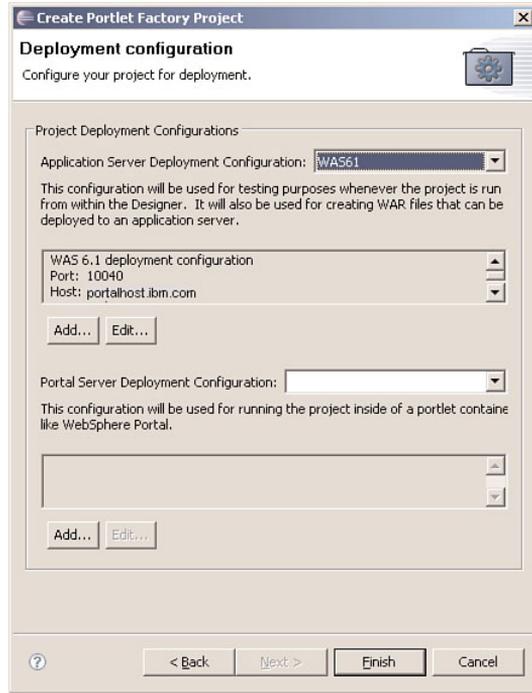


Figure 7.35 Specify application server deployment configuration

2. After the progress indicator, click No when asked whether you would like to deploy your project now as shown in Figure 7.36. You will be shown how to deploy your project in a development environment in a separate step.

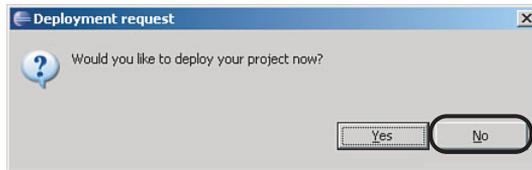


Figure 7.36 Deployment Request dialog

3. Your project `PetTravel` has been created successfully and appears in the Project Explorer window as shown in Figure 7.37.

You have successfully created the project.

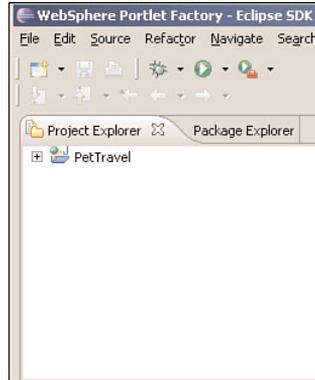


Figure 7.37 Project created

Step 2: Create and Test a Simple Portlet

In this step, you will create a simple model. You will then test the model using the deployment configuration you created in the setup section.

1. Launch the WebSphere Portlet Factory Model Wizard to create a Main and Page model. It is one of the sample models provided by the IBM WebSphere Portlet Factory product. A Main and Page model supplies a page and a main action that presents that page when the application is run.
 - a. Right-click `PetTravel` and click `New, WebSphere Portlet Factory Model` as shown in Figure 7.38.

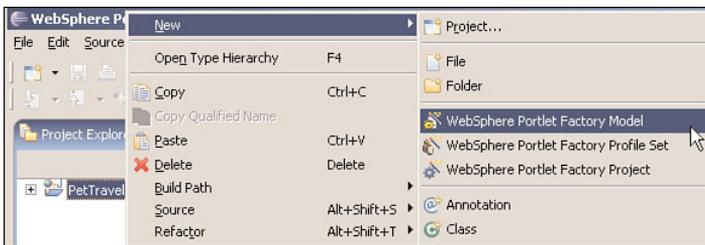


Figure 7.38 Create a new WebSphere Portlet Factory model

- b. Select `PetTravel` as the project as shown in Figure 7.39. Click `Next`.
 - c. Under `Factory Starter Models`, select `Main and Page` as shown in Figure 7.40. Click `Next`.

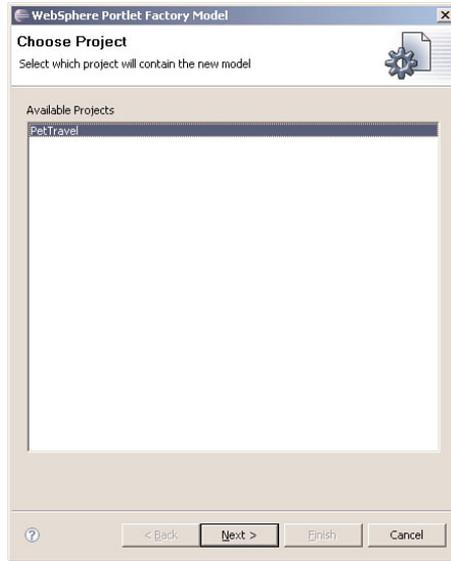


Figure 7.39 Select the project to contain the model

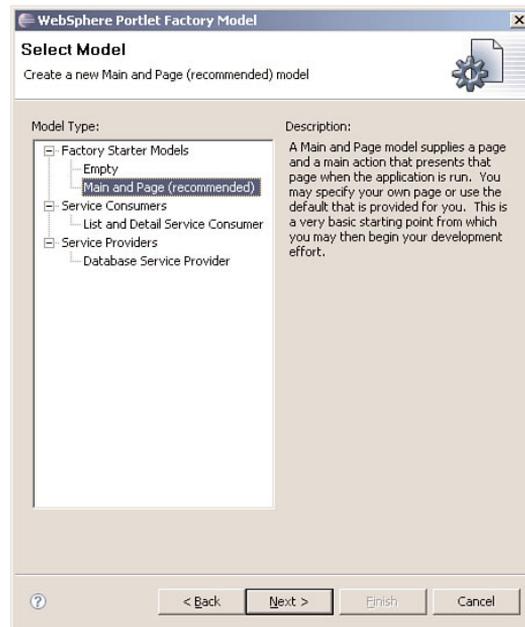


Figure 7.40 Select to create a Main and Page model

- d. Keep all the defaults as shown in Figure 7.41. Click Next.



Figure 7.41 Specify page type

- e. Enter PetTravelFAQ as the model name as shown in Figure 7.42. Click Finish.



Figure 7.42 Specify the model name and save

- 2. Modify the content of the HTML page in the model.
 - a. The `PetTravelFAQ` model has been created successfully as shown in Figure 7.43. The outline window at the bottom shows the list of builders in this model. Double-click the imported page builder `page1`.

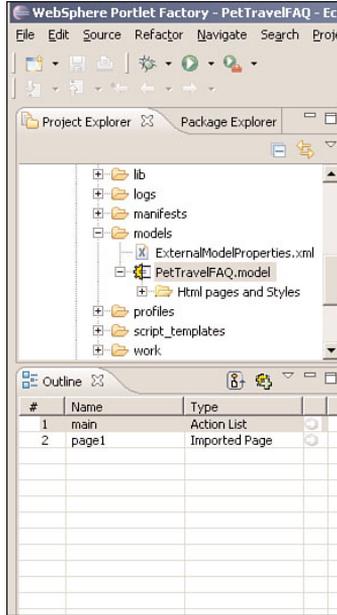


Figure 7.43 Main and Page model created

- b. Click Edit Page to modify the page content in HTML as shown in Figure 7.44.

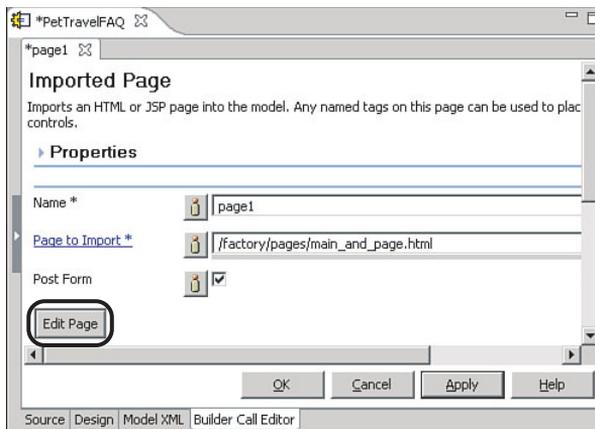


Figure 7.44 Edit page

- c. To modify the page content, follow the example as shown in Figure 7.45 or provide your own HTML content. Click Save Page, then Apply and OK to save changes for the builder.

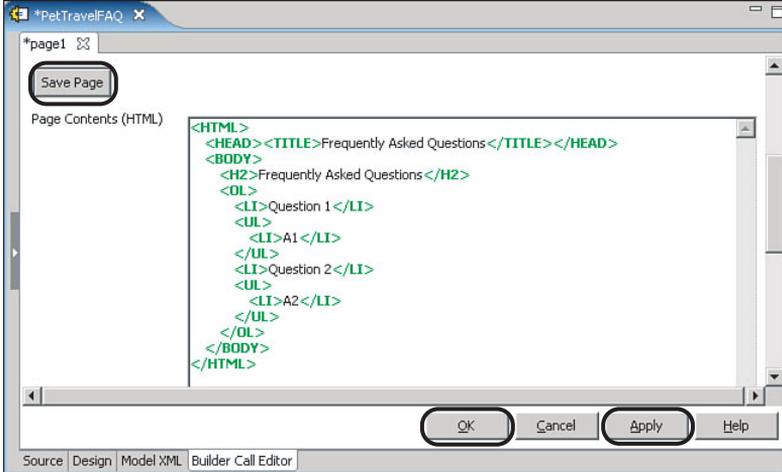


Figure 7.45 Modify page content and save builder changes

- d. Click File, Save to save changes for the model as shown in Figure 7.46.

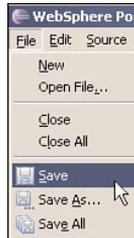


Figure 7.46 Save model changes

- 3. Deploy the project for development testing.
 - a. Right-click `PetTravel` and click Application Server WAR, Build WAR for Dev Testing as shown in Figure 7.47.

