



IBM Software Group

IBM WebSphere® Application Server Feature Pack for Web 2.0

Samples



@business on demand.

© 2008 IBM Corporation
Updated October 1, 2008

This presentation will focus on the IBM WebSphere Application Server Feature Pack for Web 2.0 installation of Samples.

Agenda

- Overview of the samples
- Installing the samples



This presentation will focus on the overview of the Samples, software requirements for the Samples and how to install the Samples.

Samples

- There are seven samples for this feature pack:
 - ▶ Ajax: Hello!
 - ▶ QuoteStreamer
 - ▶ PlantsByWebSphere (Ajax edition)
 - ▶ CourierApp (RPC adapter sample)
 - ▶ HelloWorld (RPC adapter sample)
 - ▶ Feed samples
 - ▶ SOAP sample



The IBM WebSphere Application Server Feature Pack for Web 2.0 contains 7 Samples. Each of the samples shows off specific features of the Feature Pack for Web 2.0.

Ajax: Hello!

- The Ajax: Hello! Sample application is a simple example of how to begin to use Ajax in Java™ EE applications by using the Dojo Toolkit.
- Ajax operations used in the sample without requiring a full page load in the browser:
 - ▶ input field validation
 - ▶ combination box look-ahead
 - ▶ asynchronous data retrieval
 - ▶ dynamic page updating.



The Ajax: Hello! Sample application is a simple example of how to begin to use Ajax in Java EE applications by using the Dojo Toolkit.

The application implementation is simple, making it easy to understand while still performing Ajax operations, such as input field validation, combination box look-ahead, and asynchronous data retrieval, and dynamic page updating. All of aforementioned operations are demonstrated in this sample without requiring a full page load in the browser.

QuoteStreamer sample

- Uses the Web messaging service to simulate stock quotes to a Dojo-enabled client application
- Simulated quotes are generated by a CommonJ timer that publishes messages to a Service Integration Bus topic space
- Web messaging service links a Dojo enabled client to the publish/subscribe function of WebSphere Application Server for messaging
- Communication is through Bayeux protocol



The Quote Streamer Sample application uses the Web messaging Service to simulate stock quotes to a Dojo-enabled client application. Simulated quotes are generated by a CommonJ timer that publishes messages to a Service Integration Bus topic space. The Web messaging Service links a Dojo enabled client to a WebSphere Application Server internal message broker, messaging, for Web-based publication/subscription.

Communication is achieved through the Bayeux protocol. The Dojo Toolkit's cometd client links incoming JavaScript Object Notation(JSON) messages to the Dojo event/topic system for processing. In the QuoteStreamer Sample application, multiple Dojo widgets are specified in a market report summary article in HTML format. These Dojo widgets process incoming stock quote messages and visually indicate stock changes. These visual changes include: update to the current price of a stock, daily stock price change, and daily stock percent price change. When the price of a stock changes, the daily price change and daily percent price change turn green (increase) or red (decrease) before fading back to the normal background color.

QuoteStreamer sample (continued)

- Quotes are generated on the server side at all times.
- Quotes are started by default.
- **Stop stock quote** unsubscribes to all topics on the server and disconnect the bayeux protocol.
- **Resume stock quote**, presubscribes and re-initializes the bayeux protocol connection.



The start/stop button is now at the top of the page, and when the page loads, quotes are started by default. When you hit stop, all topics on the server are unsubscribed to all topics on the server and bayeux protocol is disconnected. The bayeux protocol. Clicking on start again, presubscribes and re-initiates the bayeux protocol connection.

QuoteStreamer sample snapshot

Fictitious Market Report

2:20 pm : **Tech Stock #1** (TS1: 80.2, +0.2, +0.3) continues to gain in heavy trading which drives small gains for the **Tech Index** (TEC: 3021.0, +21.0, +0.7). Investors continue to like the results for the third quarter of Tech Stock #1. In addition, a report was just released showing important gains in the marketshare of the best selling product of Tech Stock #1.

Elsewhere in technology news, **Tech Stock #2** (TS2: 55.2, +0.2, +0.3) continues to fall for the day. Delays in shipping major new software with an earlier outlook downgrade, appear to be weighing heavily on investors' minds. As a result, **Stock Index** (IND: 10110.5, +110.5, +1.1) is down for the day. Elsewhere, the **Alternative Index** (ALT: 5040.1, +40.1, +0.8) is even in moderate trading.

11:23 am : *Spotlight Drug stores* A lot of activity on occurring the drug store front. United States based **Drug Store #1** (DS1: 10.0, 0.0, 0.2) has gained 10% in heavy activity after a positive third quarter result. This news is good for the previously troubled drug store, who has slowly been closing and remodeling stores in an effort to improve their image. With the strong third quarter, Drug Store #1 can concentrate even more on improving the image of their stores. As a result of this news, Drug Store #1 has been upgraded from hold to buy.

