Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks


All other trademarks are the property of their respective companies.
Contents

About This Document

What You Need to Know ................................................................. vii
e-docs Web Site ........................................................................ vii
How to Print the Document ............................................................ vii
Related Information ...................................................................... viii
Contact Us! ................................................................................ viii
Documentation Conventions ....................................................... ix

Getting Started with Liquid Data

From Multiple Data Sources to a Web Application in 60 Minutes (or Less) ...................... 1
Step 1. Start the Liquid Data Samples Server ........................................ 2
  Windows ............................................................................... 2
  UNIX ................................................................................... 3
Step 2. Start the WebLogic Administration Console ............................... 3
  Starting the Console ............................................................ 3
  Windows ............................................................................... 3
  UNIX ................................................................................... 3
  Logging Into the Administration Console .................................... 4
Step 3. Configure Data Sources ...................................................... 5
  Configuring a Relational Data Source Description ....................... 5
  Configure an XML File Data Source Description ........................ 8
Step 4. Start Data View Builder and Verify Data Sources ........................ 10
Step 5. Construct Your Query .............................................................. 11
  View a Demo ........................................................................... 12
Open Relevant Data Source Schemas ........................................... 12
Select a Target Schema From the Sample Liquid Data Repository .......... 12
Map Source Elements to the Target Schema ..................................... 14
Define Query Conditions ............................................................. 18
  Create a Join Between Wireless : CUSTOMER and Wireless : CUSTOMER_ORDER . 18
  Create a Join between Wireless : CUSTOMER_ID and Broadband : CUSTOMER_ID . 19
Create a Query Parameter ............................................................. 19
Set the Target Namespace ............................................................ 22
Step 6. View and Test Your Query ......................................................... 23
  Entering Test Mode .................................................................. 23
The Automatically Generated Query ................................................ 23
Testing Your Query .................................................................... 25
  Running Your Query ............................................................... 25
Summary .................................................................................... 28
Step 7. Save Your Liquid Data Project ................................................ 28
Step 8. Deploy Your Query .............................................................. 28
Step 9. Creating a Liquid Data Control in WebLogic Workshop ..................... 30
  Creating a Liquid Data Control ................................................... 31
Step 10. Creating a Liquid Data-Powered Web Application ......................... 35
  Generating a Page Flow File ....................................................... 35
Obtaining Preliminary Query Results ............................................... 39
Modifying Page Flow Source ....................................................... 41
Adding Data to the Application ..................................................... 44
  Adding Field Labels ................................................................ 45
  Adding Data Fields .................................................................. 46
Add Repeater Fields For Order Arrays ............................................. 50
About This Document

This document provides quick start steps that explain how to start the BEA Liquid Data for WebLogic server, bring up the WebLogic Server Administration Console and use it to configure some data sources, use the Data View Builder to map source and target schemas and construct a query, and run the query against the server.

What You Need to Know

This document is intended mainly for business users/developers and XML data designers who will be using Liquid Data to design distributed data solutions and construct queries. System Administrators will also find the initial steps useful as a snapshot example of the configuration process.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at http://e-docs.bea.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the Liquid Data documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader.
About This Document

and print the entire document (or a portion of it) in book format. To access the PDFs, open the Liquid Data documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

For more information in general about Java and XQuery, refer to the following sources.

- The Sun Microsystems, Inc. Java site at:
  http://java.sun.com/
- The World Wide Web Consortium XML Query section at:
  http://www.w3.org/XML/Query

For more information about BEA products, refer to the BEA documentation site at:

http://edocs.bea.com/

Contact Us!

Your feedback on the BEA Liquid Data documentation is important to us. Send us e-mail at docsupport@bea.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the Liquid Data documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA Liquid Data for WebLogic 1.0 release.