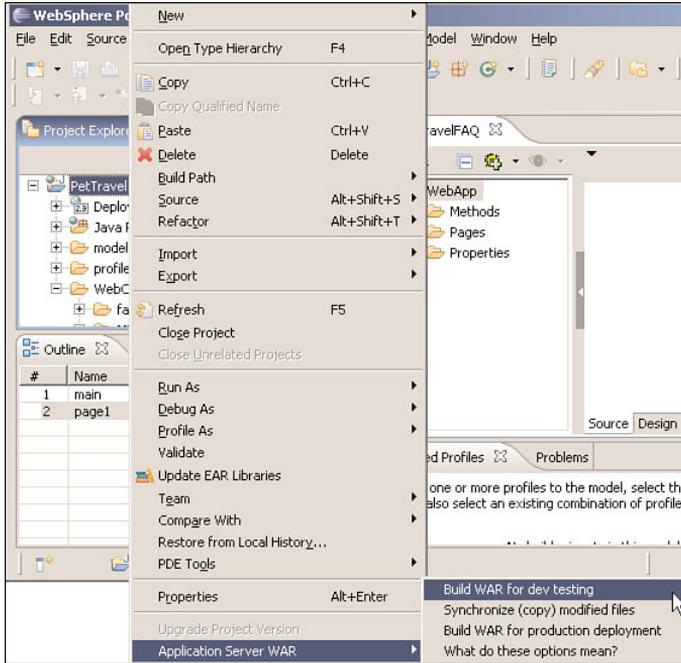


Figure 7.47 Deploy project for development testing

- 4. Test the model.
 - a. Click the Run Active Model icon as shown in Figure 7.48.

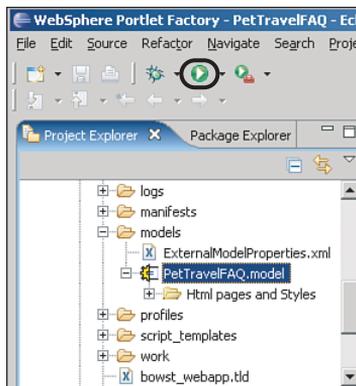


Figure 7.48 Run selected model

- b. A little setup is needed for the first time. Expand WebSphere Portlet Factory Model and select Active Model as shown in Figure 7.49. Click Run.

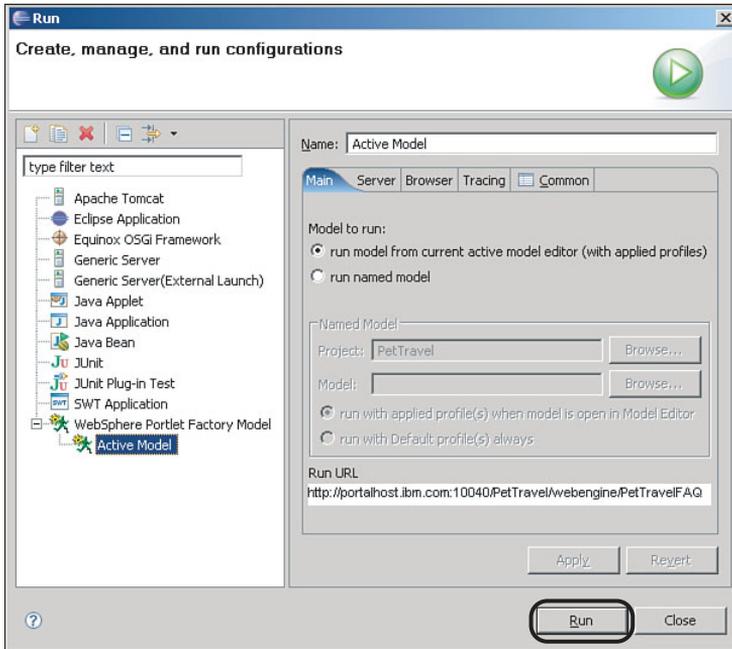


Figure 7.49 Run setup

- c. Your model should run successfully. The page you provided HTML content for is shown in a browser as in Figure 7.50.

Frequently Asked Questions

1. Question 1
 - o A1
2. Question 2
 - o A2

Figure 7.50 Main and page model successfully run

- 5. It is simple to make this model run as a portlet as well. Only one builder needs to be added.
 - a. In the outline window, click the Add a Builder Call to the Current Model icon as shown in Figure 7.51.

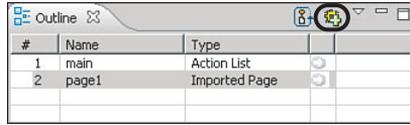


Figure 7.51 Add a builder call

- b. In the Builder Picker window as shown in Figure 7.52, select Portal Integration for the category name and Portlet Adapter for the builder. Click OK.

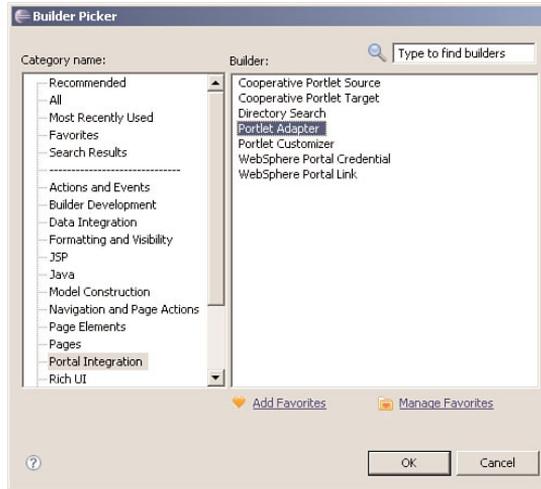


Figure 7.52 Select a builder to add to the model

- c. Enter PetTravelInfo as the name and the portlet title. Enter Pet Travel Information as the portlet short title. Click Apply and then OK as shown in Figure 7.53.

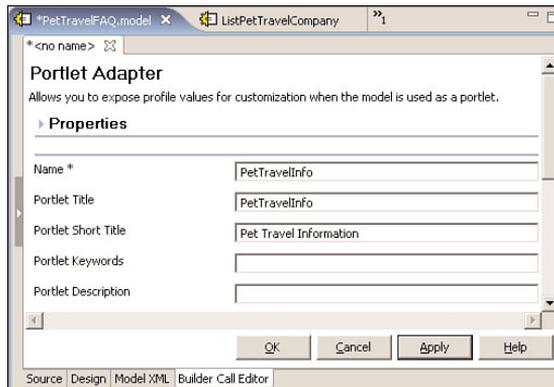
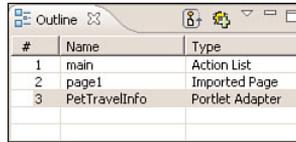


Figure 7.53 Fill in information for the Portlet Adapter builder

- d. The Portlet Adapter builder has been added to the model as shown in Figure 7.54. Click File, Save.



#	Name	Type
1	main	Action List
2	page1	Imported Page
3	PetTravelInfo	Portlet Adapter

Figure 7.54 The Portlet Adapter builder added to model

6. You only need to do this step if you will not be doing the next section to create and test a portlet that accesses a database. Right-click `PetTravel` and click Portal Server WAR, Build Portlet WAR for Production Deployment. Keep the defaults. Click Finish.

You have created and tested the model. The model is used to generate the portlet.

Tutorial 7.2: Create and Test a Portlet That Accesses a Database

In this tutorial, you will create a service provider model to retrieve data from the database and create a service consumer model to present the data.

Step 1: Create a Service Provider Model

In this step, you will create a service provider model.

1. Launch the WebSphere Portlet Factory Model Wizard to create a service provider model. It is one of the sample models provided by the IBM WebSphere Portlet Factory product. This sample model allows you to execute SQL statements against a database.
 - a. Right-click `PetTravel` and click New, WebSphere Portlet Factory Model.
 - b. Select `PetTravel` as the project. Click Next.
 - c. Under Service Providers, select Database Service Provider as shown in Figure 7.55. Click Next.
 - d. Enter `PetTravelServiceProvider` as the service name. Click Next.
 - e. In step 1 of defining the service operation as shown in Figure 7.56, select `jdbc/PetDB` from the drop-down for SQL datasource and enter `Select * from Administrator.PetTravelCompany` as the SQL statement. Click Next.

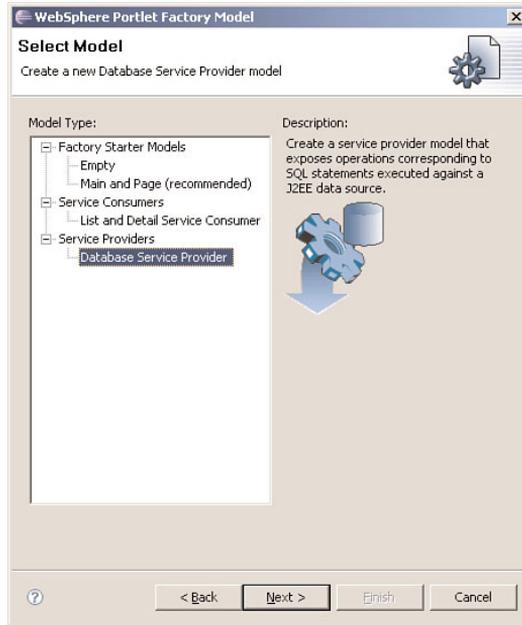


Figure 7.55 Select to create a Database Service Provider model



Figure 7.56 Step 1 of defining the service operation

- f. In step 2 of defining the service operation as shown in Figure 7.57, enter `getCompany` as the operation name and keep all other defaults. Click Next.