In other drug store news, **Drug Store #2** (DS2: 40.0, 0.0, 0.0) has continued their strong earnings as evidenced by their third quarter statement. With the addition of 20 new stores, earnings for the company should continue to grow. In other market news, the **Stock Index** (IND: 10110.5, +110.5, +1.1) continues to gain since the opening bell.

Stock Ind (IND: 10110.5, +110.5, +1.1)

Tech Ind (TEC: 3021.0, +21.0, +0.7)

Alt Ind (ALT: 5040.1, +40.1, +0.8)



The QuoteStreamer Sample page does not refresh and the fading is a Dojo effect. Actual values on the Sample are generated on the server.

The Sample uses Bayeux protocol which is a long polling model to keep the connection open, the connection is open for 30 seconds and every 5 seconds it polls for information. Asynchronous requests keep coming. The client is up and makes request to the server and the connection stays open and data can be interchanged for the 30seconds. Connections request goes to the server and if no data it can wait for a maximum of 30 seconds then it will close after that. When a new request comes to open the line data will be queued. The Sample can receive events quicker when needed.

PlantsByWebSphere sample (Ajax edition)

- An enhancement to the PlantsByWebSphere application which has been provided as a Sample with the WebSphere Application Server
- The application itself is a three-tier J2EE application consisting of Servlets, EJBs, and JSPs communicating with a back-end database
- It has been augmented with Dojo controls to add Ajax style widgets



The **PlantsByWebSphere Ajax Edition** application is an enhancement to The PlantsByWebSphere application which has been provided as a Sample since WebSphere Application Server Version 5.0. The application itself is a three-tier J2EE application which consists of Servlets, EJBs, and JSPs communicating with a back-end database. It has been augmented with Dojo controls to add Ajax style widgets to improve the usability and to highlight the rich potential of Dojo.

PlantsByWebSphere sample

- Typical Web application containing JavaServer™ Pages files, enterprise service beans, and servlets
- Sample represents how a Web 1.0 technology-based application can be enhanced with Web 2.0 technology
- Sample demonstrates how Ajax widgets can be added to a traditional J2EE Web application to improve user interaction without rewriting the server-side application



The PlantsByWebSphere application represents a typical Web application that contains JavaServer Pages files, enterprise service beans, and servlets. The Sample is intended to represent how a Web 1.0 technology-based application can be enhanced with Web 2.0 technology. Specifically, Ajax style controls using the open-source JavaScript-based Dojo Toolkit. The Sample demonstrates how adding Ajax widgets can be added to a traditional J2EE Web application to improve user interaction without rewriting the server-side application. This application contains drag-drop function, scrollable inventory, and enhanced form validation.

PlantByWebSphere sample snapshot

The screenshot displays a web application interface for a plant nursery. At the top, there is a navigation menu with tabs for Home, Flowers, Trees, Vegetables, and Accessories. The current page is titled "Vegetables" and shows a grid of six items: Ornamental Gourd (\$1.5), Grapes (\$49.0), Onion (\$9.0), Pineapple (\$87.0), and Strawberries (\$3.5). Below the grid, there is a detailed view for Strawberries, including a description: "Our brand is known for producing plump, sweet strawberries by the mid-June bucketful. Now you can grow them easily, with relatively little care, due to our patented version. Dental floss not included." and an "Add to cart" button. On the right side, a shopping cart is visible, containing one item: Pineapple (1 unit, \$87.0). The total price is \$87.00. A "Checkout" button is located below the cart. The footer of the page includes a color calibration bar, the word "Samples", and the copyright notice "© 2008 IBM Corporation".

This picture shows one of the pages for PlantsByWebSphere Sample, specifically the vegetables page. Note the shopping cart on the right side and the item preview section. You can drag-and-drop items into the cart or preview them then select to add them to the cart.

CourierApp

- Demonstrates auto population of address details on selecting a zip code or postal code.
- Address lookup invoked using RPC adapter.
- Response can be JSON or XML.

The screenshot shows a web browser window with a menu bar (File, Edit, View, History, Bookmarks, Tools, Help) and a search icon. Below the menu is a form with the following fields and controls:

- Name:
- Zipcode/Postcode:
- House number:
- Line 2:
- Line 3:
- Line 4:
- [Home](#)

At the bottom of the browser window, there is a status bar that says "Done".



CourierApp demonstrates auto population of address details on selecting a zip/postal code. Address lookup service is invoked using the RPC Adapter to get the detailed address. The response can be either JSON or XML format.