If you have any questions about this version of Liquid Data, or if you have problems installing and running Liquid Data, contact BEA Customer Support through BEA WebSupport at www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages
# Documentation Conventions

The following documentation conventions are used throughout this document.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface text</strong></td>
<td>Indicates terms defined in the glossary.</td>
</tr>
<tr>
<td>Ctrl+Tab</td>
<td>Indicates that you must press two or more keys simultaneously.</td>
</tr>
<tr>
<td><em>italics</em></td>
<td>Indicates emphasis or book titles.</td>
</tr>
<tr>
<td><strong>monospace text</strong></td>
<td>Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.</td>
</tr>
<tr>
<td><em>monospace italic text</em></td>
<td>Identifies significant words in code.</td>
</tr>
<tr>
<td><strong>monospace boldface text</strong></td>
<td>Identifies variables in code.</td>
</tr>
<tr>
<td><strong>UPPERCASE TEXT</strong></td>
<td>Indicates device names, environment variables, and logical operators.</td>
</tr>
</tbody>
</table>

Examples:
- `#include <iostream.h>`
- `void main ( )
  chmod u+w *
  \tux\data\ap
  .doc
  tux.doc
  BITMAP
  float`
- `String expr`
- `LPT1
  SIGNON
  OR`
<table>
<thead>
<tr>
<th>Convention</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>{ }</td>
<td>Indicates a set of choices in a syntax line. The braces themselves should never be typed.</td>
</tr>
</tbody>
</table>
| [ ]        | Indicates optional items in a syntax line. The brackets themselves should never be typed.  
  *Example:*  
  buildobjclient [-v] [-o name ] [-f file-list]...  
  [-l file-list]... |
| | Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed. |
| ... | Indicates one of the following in a command line:  
  • That an argument can be repeated several times in a command line  
  • That the statement omits additional optional arguments  
  • That you can enter additional parameters, values, or other information  
  The ellipsis itself should never be typed.  
  *Example:*  
  buildobjclient [-v] [-o name] [-f file-list]...  
  [-l file-list]... |
| . . . .    | Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed. |
Getting Started with Liquid Data

You can use this 10-step tutorial guide to learn how to:

- Configure data sources in Liquid Data
- Associate data sources with a target schema
- Generate and run queries in the Data View Builder
- Build a BEA WebLogic Workshop application around Liquid Data queries
- Display formatted query results as part of a web application

**From Multiple Data Sources to a Web Application in 60 Minutes (or Less)**

This chapter illustrates how you can use Liquid Data to create a query that treats multiple disparate data sources as a single data source. It assumes a large Internet company that has grown through mergers or acquisitions and, as a consequence, finds that its real-time order data must be accessed from two very separate data sources: Wireless order data from an RDBMS system; BroadBand order data via an XML document.

As you work through this Order Report example, keep in mind that each action you take generates source code, in some cases significant amounts. When you are done, you will find that your application contains nearly 500 lines of Java code, 97% of which was automatically generated. (For details see “Generated Files and Code” on page 54.)
Getting Started contains numerous illustrations in the form of screen captures. The goal is to enable you to create a query accessing multiple data sources and use that query when building and running a simple but complete web application — in less than 60 minutes!

The 10 steps in building up the Order Report web application are:

- Step 1. Start the Liquid Data Samples Server
- Step 2. Start the WebLogic Administration Console
- Step 3. Configure Data Sources
- Step 4. Start Data View Builder and Verify Data Sources
- Step 5. Construct Your Query (View a Demo)
- Step 6. View and Test Your Query
- Step 7. Save Your Liquid Data Project
- Step 8. Deploy Your Query
- Step 9. Creating a Liquid Data Control in WebLogic Workshop
- Step 10. Creating a Liquid Data-Powered Web Application

Example directories
Examples used in this chapter assume that the Liquid Data 8.1 sample domain is located at:

`<WL_HOME>/samples/domains/liquiddata/`

where `WL_HOME` is the root location of the WebLogic Server 8.1.

You can find many other examples and samples in *Liquid Data by Example*, including the Avitek Customer Self-Service Sample Application which also combines the back-end data retrieval capabilities of Liquid Data with WebLogic Workshop to create a more fully-functional customer self-service web application.

**Step 1. Start the Liquid Data Samples Server**

The Liquid Data Samples server is simply a WebLogic Server 8.1 that contains Liquid Data samples, including a Liquid Data repository. It can be started from Windows or UNIX.

**Windows**

To start the Liquid Data samples server under Windows choose the following menu option:
Step 2. Start the WebLogic Administration Console

Start —— Programs —— BEA WebLogic Platform 8.1 ——
BEA Liquid Data for WebLogic 8.1 SP2 —— Liquid Data Samples —— Samples Server
A terminal window called Samples Server will appear. When the Samples server is ready, you will see Server started in RUNNING mode.