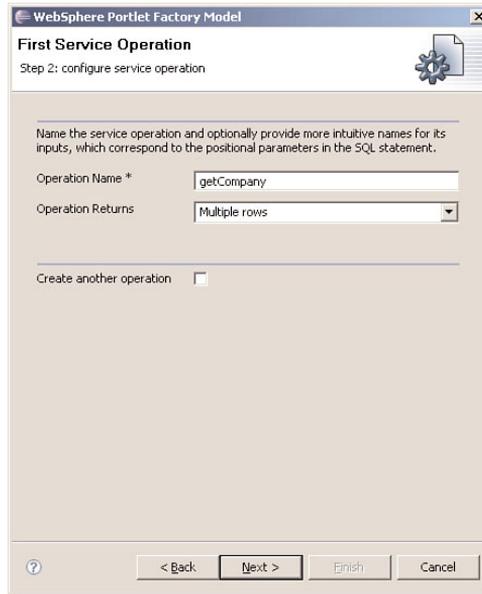


Figure 7.57 Step 2 of defining the service operation

- g. Enter `PetTravelCompany` as the model name. Click Finish.
2. The `PetTravelCompany` model has been created successfully as shown in Figure 7.58. The outline window at the bottom shows the list of builders in this model.

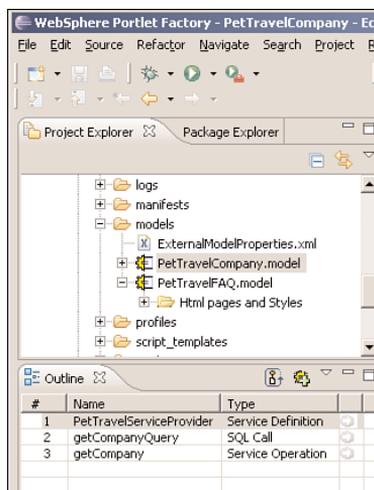


Figure 7.58 Database Service Provider model created

3. Test the model.
 - a. PetTravelCompany should be the selected model at this time. To test the model, click the Run Active Model icon.
 - b. Your model should run successfully. You should see the `getCompany` operation you defined in the browser as shown in Figure 7.59. Click `getCompany`.



Figure 7.59 Database Service Provider model run successfully

- c. You should see the data from the database table `PetTravelCompany` displayed correctly as shown in Figure 7.60.

ID	COMPANY	DESC
1	AA	Pet travel insurance.
2	BB	Pet carrier for dogs.
3	CC	Pet friendly accomodation.
4	DD	Pet sitting while you travel.

Back

Figure 7.60 Data from database retrieved and displayed

In this step, you created a service provider model. You should see that the pieces are starting to come together. The service provider retrieves the data in the database table created in the “Tutorial Setup” section.

Step 2: Create a Service Consumer Model

In this step, you will create a service consumer model that will use a simple table to display the results from the provider model.

1. Launch the WebSphere Portlet Factory Model Wizard to create a service consumer model. It is one of the sample models provided by the IBM WebSphere Portlet Factory product. This sample model displays data from a database.
 - a. Right-click `PetTravel` and click New, WebSphere Portlet Factory Model.
 - b. Select `PetTravel` as the project. Click Next.

- c. Under Service Consumers, select List and Detail Service Consumer as shown in Figure 7.61. Click Next.

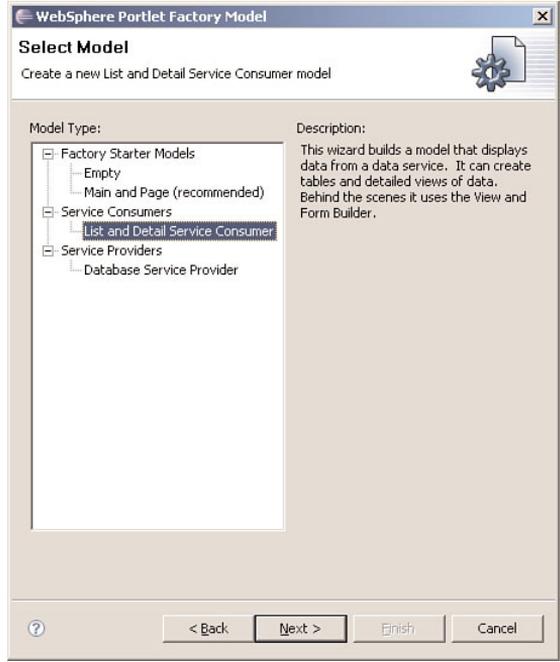


Figure 7.61 Select to create a List and Detail Service Consumer model

- d. Enter `PetTravelCompanySimpleTable` as the name and select `PetTravelCompany` as the provider model as shown in Figure 7.62. Recall that you created the `PetTravelCompany` model in the previous step. Click Next.
- e. Select `getCompany` from the drop-down for view data operation to provide the view data as shown in Figure 7.63. Recall that you defined the service operation `getCompany` to retrieve data from the database table `PetTravelCompany`. Click Next.
- f. Select `COMPANY` from the drop-down for details link column and keep all other defaults as shown in Figure 7.64. Recall that you created the database table `PetTravelCompany` with `ID`, `Company`, and `Desc` as the three columns. This is why `Company` appears as a choice in the drop-down. Click Next.

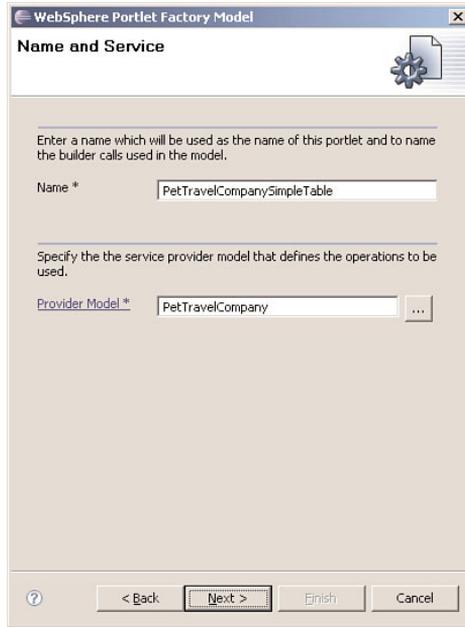


Figure 7.62 Enter name and service

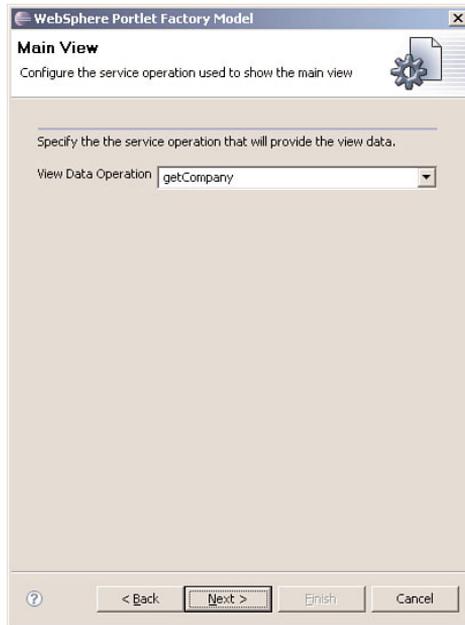


Figure 7.63 Specify service operation to provide view data

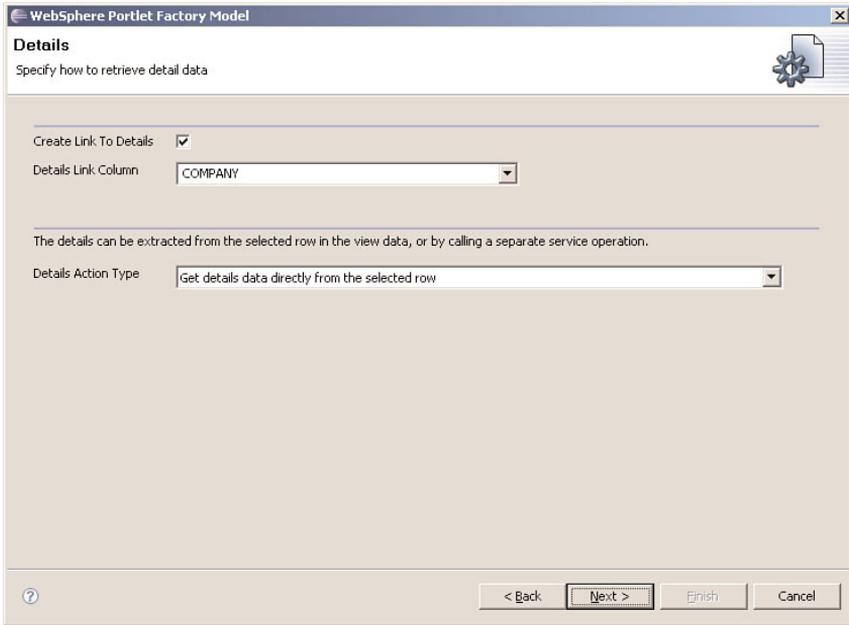


Figure 7.64 Specify how to get detail data

- g. Enter `ListPetTravelCompany` as the model name. Click Finish.
- 2. The `ListPetTravelCompany` model has been created successfully as shown in Figure 7.65. The outline window at the bottom shows the list of builders in this model.

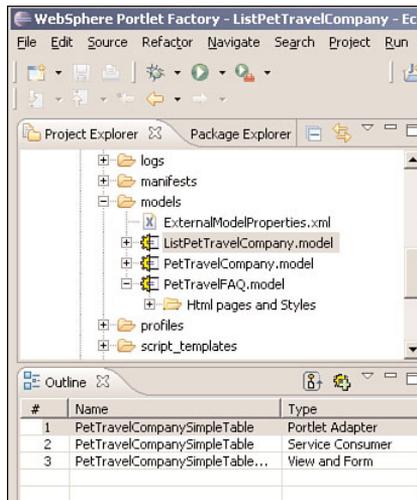


Figure 7.65 List and Detail Service Consumer model created

- 3. Test the model.
 - a. `ListPetTravelCompany` should be the selected model at this time. To test the model, click the Run Active Model icon.
 - b. You should see the data from the database table `PetTravelCompany` displayed correctly in a simple table with the Details link using the `Company` column as shown in Figure 7.66.