HelloWorld (RPC adapter sample)

- Demonstrates
 - ▶ the usage of EJB modules in RPCAdapter
 - ▶ the capabilities of RPCAdapter like handling of XML and JSON responses and specifying validators.
- The application accepts a name from the user and displays a simple message on clicking the "SayHello" button.
- The application uses RPC adapter's XML and JSON services to validate name and display message.
- The functions for validation and message display are implemented in EJB modules.



The HelloWorld Sample application demonstrates the usage of EJB modules in RPCAdapter. It also demonstrates the capabilities of RPCAdapter like handling of XML and JSON responses and specifying validators. The application contains links to access HelloWorld application using JSON or XML. If any of the links are clicked the corresponding page is displayed. The resulting page accepts a name from the user and displays a simple message on clicking the "SayHello" button. The application uses RPC adapter's XML and JSON services to validate name and display message. The functions for validation and message display are implemented in EJB modules.

Feed samples (1)

- The feed samples demonstrate the Atom and RSS support in Abdera.

- ▶ Generate an Atom feed.

- ▶ Read an Atom feed.

- ▶ Read an RSS feed

- ▶ O'Reilly site:

- <http://www.oreillynet.com/pub/feed/1>

- ▶ RSS feed from CNN:

- http://rss.cnn.com/rss/cnn_topstories.rss



The Feed samples demonstrate the Atom and RSS support in Abdera. You can generate or read Atom feeds using a simple example provided. You can also perform a filtered read on the Atom content. For example, choose an Atom feed from the O'Reilly site listed on this slide, or the RSS example from CNN.

Feed samples (continued)

- Samples for Atom Publishing Protocol (APP)
 - ▶ Demonstrate the ability to create, update, delete, and retrieve entries in a feed. Also provides options to retrieve the service document for the deployed Abdera server, and the associated feed.
- Use the default Abdera Servlet, to try out the APP samples.
- The APP samples require HTTP operations to work with the feed entries.



The Samples for Atom Publishing Protocol demonstrate the ability to create, update, delete, and retrieve entries in a feed. They also provide options to retrieve the service document for the deployed Abdera server, and the associated feed. Use the default Abdera servlet to try out the Atom Publishing Protocol samples. These samples require HTTP operations to work with the feed entries.

SOAP sample

- SOAP is a standard XML-based protocol for invoking SOAP based Web services.
- The application implementation is a simple usage of the IBM Extension to the Dojo toolkit for working with SOAP services.
- The sample makes a SOAP call to United States' National Oceanic and Atmospheric Administration (NOAA) weather service to map a United States ZIP code to its corresponding latitude and longitude.



SOAP is a standard XML-based protocol for invoking SOAP based Web services. The application implementation is a simple usage of the IBM Extension to the Dojo toolkit for working with SOAP services. The simplicity of the application makes it easy to understand how to use the extension while still performing an interesting operation. The sample makes a SOAP call to United States' National Oceanic and Atmospheric Administration (NOAA) weather service to map a United States ZIP code to its corresponding latitude and longitude.

Section

Installing the samples



The next section will cover how to install the Samples.

Options for installing a sample

- Eclipse IDE can be used to install a sample application into your application
- Administration console can be used to install a sample application
- Command line can be used to install a sample application
- Rational® Application Developer or other Eclipse based IDE's can be used to install a sample application
- Refer to the sample documentation that comes with the IBM WebSphere Application Server Feature Pack for Web 2.0 for detailed installation instructions



There are a several ways to install the Sample. Eclipse Integrated Development Environment (IDE) can be used to install a Sample application into your application. Another option is installing through the command line. Other tools like Rational application Developer or other Eclipse based IDE's can be used to install a Sample application. For complete details, see the information center or the documentation that is included with the feature pack.

Useful links

- WebSphere Application Server Version 6.1 Information Center
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/welcome_base.html
- Eclipse Tools 3.3 (<http://www.eclipse.org/>), is an open-source community, whose projects are focused on building an open-development platform that includes extensible frameworks, tools and runtime environments for building, deploying and managing software across the life cycle.
- Eclipse Web Tools Platform (WTP 2.5 or higher) extends the Eclipse platform with tools for developing J2EE Web applications:
<http://www.eclipse.org/webtools/>
- The Ajax Toolkit Framework (<http://www.eclipse.org/atf>) provides an extensible framework and exemplary tools for building IDEs.
- Rational Application Developer Version 7.5 is one of the premier tools for developing software solutions and applications:
<http://www.ibm.com/developerworks/rational/products/rad/>



Here are some useful links

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_was_web20fp_Samples.ppt

This module is also available in PDF format at: [../was_web20fp_Samples.pdf](..../was_web20fp_Samples.pdf)



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM Rational WebSphere

A current list of other IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

EJB, J2EE, Java, JavaServer, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