UNIX

To start the Liquid Data samples server under UNIX run the startWeblogic.sh shell command in the following domain:

<WL_HOME>/samples/domains/liquiddata/startWeblogic.sh

When the Samples server is ready you will see Server started in RUNNING mode.

Note: For more information on setting up and starting the Samples server, see Post-Installation Tasks in the Liquid Data Installation Guide. For more information on starting Liquid Data servers in the various preconfigured domains, see Starting and Stopping the Server in the Liquid Data Administration Guide.

Step 2. Start the WebLogic Administration Console

The Liquid Data repository is managed and configured through the Liquid Data node of the WebLogic Administration Console. It is here that data sources are defined, security policies established, and so forth.

Starting the Console

The WebLogic Administration Console can be accessed from Windows or UNIX.

Windows

To start the WebLogic Administration Console under Windows choose the following menu option:

Start —— Programs —— BEA WebLogic Platform 8.1 —— BEA Liquid Data for WebLogic 8.1 SP2 —— Liquid Data Samples —— Admin Console

UNIX

To start the WebLogic Administration Console for the Liquid Data Samples server running on your local machine, enter the following URL in a web browser:

http://localhost:7001/console
Logging Into the Administration Console

Log in to the console by providing the following default username and password for the Samples server:

<table>
<thead>
<tr>
<th>Field</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>system</td>
</tr>
<tr>
<td>Password</td>
<td>security</td>
</tr>
</tbody>
</table>

After you log in, the Console home page is displayed. Note that Liquid Data is the bottom element in the left pane (Figure 2).
Step 3. Configure Data Sources

Figure 2  WebLogic Administration Console

Note: For more information on using the Liquid Data node of the WebLogic Administration Console, see the Liquid Data Administration Guide.

Step 3. Configure Data Sources

This section describes how to use the WebLogic Administration Console to configure the two sample data sources used in the tutorial customer order report:

- A relational database data source description for customer and order information.
- An XML data source description for order information.

Configuring a Relational Data Source Description

A first step in making a data source available to Liquid Data is to register your data source with the Administration Console. To add a description of your RDBMS data source to the Liquid Data Server registry follow these steps:

1. In the left pane, click the Liquid Data node (bottom icon in the left pane; see Figure 2).
2. In the right pane, click the Data Sources tab (under Configuration).

3. Click the Relational Databases tab.

4. Click the Configure a New Relational Database Data Source Description link to display the Configure tab (Figure 4).

Figure 3 Creating a New Relational Database Data Source

5. Fill in the fields for your data source as described in Table 2:
6. Click Create. If you look in the list of relational database sources, you will find your newly created data source definition. (For additional information see Configuring Access to Relational Databases in the Liquid Data Administration Guide.)
Configure an XML File Data Source Description

To create a description of your XML data source, follow these steps:

1. In the left pane, click the Liquid Data node.
2. Click the XML Files tab (next to Relational Databases).
3. Click the Configure a New XML Data Source Description link to access the Configure dialog box (Figure 5).
Step 3. Configure Data Sources

Figure 5  Defining a XML Data Source Definition

4. Fill in the fields for your XML data source as described in Table 3.

Table 3  Liquid Data Data Source Description for My-PB-WL Data Source

<table>
<thead>
<tr>
<th>Field</th>
<th>Value to Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>MyBroadBand-LD-DS</td>
</tr>
<tr>
<td></td>
<td>This is the name of the data source description used to register the XML data source with the Liquid Data server.</td>
</tr>
<tr>
<td>Data File</td>
<td>b-co.xml</td>
</tr>
<tr>
<td></td>
<td>This is the XML data file name. It must correspond to an XML data file located in the Liquid Data server repository. (The example file name provided above resides in the Samples server repository.)</td>
</tr>
<tr>
<td>Schema File</td>
<td>b-co.xsd</td>
</tr>
<tr>
<td></td>
<td>Schema associated with the XML file. The schema you specify here must be in the Liquid Data server repository.</td>
</tr>
<tr>
<td>Namespace URI</td>
<td>Not needed for this basic example.</td>
</tr>
</tbody>
</table>
5. Click Create. If you look in the list of XML data sources, you will find your newly created data source definition. (For additional information see Configuring Access to XML Files in the Liquid Data Administration Guide.)