ID	COMPANY	DESC
1	AA	Pet travel insurance.
2	BB	Pet carrier for dogs.
3	CC	Pet friendly accomodation.
4	DD	Pet sitting while you travel.

Figure 7.66 List and Detail Service Consumer model run successfully

- 4. Build a portlet WAR file for deploying in production. IBM WebSphere Portal is the production server.
 - a. Right-click `PetTravel` and click Portal Server WAR, Build Portlet WAR for Production Deployment as shown in Figure 7.67.



Figure 7.67 Build portlet WAR for production

- b. Keep the defaults as shown in Figure 7.68. Click Finish.

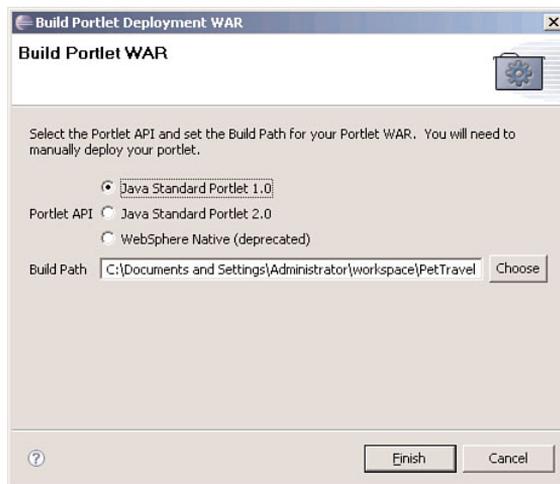


Figure 7.68 Select portlet API and build path

- c. Go to the above build path. You should see the `PA_PetTravel.war` WAR file.

You have created a service consumer model to present data in a simple table.

Tutorial 7.3: Deploy a Portlet

In this tutorial, you will install the portlets you built in Tutorials 7.1 and 7.2 in IBM WebSphere Portal. You will then add the portlets to portal pages and set permissions for the portlets.

Step 1 shows you how to install a portlet in a production machine/environment performed by an administrator. If you want to do additional testing in a test machine/environment first with WebSphere Portal installed, you can deploy directly from WebSphere Portlet factory using a deployment configuration. Remaining steps on adding a portlet to a portal page or setting access permissions are the same.

Step 1: Install a Portlet in IBM WebSphere Portal

In this step, you will install the portlets to WebSphere Portal.

1. Log in to WebSphere Portal.
 - a. Start the `WebSphere_Portal` server if it is not already running.
 - b. Open a browser and enter the URL `http://<hostname>:<port_number>/wps/portal`, where *<hostname>* is the fully qualified host name of the machine that is running your WebSphere Portal, and *<port_number>* is the port number displayed on the confirmation panel during your install.

For example, enter `http://portalhost.ibm.com:10040/wps/portal`

- c. Log in using your WebSphere Portal administrator user ID and password.
2. Install the portlets.
 - a. Click Administration at the top of the page as shown in Figure 7.69.



Figure 7.69 Select WebSphere Portal administration

- b. In the left pane, click Portlet Management, Web Modules as shown in Figure 7.70.
- c. On the Manage Web Modules page, click Install as shown in Figure 7.71.
- d. The Installing a Web Module Wizard appears. In step 1 of the wizard as shown in Figure 7.72, browse to the build path that you specified when you built the portlet WAR. Select `PA_PetTravel.war`. Click Next.



Figure 7.70 Portlet management

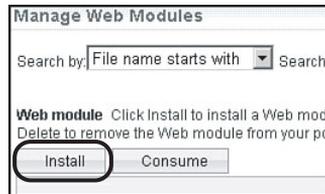


Figure 7.71 Manage Web modules

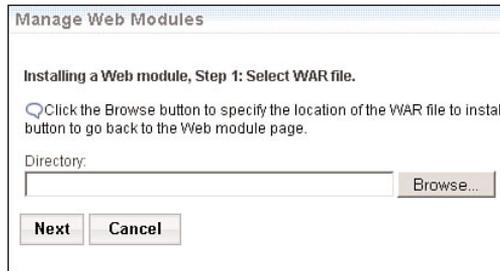


Figure 7.72 Step 1 of Installing a Web Module Wizard

For example: browse to C:\Documents and Settings\Administrator\workspace\Pet-Travel\PA_PetTravel.war.

- e. In Step 2 of the wizard as shown in Figure 7.73, review the portlets that are going to be installed. You should see the two portlets that you created. Click Finish.

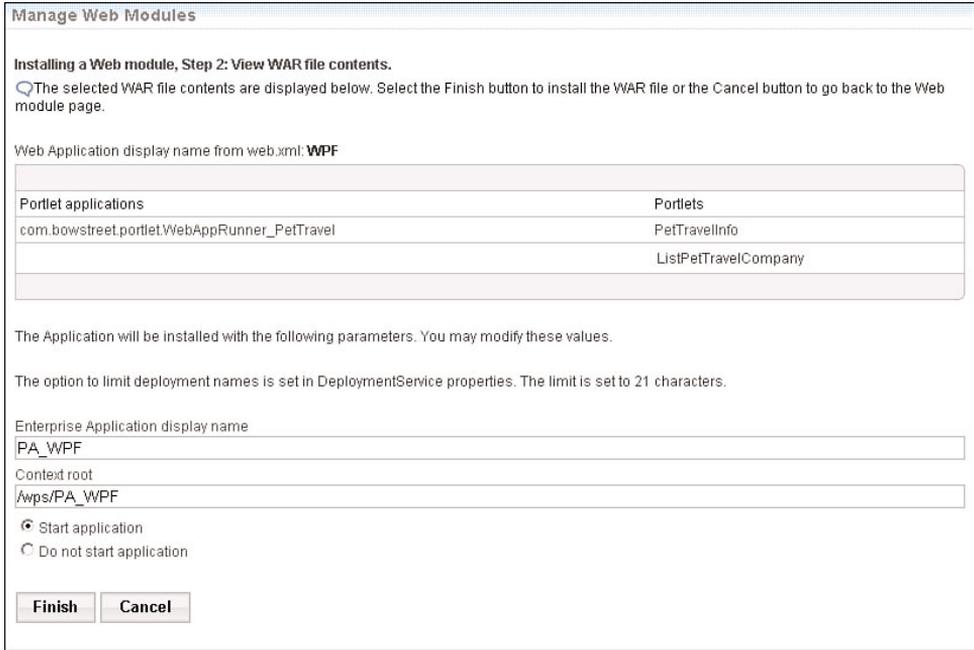


Figure 7.73 Step 2 of Installing a Web Module Wizard

- f. Once the installation is complete, a message indicating that the installation was successful is shown at the top of the page as in Figure 7.74.



Figure 7.74 Portlet successfully installed

- 3. To verify the installation, search by File Name Contains and enter `PetTravel` as the search criteria. You should see `PA_PetTravel.war` in the list of Web modules as shown in Figure 7.75.

In this step, you have installed the portlet application that was developed using WebSphere Portlet Factory.

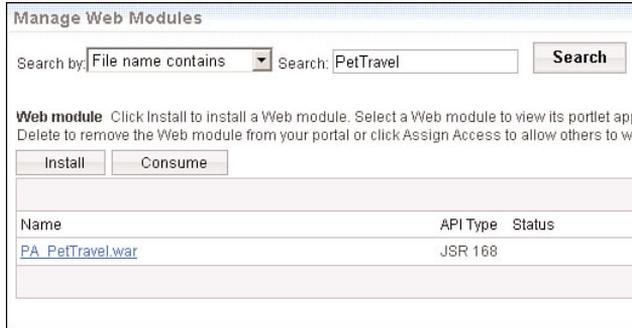


Figure 7.75 Verify portlet installation

Step 2: Add a Portlet to a WebSphere Portal Page

In the previous step, you installed the portlets. They will not appear in the portal site yet. They need to be added to portal pages. In this step, you will create a portal page and add the two portlets to the portal page.

1. Create a portal page.
 - a. You should still be in the Administration page. In the left pane, click Portal User Interface, Manage Pages as shown in Figure 7.76.



Figure 7.76 Manage pages

- b. On the Manage Pages page, click Content Root. When you are at Content Root, click Home. Click New Page to create a new page under Home as shown in Figure 7.77.

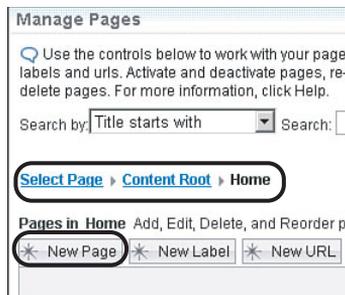


Figure 7.77 Create new page

- c. Enter `Pet Travel` for the title as shown in Figure 7.78. Click OK.

Page Properties

New page: Content Root

Use the controls below to work with your pages options.

Title:

Unique Name:

Note: If the unique name you entered for this page
Friendly URL name:

Theme:
-----Portal Default Theme-----

Theme Style (Theme Policy):
SingleTopNav

Icon:

I want to make this page my private page

Page Properties

Type of Page

Page Cache Options

Figure 7.78 Page properties for new page

- d. You should see `Pet Travel` in the list of pages as shown in Figure 7.79.