Step 4. Start Data View Builder and Verify Data Sources

Next, you are ready to start the Windows-based Data View Builder. This is where you will create and test your query. You can start the Data View Builder two ways:

1. Choose:

   Start — Programs — BEA WebLogic Platform 8.1 — BEA Liquid Data for WebLogic 8.1 SP2 — Data View Builder

2. Or you can start the Data View Builder by double-clicking on the file:

   <WL_HOME>\liquiddata\DataViewBuilder\bin\DVBuilder.cmd

   If you are not already connected to the Liquid Data Samples server, enter the following address in the Server URL field in the Welcome... dialog box (Figure 6).

   t3://localhost:7001

   No username or password is needed; leave those fields blank.

   Figure 6  Data View Builder Login Dialog

3. Click the Login button.

   The Data View Builder workspace and tools appear. You can click on the buttons on the left Navigation panel to view the various data sources available from the Liquid Data server to which the Data View Builder is connected.
Step 5. Construct Your Query

The Data View Builder allows you to build queries (and data views) through graphical representations of your data and any query conditions.

You first define your query by dragging and dropping (mapping) elements from source to target schemas. Constants, parameters, and functions are available to fully express a query statement.

The customer order report query you are building assumes that you have customer orders stored in two data sources (Wireless in a relational database and BroadBand in an XML file). The query will return the order information for any particular customer.

To construct the example customer order report query, you need to:

4. Once configured, your data sources are immediately available. In the Sources area:
   - Click Relational Databases to view available RDBMS data sources. This list should include your new relational database data source: MyWireless-LD-DS
   - Click XML Files to view available XML data sources. This list should include your new XML data source: MyBroadBand-LD-DS
Open Relevant Data Source Schemas

Select a Target Schema From the Sample Liquid Data Repository

Map Source Elements to the Target Schema

Define Query Conditions

Create a Query Parameter

View a Demo

Constructing the “Order Query” Demo... If you are looking at this documentation online, you can click the Demo button to view an animated demonstration of how to build up the conditions and create the mappings that are similar to this example. With minor variations, the Order Query demo previews Steps 1-7 of this chapter.

Open Relevant Data Source Schemas

To start building a query, first select the schemas for the data sources you need.

1. Open the data source schemas from the Builder toolbar as follows:
   - In the XML Files area double-click on MyBroadBand-LD-DS to move the associated schema into your work area.
   - Click Relational Databases and double-click on MyWireless-LD-DS to open the associated schema.

   The schemas for each of the data sources appear.

Note: Move data sources around on the workspace as needed.

Select a Target Schema From the Sample Liquid Data Repository

A target schema represents a structure of your query results. (For additional information on target schemas see “Understanding Target Schemas” in the Schemas and Namespaces in Liquid Data chapter of Building Queries and Data Views.)

For this example, use customerOrderReport.xsd as your target schema.

To add the target schema:
Step 5. Construct Your Query

1. Choose the menu command File —> Set Target Schema. This brings up the Liquid Data file browser.

2. Scroll to the Liquid Data Repository, the topmost item in the scroll list.

**Figure 8  Selecting a Target Schema from the Liquid Data Repository**

3. Select `customerOrderReport.xsd` and click Open to set this as your target schema.

4. In the target schema panel of the Data View Builder right-click on the topmost node of the target schema (CustomerOrderReport) and choose Expand to display all the schema elements (Figure 8).

**Note:** Schemas are located in the following Liquid Data repository directory:

```<WL_HOME>/samples/domains/liquiddata/ldrepository/schemas```
At this point, your work area contains two data source schemas and a target schema.

**Map Source Elements to the Target Schema**

The Data View Builder generates queries based on your mapping of source elements to your target schema.

The CustomerOrderReport target schema includes a CUSTOMER_ORDER node for both Wireless and BroadBand. Map CUSTOMER_ORDER elements from each data source to the corresponding CUSTOMER_ORDER elements in the target schema as follows:
Step 5. Construct Your Query

- Drag and drop all elements in the Wireless data source CUSTOMER_ORDER into the corresponding elements in the target schema's wireless_orders: CUSTOMER_ORDER node (Figure 10).

- Drag and drop all elements contained in the BroadBand data source CUSTOMER_ORDER into the corresponding elements in the target schema's wireless_orders: CUSTOMER_ORDER node (Figure 11).

Figure 10  Mapping Elements in Wireless : CUSTOMER_ORDER to the Target Schema
In both cases, you mapped the sub-elements: ORDER_DATE, ORDER_ID, CUSTOMER_ID, SHIP_METHOD, and TOTAL_ORDER_AMOUNT to corresponding elements in your target schema.

**Note:** As you map elements from source to target, descriptions appear in the lower part of the workspace (Figure 13). If you need to delete an incorrect mapping, click in the Mapping ID field to highlight the row you wish to delete. Then click the Trash icon on your lower left.

Use the Wireless data source schema to populate the CUSTOMER element in the target schema.

- Drag and drop the sub-elements contained in the Wireless data source CUSTOMER into the corresponding sub-elements in the target schema CUSTOMER node. As you can see, not all elements in a source schema need to be mapped to the target.
At this point, you have completed the source-to-target mappings and your target schema is ready.

The 16 source-to-target mappings you created appear on the Mappings tab (Figure 13).
Getting Started with Liquid Data

Figure 13  Complete List of Source-to-Target Mappings

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/PRODUCT_TYPE</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/CUSTOMER_ID</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/CUSTOMER_ID</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/CUSTOMER_ID</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/STATE</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/STATE</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/CITY</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/CITY</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/ZIP</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/ZIP</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/CURRENT_ADDRESS</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/CURRENT_ADDRESS</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/PAST_ADDRESS</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/PAST_ADDRESS</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/LAST_NAME</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/LAST_NAME</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/FIRST_NAME</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/FIRST_NAME</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/DATE_OF_BIRTH</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/DATE_OF_BIRTH</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/PHONE_NUMBER</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/PHONE_NUMBER</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/EMAIL_ADDRESS</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/EMAIL_ADDRESS</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/HOME_PHONE</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/HOME_PHONE</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/MOBILE_PHONE</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/MOBILE_PHONE</td>
</tr>
<tr>
<td>[myBroadband-ID-US]/db/CUSTOMER_ORDER/SHORT_NAME</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/SHORT_NAME</td>
</tr>
</tbody>
</table>

Define Query Conditions

The purpose of the query is to retrieve Broadband and Wireless transactions for a specified customer. The following sections describe how to define conditions for the project you are building:

- Create a Join Between Wireless : CUSTOMER and Wireless : CUSTOMER_ORDER
- Create a Join between Wireless : CUSTOMER_ID and Broadband : CUSTOMER_ID
- Create a Query Parameter

Create a Join Between Wireless : CUSTOMER and Wireless : CUSTOMER_ORDER