Manage Pages

Use the controls below to work with labels and urls. Activate and deactivate pages. For more information, click

Search by:

[Select Page](#) > [Content Root](#) > [Home](#)

Pages in Home Add, Edit, Delete, and

New Page New Label New Label

Title

- [Getting Started](#)
- [Web 2.0 Introduction](#)
- [Feeds](#)
- [Gadgets](#)
- [Pet Travel](#)

Figure 7.79 Verify page creation

- e. Select Home, and you will see the `Pet Travel` page.
- 2. Add portlets to a portal page.
 - a. Click the Page Menu icon and click Edit Page Layout as shown in Figure 7.80.



Figure 7.80 Launch page menu

- b. Click Add Portlets in the left column in the layout as shown in Figure 7.81.

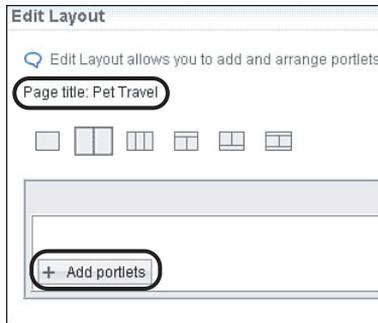


Figure 7.81 Edit layout

- c. A search screen renders on the page to let you search for the portlet you want to add. Search by Title Contains and enter `PetTravel` as the search criteria as shown in Figure 7.82.



Figure 7.82 Search for portlet

- d. In the search results, select the check box for both `ListPetTravelCompany` and `PetTravelInfo` as shown in Figure 7.83. Click OK.
 - e. `ListPetTravelCompany` and `PetTravelInfo` appear in the left column of the layout page as shown in Figure 7.84. Click the down arrow to the right of `ListPetTravelCompany` to move it below `PetTravelInfo`. Click Done.

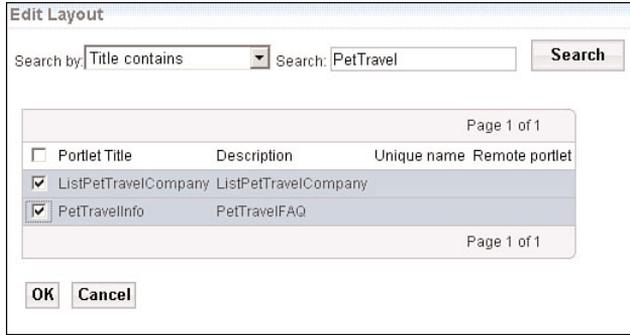


Figure 7.83 Select portlet

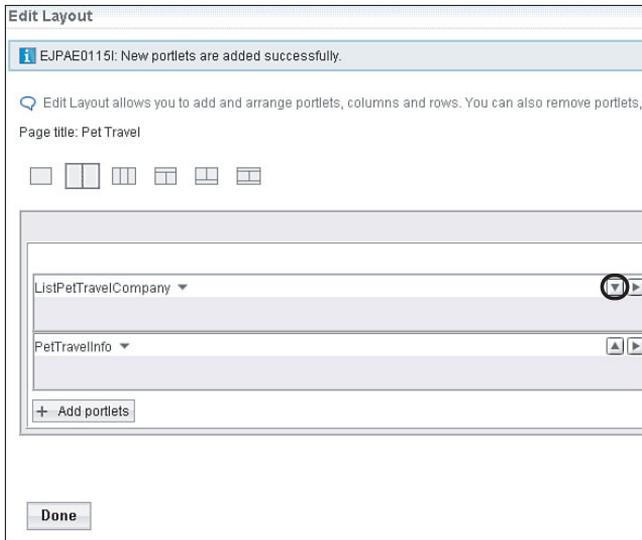


Figure 7.84 Portlet added

3. Your two portlets appear in the portal site as shown in Figure 7.85.

In this step, you have added the two portlets to a portal page. They now appear in the portal site.

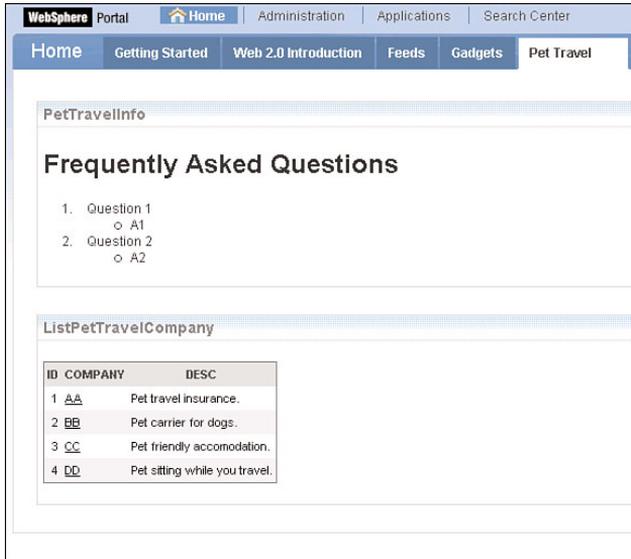


Figure 7.85 Portlets added to home page

Step 3: Access a Portlet as a New User

In this step, enroll in WebSphere Portal as a new user by filling out your own profile information; then log into Portal using the new ID.

1. Sign up as a new WebSphere Portal user.
 - a. Open a browser and enter the URL `http://<hostname>:<port_number>/wps/portal` where `<hostname>` is the fully qualified host name of the machine that is running your WebSphere Portal, and `<port_number>` is the port number.
For example, enter `http://portalhost.ibm.com:10040/wps/portal`.
 - b. Click the Sign Up link above the Log In button.
 - c. Fill in your user profile information to enroll in WebSphere Portal. You can choose your own user ID in Portal. In the example that follows as shown in Figure 7.86, the User ID registered is Mary. Click OK.
 - d. A message is displayed saying the user was created successfully.

Edit My Profile

🗨 New user enrollment. Provide the information requested

* User ID:

* Password:

* Confirm Password:

First Name:

* Last Name:

Email:

Preferred language:

* Required Field

Figure 7.86 Sign up as new user in WebSphere Portal

2. Log in to Portal using the newly created user Mary.
3. Do you see the two portlets on the *Pet Travel* page? The portlets are not on user Mary's *Pet Travel* page. When you installed the portlets in WebSphere Portal, you did not grant users with access permission to the portlets. That is why you can only view the portlets as the administrator. No other users can view them at this point.
4. Exit WebSphere Portal by clicking Log Out in the upper-right corner.

In this step, you logged in as a new user and were not able to access the portlets because you have not been granted access yet.

Step 4: Set Access Permissions for a Portlet

In this step, all authenticated portal users will be granted access to the portlets.

1. Go to the administrative function for setting permissions for portlets. Grant users access to the newly installed portlets.
 - a. Log in to the WebSphere Portal as the Administrator again.
 - b. Click Administration at the top of the page.
 - c. Click Access, Resource Permissions.

- 2. Grant users access to the newly installed portlets.
 - a. In the Resource Permissions page, as shown next in Figure 7.87, click Portlets.



Figure 7.87 Resource Permissions page

- b. Search by Title Contains and enter `PetTravel` as the search criteria.
 - c. Click the Assign Access icon displayed beside the `PetTravelInfo` resource as shown in Figure 7.88.

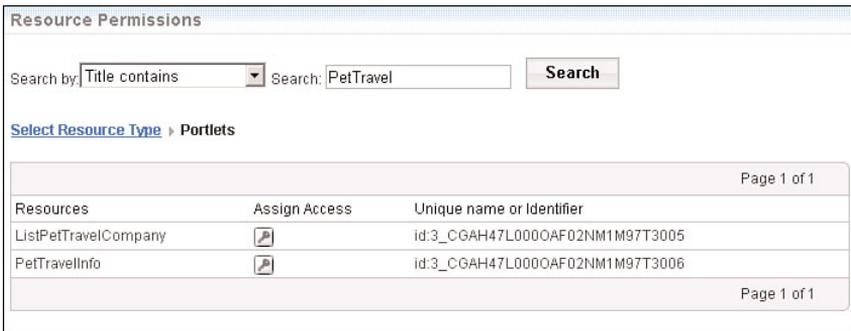


Figure 7.88 Assign Access

- d. Click the Edit Role icon beside the User role as shown in Figure 7.89.
 - e. There are currently no members in this role, as shown next in Figure 7.90. This explains why you weren't able to see the `PetTravelInfo` portlet when you logged in as a user who is not an administrator (for example, Mary). Click Add to add members to this role.

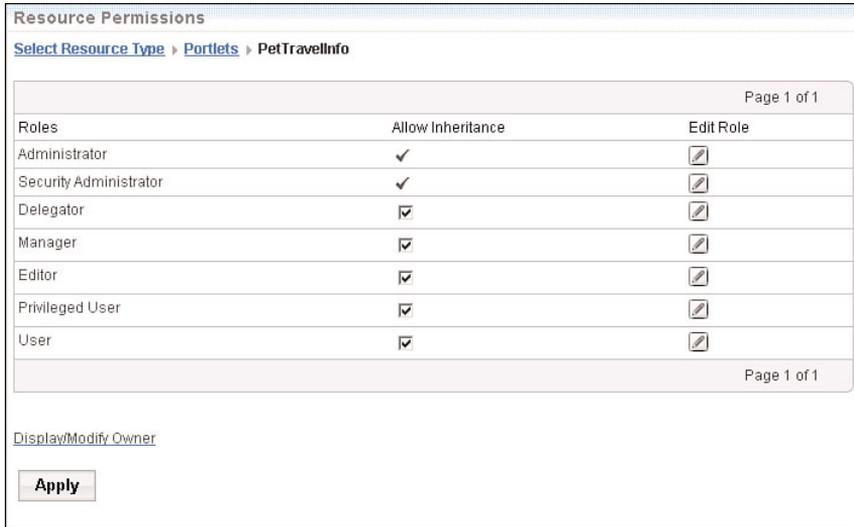


Figure 7.89 Modify access for a role



Figure 7.90 Members with access are shown

- f. Select the All Authenticated Portal Users check box. Click OK.
- 3. Log out of Portal, and log in as a user who is not an administrator. For example, log in as Mary.
- 4. What do you see in the Pet Travel page? You should see PetTravelInfo but not ListPetTravelCompany as shown in Figure 7.91. You have granted users access to the PetTravelInfo portlet but have not done that for the ListPetTravelCompany portlet yet.