In the Wireless data source drag and drop the element CUSTOMER : CUSTOMER_ID to CUSTOMER_ORDER : CUSTOMER_ID to create a join between CUSTOMER and CUSTOMER_ORDER:

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>[myWireless-LD-DS]/db/CUSTOMER/CUSTOMER_ID</td>
<td>[myWireless-LD-DS]/db/CUSTOMER_ORDER/CUSTOMER_ID</td>
</tr>
</tbody>
</table>

The syntax of this join appears on the first row of the Conditions tab.

Note: Whenever you create a query condition, the Conditions tab is automatically displayed in the lower part of the workspace (Figure 14).
Create a Join between Wireless : CUSTOMER_ID and Broadband : CUSTOMER_ID

Drag and drop the Wireless element CUSTOMER : CUSTOMER_ID onto the BroadBand data source element CUSTOMER_ORDER : CUSTOMER_ID:

<table>
<thead>
<tr>
<th>Join Element</th>
<th>Join Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>[myWireless-LD-DS]/db/CUSTOMER/CUSTOMER_ID</td>
<td>[myBroadBand-LD-DS]/db/CUSTOMER_ORDER/CUSTOMER_ID</td>
</tr>
</tbody>
</table>

The syntax of this join is displayed on the next row on the Conditions tab.

Figure 14 Query Conditions After Selection of Two Joins

<table>
<thead>
<tr>
<th>Enabled</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[myWireless-LD-DS]/db/CUSTOMER/CUSTOMER_ID eq [myWireless-LD-DS]/db/CUSTOMER_ORDER/CUSTOMER_ID</td>
</tr>
<tr>
<td>2</td>
<td>[myWireless-LD-DS]/db/CUSTOMER/CUSTOMER_ID eq [myBroadBand-LD-DS]/db/CUSTOMER_ORDER/CUSTOMER_ID</td>
</tr>
</tbody>
</table>

Note: If you run the query at this point, your results retrieve information on all customers in your customer database.

Create a Query Parameter

Although you could use a constant to “hard-code” your query for a particular customer, creating a query parameter is a more flexible approach. Using a query parameter, you can retrieve information on any customer in your data set.

1. On the Builder Toolbar (left navigation panel) click the Toolbox tab.
2. Select Query Parameters.
3. You need to name the parameter and specify a type. In this example, use myCustomer as the name and select string from the Type pulldown menu (see Figure 14).
4. Click Create.
5. Next you need to associate your new query parameter with a data element. Do this by dragging the myCustomer query parameter to the CUSTOMER : CUSTOMER_ID element of MyWirelessLD-DS database schema:

<table>
<thead>
<tr>
<th>Join Element</th>
<th>Join Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Query Parameter] myCustomer</td>
<td>[myWireless-LD-DS]/db/CUSTOMER/CUSTOMER_ID</td>
</tr>
</tbody>
</table>

When creating a parameter you need to specify a name... ...

... and a type
Step 5. Construct Your Query

You have now set up all the source conditions needed for your query.

Figure 16  Mapping a Parameter to a Source Schema Element

Figure 17  Complete Set of Conditions
Set the Target Namespace

Although not necessary in all cases, it is good practice to set a namespace for your target schema. See “Understanding XML Namespaces” in the Overview and Key Concepts chapter of Building Queries and Data Views.

To set a target namespace for the customerOrderReport target schema follow these steps:

1. Click on the top node of the target schema (CustomerOrderReport).
2. Select Target Namespace from the Data View Builder Query menu (Query —> Target Namespace).

Figure 18 Specifying a Target Namespace

3. Enter values shown in Table 4 for the sample target namespace prefix and URI (Figure 18).

Table 4 Namespace Prefix and URI Values

<table>
<thead>
<tr>
<th>Field</th>
<th>Defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>custorder</td>
</tr>
<tr>
<td>URI</td>
<td>urn:schemas-bea-com:ld-custorder</td>
</tr>
</tbody>
</table>

4. Click Ok. Your new target namespace setting will appear in the topmost node of your target schema (Figure 19).

Figure 19 Target Schema with New Namespace

5. Using the File —> Save Target Schema command, navigate to the Liquid Data Server repository directory and save your schema to the new filename: custOrdRpt.xsd
Step 6. View and Test Your Query

At this point you have assembled all the query elements. Specifically you have mapped data source elements to your target schema, you have created the necessary joins, and you have specified a query parameter functioning similarly to a SQL WHERE clause.

Entering Test Mode

When you enter Test mode, your query is automatically generated.

Click the Test tab at the top of the workspace (Figure 20). After a few moments your query will automatically appear.

Figure 20  Supplying a Query Parameter

The Automatically Generated Query

The automatically generated XQuery is shown in Listing 5.
Getting Started with Liquid Data

Listing 5  Generated customerOrderReport Query