In this step, you set access permissions to the portlet PetTravelInfo and confirmed that all portal users now have access to the portlet.

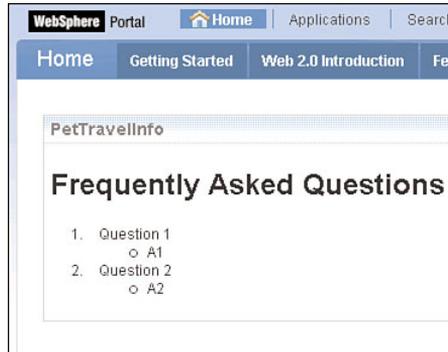


Figure 7.91 PetTravellInfo accessible by all users

Summary

In this chapter, you were introduced to the software IBM WebSphere Portlet Factory for developing portlets and the IBM WebSphere Portal for building, managing, and hosting portals.

First, as an overview, the chapter described key concepts and features. Then the tutorial was used to give you hands-on experience and reinforce the material introduced.

In the three-part tutorial, you developed and tested a simple portlet that displays information and a portlet that returns data from a database. You also deployed the portlets and saw them working in a production environment.

Tutorial summary:

- Tutorial 7.1: Create and Test a Simple Portlet
 - Create a project
 - Create a model
 - Add a builder
 - Create a portlet
 - Test a model
 - Build a WAR file for production deployment
- Tutorial 7.2: Create and Test a Portlet That Accesses a Database
 - Create a service provider model
 - Create a service consumer model

- Tutorial 7.3: Deploy a Portlet
 - Install a portlet
 - Create a portal page
 - Add a portlet to a portal page
 - Edit portal page layout
 - Give access to users on new portlets
 - Illustrate single login to portal website rather than to individual application

Index

A

access permissions, setting for portlets, 294-296
accessing portlets, 293-294
activity diagrams, 14
administrative console (WebSphere Application Server). *See* integrated solutions console (WebSphere Application Server)
administrator setup (WSRR tutorial), 129
aggregation association relationships, 25
annotations, JPA entity classes, 81-82
artifacts (services)
 importing, 143-146
 loading, 150-151
 publishing, 116, 119
 storing, 157
assembling business processes, 181-189
assembly diagrams, 158, 181-183
Assign nodes, creating for variables, 170-172

associating runtime artifacts with WebSphere Message Broker Toolkit, 219-220
association relationships
 aggregation association relationships, 25
 directed association relationships, 24-25
authentication, LTPA security tokens, 8
authorization constraints, 328

B

Basic Profile (WSRR), 117
BPM (Business Process Management), SOA support for, 159
Broker Administration perspective, 202-203
Broker Application Development perspective, 202
broker domain configuration setup, 212-213
broker domains, 200
brokers, 200
builders (WebSphere Portlet Factory), 244
buses, ESB, 5-6, 117
business model templates (WSRR), 120
 setup, 130-131
business objects
 creating, 164-166
 defined, 157
Business Process Choreographer Explorer, 186, 194, 196
Business Process Management (BPM), SOA support for, 159
business processes, 155, 157
 designing, WebSphere Business Modeler (IBM), 3
 integration tutorial
 assembling and executing business processes, 181-189
 CD-ROM contents, 161-162
 creating business processes, 162-181
 deploying to WebSphere Process Server, 189-196
 system requirements, 161
 process portals, 5

C

- calculator service security
 - tutorial
 - CD-ROM contents, 308-309
 - enabling service provider security, 309-324
 - enabling service requester security, 325-340
 - system requirements, 307-308
 - testing, 341
- class diagrams, creating, 23-32
- classification systems, loading, 133-134
- classifications (service metadata), 116, 122
- collaboration (WebSphere Portal), 246
- columns (databases),
 - troubleshooting unresolved columns, 100
- composite applications (WebSphere Portal), 246
- concepts (services)
 - creating, 137-139, 147-148, 151-152
 - deprecating, 152-153
 - governance, 139-140
 - publishing, 116, 120
 - validations, 120
- conceptual protocols, UML-to-Java transformations, 40
- configuration managers, 200
- configuration profiles (WSRR), 117
- configuring
 - databases with WebSphere Message Broker Toolkit, 215-218
 - input message parse options, 227
 - message flow, 225-227
 - service providers for WSSM token consumption, 310-317
 - service requesters for WSSM token generation, 327-333
 - STS trust service chain
 - for token transformation, 333-338
 - for token validation, 317-322
 - connectivity services, 5-6.
 - See also* WebSphere Message Broker
 - consuming security tokens, 303
 - content management (WebSphere Portal), 246
 - custom relationships (service metadata), 116

D

- data models, creating, 75-76, 79-80
- database setup, 210-212
- database tables
 - data models, creating, 75-76, 79-80
 - database connections, creating, 73-74
- databases
 - configuring with WebSphere Message Broker Toolkit, 215-218
 - creating in DB2, 255-262
 - troubleshooting unresolved database columns, 100
- datasource NameNotFound errors, troubleshooting, 102
- datasources, creating, 262-266
- DB2, creating databases and tables, 255-262
- DDL, 78
- Debug perspective, 203
- defining message flow logic, 230-232
- deploying
 - business process integration tutorial to WebSphere Process Server, 189-196
 - module projects to WebSphere Process Server, 185-186
 - portlet tutorial, 286-296

- service providers to WAS, 323-324
- service requesters to WAS, 338-340
- Web services
 - changing Web service endpoints, 107-109
 - installing enterprise applications, 105-107
 - port examinations, 110-112
 - server log files, 112-113
 - setting up databases, 104
 - starting integrated solutions console (WebSphere Application Server), 104
 - starting WebSphere Application Server, 104
 - stopping WebSphere Application Server, 114
 - updating JDBC datasources, 110
- deployment configurations, creating, 250-253
- directed association relationships, 24-25
- document groups, saving, 120
- dynamic endpoint selection, 117

E

- EAR (Enterprise Application) files, 58
 - exporting projects as, 189
 - installing, 190-192
 - starting, 190-191
 - Web services, exporting as EAR files, 103
- Eclipse workbench, importing artifacts, 143-146
- enabling
 - service provider security, 309-324
 - service requester security, 325-340
- endpoints, changing, 192-194

Enterprise Explorer, Set Up
 Deployment Wizard, 97
 enterprise service bus (ESB),
 5-6, 117
 entity classes (JPA)
 annotations of, 81-82
 orm.xml files, 83
 persistence.xml files, 84
 JDBC configurations in,
 97-98
 ESB (enterprise service bus),
 5-6, 117
 ESQL (Extended Structured
 Query Language), 201
 creating for search query,
 227-230
 executing business processes,
 194-196
 business process integration
 tutorial, 181-189
 exporting projects as EAR
 files, 189
 Extended Structured Query
 Language (ESQL), 201
 creating for search query,
 227-230
 external Web services
 assembling, 181-183
 copying WSDL interface files
 from, 164

F-G-H

federation, defined, 299. *See also*
 TFIM
 finding services, 116-121,
 141-143
 flexibility, 247
 Generate DDL Wizard (Rational
 Application Developer), 78
 generating security tokens, 302
 governance (services), 9,
 116-117
 WSRR tutorial
 administrator setup, 129
 business model template
 setup, 130-131

CD-ROM contents, 128
 finding services, 141-143
 loading classification
 system, 133-134
 publishing services,
 135-140
 reusing services, 141-148
 system requirements, 128
 updating services,
 148-153
 Governance Enablement Profile
 (WSRR), 117
 HTML (Hypertext Markup
 Language), publishing service
 designs, 38-39
 human tasks, 157
 creating in business
 processes, 174-176

I

IBM Information Server, 4
 IBM Rational Software
 Architect, 4
 IBM Tivoli Access Manager, 8
 IBM WebSphere Business
 Modeler, 3
 RSA support for, 15
 IBM WebSphere Business
 Service Fabric, 6
 IBM WebSphere DataPower
 Integration Appliance X150, 6
 IBM WebSphere Enterprise
 Service Bus, 6
 IBM WebSphere Message
 Broker. *See* WebSphere
 Message Broker
 IBM WebSphere Portal. *See*
 WebSphere Portal
 IBM WebSphere Portlet Factory.
See WebSphere Portlet Factory
 IBM WebSphere Service
 Registry and Repository. *See*
 WSRR (WebSphere Service
 Registry and Repository)
 identity, 7

identity management. *See* TFIM
 identity providers, 299
 impact analysis (WSRR), 124,
 148, 150
 implementation types (services),
 156-158
 importing
 artifacts, 143-146
 project interchange files, 162
 service documents, 135-136
 Web services to
 workbench, 162
 Information Server (IBM), 4
 input message parse options,
 configuring, 227
 installing
 EAR files, 190-192
 portlets to WebSphere Portal,
 286-288
 integrated solution diagrams,
 183-185
 integrated solutions console
 (WebSphere Application
 Server), 58
 starting, 104, 190
 URI, 58
 integrating services, 3, 155-159
 integration, 247
 integration solutions
 creating, 163
 defined, 157
 integration specialists, 155-156
 integration tutorial
 assembling and executing
 business processes, 181-189
 CD-ROM contents, 161-162
 creating business processes,
 162-181
 deploying to WebSphere
 Process Server, 189-196
 system requirements, 161
 invoke nodes, creating for
 business processes, 172-173

J

Java

Java-to-UML transformations, 46-50

JPA. *See* JPA (Java Persistence API)

UML-to-Java transformations, 40-41, 44-46

Java classes, creating Web service Java classes, 67-68

Java EE, EAR files, 58

Java EE perspective (Rational Application Developer), 56

Java objects, creating, 66-67

JDBC

datasources, updating, 110

persistence.xml files, configuring, 97-98

JPA (Java Persistence API)

entities, creating, 91-95

entity classes

annotations of, 81-82

orm.xml files, 83

persistence.xml files, 84, 97-98

Rational Application

Developer's support for, 83-84

Web service invocation, 80-84

creating JPA entities, 91-95

creating utility classes, 95-96

creating Web applications for Web service invocation, 87, 90-91

creating Web projects, 84

exporting projects as EAR files, 103

generating Web service clients, 86

JDBC configurations in persistence.xml files, 97-98

modifying Web

applications, 98-101

running Web applications, 101-102

running Web applications

in browsers, 91

troubleshooting

datasource

NameNotFound

errors, 102

troubleshooting

unresolved database

columns, 100

K-L

libraries, 157

library book search service

tutorial

associating runtime artifacts, 219-220

CD-ROM contents, 210

configuring input message parse options, 227

configuring library database, 215-218

configuring message flow, 225-227

creating ESQL for search query, 227-230

creating message definitions, 222-225

creating message mappings, 232-237

defining message flow logic, 230-232

setup, 210-215

broker domain

configuration setup, 212-213

database setup, 210-212

message queue setup, 213-215

system requirements,

208-209

testing message flow, 237-241

library projects, creating

integration solutions, 163

life cycles (services), 117

lifelines, sequence diagrams, 34-35

logging into WebSphere Portal, 293-294

LTPA (Lightweight Third-Party Authentication) security tokens, 8

M

managing services, 9

mappings, creating for messages, 232-237

message definitions, 201

creating, 222-225

message flows, 201

configuring, 225-227

defining logic for, 230-232

testing, 237-241

message maps, 201

creating, 232-237

message queue setup, 213-215

message sets, 201

messages, sequence diagrams, 34-35

metadata (services), 116-117, 122-126

mixed protocols, UML-to-Java transformations, 40

models

creating and testing in portlet tutorial, 270-278

WebSphere Portlet Factory, 245

module projects

creating integration solutions, 163

deploying to WebSphere Process Server, 185-186

modules, 157

MQSeries. *See* WebSphere MQ

N–O

New Physical Data Model Wizard (Rational Application Developer), 76
 notifications (WSRR), 124
 orm.xml files, JPA entity classes, 83
 OWL (Web Ontology Language), 120

P

page layout of portals, 245
 parse options for input message, configuring, 227
 permissions, setting for portlets, 294-296
 persistence.xml files
 JDBC configurations in, 97-98
 JPA entity classes, 84
 personalization (WebSphere Portal), 246
 perspectives (WSRR Web user interface), 119
 Plug-In Development perspective, 203
 portal pages, 244
 adding portlets to, 289-292
 portal server runtime, 245
 portal Web sites, 244
 portals. *See also* WebSphere Portal
 explained, 243-244
 page layout, 245
 skins, 245
 themes, 245
 portlet applications, 244
 portlet containers, 245
 Portlet Factory Designer, 244-245
 portlet tutorial
 CD-ROM contents, 250
 deployment, 286-296
 model creation and testing, 270-278
 project creation, 267-269
 service consumer model creation, 281-286
 service provider model creation, 278-281
 setup, 250-253, 256-266
 system requirements, 250
 portlets, 244
 accessing, 293-294
 adding to portal pages, 289-292
 installing to WebSphere Portal, 286-288
 servlets versus, 245
 setting access permissions, 294-296
 ports, 110-112
 process portals, 5
 Profile Management Tool (WebSphere Application Tool), 59
 programming APIs (WSRR), 121
 project interchange files, importing, 162
 projects
 creating in portlet tutorial, 267-269
 exporting as EAR files, 189
 properties (service metadata), 116, 122
 publishing
 service designs (HTML files), 38-39
 services, 116-121
 WSRR tutorial, 135-140

Q–R

Query Wizard (WSRR Web user interface), 121
 queue managers, 200
 queues, 200
 Rational Application Developer, 55
 application development, support for, 57
 Generate DDL Wizard, 78
 Java classes, creating Web service Java classes, 67-68
 Java EE perspective, 56
 Java objects, creating, 66-67
 JPA support, 83-84
 New Physical Data Model Wizard, 76
 Run SQL Wizard, 79
 SOA support for, 59-60
 Web projects, creating, 66
 Web service creation tutorial, 63-65
 CD-ROM contents, 62-63
 creating database tables, 73-76, 79-80
 creating Java classes, 67-68
 creating Java objects, 66-67
 creating Web projects, 66
 creating Web services, 68
 system requirements, 62
 testing Web services, 71
 WSDL interface, 69
 Web service invocation tutorial, 80, 84
 creating JPA entities, 91-95
 creating utility classes, 95-96
 creating Web applications for Web service invocation, 87, 90-91
 creating Web projects, 84
 exporting projects as EAR files, 103
 generating Web service clients, 86
 JDBC configurations in persistence.xml files, 97-98
 JPA annotations, 82
 JPA entity classes, 81
 modifying Web applications, 98-101
 running Web applications, 101-102

- running Web applications
 - in browsers, 91
 - troubleshooting
 - datasource
 - NameNotFound
 - errors, 102
 - troubleshooting
 - unresolved database
 - columns, 100
 - Web Service Wizard, 63, 68
 - Web services, support for, 60
 - WebSphere Application
 - Server integration, 57
 - workspace locations,
 - selecting, 55
 - Rational Software Architect (IBM), 4
 - reconciled protocols, UML-to-Java transformations, 40
 - registries (service), 9
 - Registry and Repository. *See* WSRR
 - relationships (service metadata), 116, 123
 - relationships
 - aggregation association
 - relationships, 25
 - directed association
 - relationships, 24-25
 - resources for information, 344
 - reusing services, 115-116
 - WSRR tutorial, 128, 141-148
 - role-based access (WebSphere Portal), 246
 - RSA (Rational Software Architect)
 - class diagrams, creating, 23-32
 - defining, 13
 - RTE, support for, 40-41, 44-50
 - sequence diagrams, creating, 33-35, 38
 - SOA, support for, 15
 - UML, support for, 13
 - UML models, creating, 18, 21
 - use case diagrams, creating, 21-22
 - WebSphere Business Modeler (IBM), support for, 15
 - RTE (round-trip engineering), 40-41, 44-50
 - Run SQL Wizard (Rational Application Developer), 79
 - runtime artifacts, associating with WebSphere Message Broker Toolkit, 219-220
- S**
- SAML (Security Assertion Markup Language) security tokens, 8
 - saving document groups, 120
 - search capabilities (WebSphere Portal), 246
 - search queries, creating ESQL for, 227-230
 - security. *See also* TFIM
 - calculator service tutorial
 - CD-ROM contents, 308-309
 - enabling service provider security, 309-324
 - enabling service requester security, 325-340
 - system requirements, 307-308
 - testing, 341
 - identity, 7
 - services, 7-9
 - Tivoli Access Manager (IBM), 8
 - transactions, 7
 - trust relationships, 299
 - establishing, 302-303
 - WebSphere Application Server, 59
 - WS-Security, 300
 - WS-Trust, 300-302
 - security constraints, 328
 - Security Token Service (STS), 300-303
 - security tokens, 7-8
 - consuming, 303
 - generating, 302
 - sequence diagrams
 - creating, 33-35, 38
 - lifelines in, 34-35
 - messages, 34-35
 - server logs, 112-113
 - service components, 158
 - service consumer models,
 - creating in portlet tutorial, 281-286
 - service documents, importing, 135-136
 - service provider models, creating in portlet tutorial, 278-281
 - service providers, 115
 - configuring for WSSM token consumption, 310-317
 - defined, 299
 - deployment to WAS, 323-324
 - enabling security, 309-324
 - service requesters
 - configuring for WSSM token generation, 327-333
 - deployment to WAS, 338-340
 - enabling security, 325-340
 - services
 - connectivity, 5-6
 - consumers, 115
 - creating, 3
 - defined, 2, 155
 - designing, 2-3, 16-18, 21-35, 38-41, 44-51
 - dynamic endpoint
 - selection, 117
 - finding, 116-121, 141-143
 - governing, 9, 116-117
 - implementation types, 156-158
 - integrating, 3, 155-159
 - life cycles, 117
 - managing, 9
 - metadata, 116-117, 122-126
 - publishing, 116-121, 135-140
 - registries, 9