```xml
namespace custorder = "urn:schemas-bea-com:ld-custorder"

<custorder:CustomerOrderReport>
  {
    for $MyWireless_LD_DS.CUSTOMER_1 in document("MyWireless-LD-DS")/db/CUSTOMER
    where ($#myCustomer of type xs:string  eq
    $MyWireless_LD_DS.CUSTOMER_1/CUSTOMER_ID)
    return
    <customerOrder>
      <CUSTOMER>
        <FIRST_NAME>{ xf:data($MyWireless_LD_DS.CUSTOMER_1/FIRST_NAME) }</FIRST_NAME>
        <LAST_NAME>{ xf:data($MyWireless_LD_DS.CUSTOMER_1/LAST_NAME) } </LAST_NAME>
        <CUSTOMER_ID>{ xf:data($MyWireless_LD_DS.CUSTOMER_1/CUSTOMER_ID) } </CUSTOMER_ID>
        <STATE>{ xf:data($MyWireless_LD_DS.CUSTOMER_1/STATE) } </STATE>
        <EMAIL_ADDRESS>{ xf:data($MyWireless_LD_DS.CUSTOMER_1/EMAIL_ADDRESS) } </EMAIL_ADDRESS>
        <TELEPHONE_NUMBER>{ xf:data($MyWireless_LD_DS.CUSTOMER_1/TELEPHONE_NUMBER) } </TELEPHONE_NUMBER>
      </CUSTOMER>
    <wireless_orders>
      { for $MyWireless_LD_DS.CUSTOMER_ORDER_2 in document("MyWireless-LD-DS")/db/CUSTOMER_ORDER
      where ($MyWireless_LD_DS.CUSTOMER_1/CUSTOMER_ID  eq
      $MyWireless_LD_DS.CUSTOMER_ORDER_2/CUSTOMER_ID)
      return
      <CUSTOMER_ORDER>
        <ORDER_DATE>{ xf:data($MyWireless_LD_DS.CUSTOMER_ORDER_2/ORDER_DATE) } </ORDER_DATE>
        <ORDER_ID>{ xf:data($MyWireless_LD_DS.CUSTOMER_ORDER_2/ORDER_ID) } </ORDER_ID>
        <CUSTOMER_ID>{ xf:data($MyWireless_LD_DS.CUSTOMER_ORDER_2/CUSTOMER_ID) } </CUSTOMER_ID>
        <SHIP_METHOD>{ xf:data($MyWireless_LD_DS.CUSTOMER_ORDER_2/SHIP_METHOD) } </SHIP_METHOD>
        <TOTAL_ORDER_AMOUNT>{ xf:data($MyWireless_LD_DS.CUSTOMER_ORDER_2/TOTAL_ORDER_AMOUNT) } </TOTAL_ORDER_AMOUNT>
      </CUSTOMER_ORDER>
    }}
  </wireless_orders>
  <broadband_orders>
  { }
</custorder:CustomerOrderReport>
```

24  Getting Started
Step 6. View and Test Your Query

Testing Your Query

Once generated, your query is ready to be tested. However, you need to supply the customer ID parameter you set up in “Create a Query Parameter” on page 19.

Click in the Value field next to myCustomer and directly enter CUSTOMER_1, as (see Figure 20).

Note: Query parameters are case sensitive.

Running Your Query

When you are ready to run the query, click the Run query button on the toolbar in the upper left (Figure 21).
Figure 21  Run Query Button

The Liquid Data server processes the query and returns results. Right-click —> Expand on the top line of the generated results (customerOrder) to view your data in an outline-like arrangement.

Figure 22  Test Mode Displaying Query and Result in Tree Format

View as XML Checkbox
If you click the View as XML checkbox in the lower part of the Test view area of the Data View Builder, query results are displayed in XML (see Listing 6).

Listing 6 Query Result in XML