- reusing, 115-116, 128, 141-148
- security, 7, 9
- updating, 148-153
- servlets, portlets versus, 245
- Set Up Deployment Wizard (Enterprise Explorer), 97
- sharing service designs (HTML files), 38-39. *See also* reusing
- single sign-on (SSO), WebSphere Portal, 246
- skins of portals, 245
- snippets, creating visual snippets, 176-180
- SOA (Service-Oriented Architecture)
 - BPM support for, 159
 - Rational Application Developer support for, 59-60
 - RSA support for, 15
- services
 - connectivity, 5-6
 - creating, 3
 - defining, 2
 - designing, 2-3
 - governing, 9
 - integrating, 3
 - managing, 9
 - registries, 9
 - security, 7, 9
- STS support for, 304
- TFIM support for, 304-305
- WebSphere Portal support for, 247
- WebSphere Portlet Factory support for, 247
- WebSphere Application Server support for, 61
- WebSphere Integration Developer support for, 159-160
- WebSphere Message Broker support for, 204-207
- WebSphere Process Server support for, 159
- WSRR support for, 126-127
- WSSM support for, 304
- SOAP API, 121
- SQL, Run SQL Wizard (Rational Application Developer), 79
- starting
 - EAR files, 190-191
 - integrated solutions console, 190
 - WebSphere Process Server, 189
- sticky notes, creating for business processes, 180-181
- stopping WebSphere Application Server, 114
- storing artifacts (services), 157
- structure diagrams, 14
- STS (Security Token Service), 300-303
 - SOA support for, 304
 - trust service chain, configuring
 - for token transformation, 333-338
 - for token validation, 317-322
- system requirements, 307-308
- testing, 341
- SOA support for, 304-305
- trust relationships, establishing, 302-303
- themes of portals, 245
- time zone converter services design tutorial, 16
 - CD-ROM contents, 18
 - class diagrams, creating, 23-32
 - RTE, 40-41, 44-50
 - running, 51
 - sequence diagrams, creating, 33-35, 38
 - service designs, publishing, 38-39
 - service specifications, 17
 - system requirements, 17
 - UML models, creating, 18, 21
 - use case diagrams, creating, 21-22
- Tivoli Access Manager (IBM), 8
- Tivoli Federated Identity Manager. *See* TFIM
- to-do tasks, creating in business processes, 181
- tokens
 - consumption, configuring service providers for, 310-317
 - generation by WSSM, configuring service requesters for, 327-333
 - security tokens, 7-8
 - consuming, 303
 - generating, 302
 - transformation, configuring STS trust service chain for, 333-338
 - validation, configuring STS trust service chain for, 317-322
- transaction security, 7
- transformation nodes, 201

- transformations
 - Java-to-UML
 - transformations, 46-50
 - transformation
 - configurations, 40
 - UML-to-Java
 - transformations, 40-41, 44-46
- troubleshooting
 - datasource NameNotFound
 - errors, 102
 - unresolved database
 - columns, 100
- trust relationships, 299
 - establishing, 302-303
- trust service chain (STS),
 - configuring
 - for token transformation, 333-338
 - for token validation, 317-322

U-V

- UML (Unified Modeling Language)
 - defining, 14
 - diagrams, 14
 - Java-to-UML
 - transformations, 46-50
 - models
 - creating, 18, 21
 - defining, 14
 - RSA support for, 13
 - UML-to-Java
 - transformations, 40-41, 44-46
- unresolved database columns,
 - troubleshooting, 100
- updating
 - JDBC datasources, 110
 - services, 148-153
- URI (uniform resource indicators),
 - integrated solutions console (WebSphere Application Server), 58
- use case diagrams,
 - creating, 21-22

- utility classes,
 - creating, 95-96
- validations of concepts (services), 120
- variables
 - Assign nodes,
 - creating, 170-172
 - creating for business processes, 169-170
- visual snippets,
 - creating for business processes, 176-180

W

- WAS. *See* WebSphere
 - Application Server
- Web 2.0 support (WebSphere Portal), 246
- Web Ontology Language (OWL), 120
- Web projects,
 - creating, 66
- Web services
 - changing endpoints, 192-194
 - clients
 - changing bindings of, 108-109
 - generating, 86
 - creation tutorial, 63-65
 - CD-ROM contents, 62-63
 - creating database tables, 73-76, 79-80
 - creating Java classes, 67-68
 - creating Java objects, 66-67
 - creating Web projects, 66
 - creating Web services, 68
 - system requirements, 62
 - testing Web services, 71
 - WSDL interface, 69
 - deployment tutorial, 63-65
 - changing Web service endpoints, 107-109
 - installing enterprise applications, 105-107
 - port examinations, 110-112
 - server log files, 112-113
 - setting up databases, 104
 - starting integrated solutions console (WebSphere Application Server), 104
 - starting WebSphere
 - Application Server, 104
 - stopping WebSphere
 - Application Server, 114
 - system requirements, 62
 - updating JDBC
 - datasources, 110
 - external Web services
 - assembling, 181-183
 - copying WSDL interface files from, 164
 - importing to workbench, 162
 - invocation tutorial, 80, 84
 - creating JPA entities, 91-95
 - creating utility classes, 95-96
 - creating Web applications
 - for Web service invocation, 87, 90-91
 - creating Web projects, 84
 - exporting projects as EAR files, 103
 - generating Web service clients, 86
 - JDBC configurations in
 - persistence.xml files, 97-98
 - JPA annotations, 82
 - JPA entity classes, 81
 - modifying Web applications, 98-101
 - running Web applications, 101-102
 - running Web applications in browsers, 91
 - troubleshooting
 - datasource NameNotFound
 - errors, 102
 - troubleshooting
 - unresolved database columns, 100

- port information, editing, 109
- Rational Application
 - Developer's support for, 60
 - references, declaring, 63
- Web Service Wizard
 - (Rational Application Developer), 63, 68
 - @WebServiceRef
 - annotation, 64
 - WSIL files, 64
- Web services security
 - management. *See* WSSM
- Web sites for information, 344
- Web user interface (WSRR), 118
 - loading, 131
 - perspectives, 119
 - Query Wizard, 121
- WebSphere Application Server (WAS), 55, 159
 - datasources, creating, 262-266
 - integrated solutions
 - console, 58
 - starting, 104
 - URI, 58
 - Profile Management Tool, 59
 - profiles, 59
 - Rational Application
 - Developer integration, 57
 - security, 59
 - service provider deployment
 - to, 323-324
 - service requester deployment
 - to, 338-340
 - SOA support for, 61
 - starting, 59
 - stopping, 114
 - Web service deployment
 - tutorial, 63-65
 - changing Web service endpoints, 107-109
 - installing enterprise applications, 105-107
 - port examinations, 110-112
 - server log files, 112-113
 - setting up databases, 104
 - starting integrated solutions console
 - (WebSphere Application Server), 104
 - starting WebSphere
 - Application Server, 104
 - stopping WebSphere
 - Application Server, 114
 - system requirements, 62
 - updating JDBC
 - datasources, 110
- WebSphere Business Modeler (IBM), 3
 - RSA support for, 15
- WebSphere Business Service Fabric (IBM), 6
- WebSphere DataPower
 - Integration Appliance X150 (IBM), 6
- WebSphere Enterprise Service Bus (IBM), 6
- WebSphere Integration Developer, 155-159
 - business process integration
 - tutorial
 - assembling and executing business processes, 181-189
 - CD-ROM contents, 161-162
 - creating business processes, 162-181
 - system requirements, 161
 - SOA support for, 159-160
- WebSphere Message Broker (IBM), 6, 199
 - key concepts, 200-201
 - library book search service
 - tutorial
 - associating runtime artifacts, 219-220
 - CD-ROM contents, 210
 - configuring input message
 - parse options, 227
 - configuring library database, 215-218
 - configuring message flow, 225-227
 - creating ESQL for search query, 227-230
 - creating message
 - definitions, 222-225
 - creating message
 - mappings, 232-237
 - defining message flow
 - logic, 230-232
 - setup, 210-215
 - system requirements, 208-209
 - testing message flow, 237-241
 - SOA support for, 204-207
 - WebSphere MQ versus, 199-200
 - WebSphere Message Broker Toolkit, 201
 - associating runtime artifacts, 219-220
 - Broker Administration
 - perspective, 202-203
 - Broker Application
 - Development
 - perspective, 202
 - database configuration, 215-218
 - Debug perspective, 203
 - Plug-In Development
 - perspective, 203
 - WebSphere MQ
 - message queue setup, 213-215
 - WebSphere Message Broker
 - versus, 199-200
 - WebSphere Portal (IBM), 5, 243-247
 - capabilities of, 246-247
 - installing portlets, 286-288
 - logging in, 293-294
 - portlet tutorial
 - CD-ROM contents, 250
 - deployment, 286-296
 - setup, 250-253, 256-266
 - system requirements, 250
 - SOA support for, 247

- WebSphere Portlet Factory (IBM), 5, 243-245
 - deployment configurations, creating, 250-253
 - portlet tutorial
 - CD-ROM contents, 250
 - deployment, 286-296
 - model creation and testing, 270-278
 - project creation, 267-269
 - service consumer model creation, 281-286
 - service provider model creation, 278-281
 - setup, 250-253, 256-266
 - system requirements, 250
 - SOA support for, 247
 - WebSphere Portlet Factory Designer. *See* Portlet Factory Designer
 - WebSphere Process Server, 159
 - business process integration tutorial
 - CD-ROM contents, 161-162
 - deploying module projects to, 185-186
 - deployment, 189-196
 - system requirements, 161
 - SOA support for, 159
 - starting, 189
 - WebSphere Service Registry and Repository. *See* WSRR
 - workbench (Eclipse), importing artifacts, 143-146
 - Web services, 162
 - WS-Security, 300
 - WS-Trust, 300-302
 - WSDL files, 69
 - copying for business processes, 164
 - creating for business processes, 167
 - WSIL (Web Services Inspection Language) files, 64
 - WSRR (WebSphere Service Registry and Repository), 9, 116
 - services
 - dynamic endpoint selection, 117
 - finding, 116-121
 - governance, 117
 - metadata, 116-117, 122-126
 - publishing, 116-121
 - SOA support for, 126-127
 - tutorial
 - administrator setup, 129
 - business model template setup, 130-131
 - CD-ROM contents, 128
 - finding services, 141-143
 - loading classification system, 133-134
 - publishing services, 135-140
 - reusing services, 141-148
 - system requirements, 128
 - updating services, 148-153
 - Web user interface, 118
 - perspectives, 119
 - Query Wizard, 121
 - WSSM (Web Services security management), 302-303
 - SOA support for, 304
 - token consumption, configuring service providers for, 310-317
 - token generation, configuring service requesters for, 327-333
- X–Z**
- XML (Extensible Markup Language)
 - orm.xml files, JPA entity classes, 83
 - persistence.xml files
 - JDBC configurations in, 97-98
 - JPA entity classes, 84