```xml
<prefix1:CustomerOrderReport xmlns:prefix1="urn:schemas-bea-com:ld-custorder">
  <customerOrder>
    <CUSTOMER>
      <FIRST_NAME>JOHN_1</FIRST_NAME>
      <LAST_NAME>KAY_1</LAST_NAME>
      <CUSTOMER_ID>CUSTOMER_1</CUSTOMER_ID>
      <STATE>TX</STATE>
      <EMAIL_ADDRESS>abc@abc.com</EMAIL_ADDRESS>
      <TELEPHONE_NUMBER>4081231234</TELEPHONE_NUMBER>
    </CUSTOMER>
    <wireless_orders>
      <CUSTOMER_ORDER>
        <ORDER_DATE>2002-03-06</ORDER_DATE>
        <ORDER_ID>ORDER_ID_1_0</ORDER_ID>
        <CUSTOMER_ID>CUSTOMER_1</CUSTOMER_ID>
        <SHIP_METHOD>AIR</SHIP_METHOD>
        <TOTAL_ORDER_AMOUNT>1000</TOTAL_ORDER_AMOUNT>
      </CUSTOMER_ORDER>
      <CUSTOMER_ORDER>
        <ORDER_DATE>2002-03-06</ORDER_DATE>
        <ORDER_ID>ORDER_ID_1_1</ORDER_ID>
        <CUSTOMER_ID>CUSTOMER_1</CUSTOMER_ID>
        <SHIP_METHOD>AIR</SHIP_METHOD>
        <TOTAL_ORDER_AMOUNT>2000</TOTAL_ORDER_AMOUNT>
      </CUSTOMER_ORDER>
    </wireless_orders>
    <broadband_orders>
      <CUSTOMER_ORDER>
        <ORDER_DATE>2002-04-09</ORDER_DATE>
        <ORDER_ID>ORDER_ID_1_0</ORDER_ID>
        <CUSTOMER_ID>CUSTOMER_1</CUSTOMER_ID>
        <SHIP_METHOD>AIR</SHIP_METHOD>
        <TOTAL_ORDER_AMOUNT>1000.00</TOTAL_ORDER_AMOUNT>
      </CUSTOMER_ORDER>
      <CUSTOMER_ORDER>
        <ORDER_DATE>2002-04-09</ORDER_DATE>
        <ORDER_ID>ORDER_ID_1_1</ORDER_ID>
        <CUSTOMER_ID>CUSTOMER_1</CUSTOMER_ID>
        <SHIP_METHOD>AIR</SHIP_METHOD>
        <TOTAL_ORDER_AMOUNT>1500.00</TOTAL_ORDER_AMOUNT>
      </CUSTOMER_ORDER>
    </broadband_orders>
  </customerOrder>
</prefix1:CustomerOrderReport>
```
Summary
You have successfully built and run a query using Liquid Data.

Note: To switch back to Data View Builder Design mode, click the Design View tab on the toolbar.

Step 7. Save Your Liquid Data Project
Now that you have successfully built and run a query, you can save your Data View Builder project. The next time you need to run or modify your query or Data View, you can simply reopen the project.

To save your project:
1. Choose the menu option File —> Save Project.

2. Navigate to the directory where you want to save your project. Enter `customerReport` in the File name field.

3. Click Save.

   The file is saved with a `.qpr` extension.

   See “Working With Projects” in the `Starting the Builder and Touring the GUI` chapter of `Building Queries and Data Views`.

Step 8. Deploy Your Query
To use the results of your query in an application such as WebLogic Workshop, you first need to associate a saved query with a saved target schema. This is known as deploying a query and provides the metadata necessary to create an XMLBean that provides data access services in Workshop. (See “Deploying a Query” in the `Testing Queries` chapter of `Building Queries and Data Views`.)

Follow these steps to deploy your query:

1. In Test mode, select Deploy Query from the Query menu (Query —> Deploy Query). A dialog box appears where you can choose to save your current schema to the Liquid Data repository.
Step 8. Deploy Your Query

2. Choose Yes. When the repository file browser appears you can use the supplied target schema name of custOrdRpt.xsd

3. Next, you will be asked if you want to save the current query. Again select Yes.

4. Save your query to the name custOrdRpt.
5. The Deploy Stored Query dialog box appears, populated with the name of your query and target schema.

![Figure 26 Creating a Stored Query](image)

5. Click Deploy. You should see a Deployment Successful! message.

![Figure 27 Deploying custOrdRpt Query with custOrdRpt Schema](image)

6. Click Deploy. You should see a Deployment Successful message.

**Step 9. Creating a Liquid Data Control in WebLogic Workshop**

Once you have deployed a query, you can quickly make it available to WebLogic Workshop. (For details see “Using WebLogic Workshop Controls to Develop Liquid Data Applications” in the Application Developer's Guide.)
In the following two sections you will undertake:

- Creating a Liquid Data Control
- Generating a Page Flow File
- Obtaining Preliminary Query Results
- Modifying Page Flow Source
- Adding Data to the Application
- Testing Your Results

Creating a Liquid Data Control

A Liquid Data control provides access from WebLogic Workshop to queries that have been deployed to a Liquid Data server.

1. Open WebLogic Workshop from the Windows Start menu:
   
   Start —> Programs —> BEA WebLogic Platform 8.1 —> WebLogic Workshop 8.1

2. Create a new WebLogic Workshop application (File —> New —> Application). Use custOrdRpt as the application name. Notice that the selected application type is Default Application.
When you create a default application, a project of type Web Project is automatically built using the name of the application plus Web. In this case the project is custOrdRptWeb.


4. Right-click on the custOrdRpt folder and select New —> Java Control.
5. Select Liquid Data from the list of available Java control extensions.

Enter `custOrdRpt` as the filename for the control (Figure 31). The Liquid Data control enables you to turn Liquid Data queries into XML Beans which can be used by a WebLogic Workshop application.

Figure 30  Creating a New Java Control

![Creating a New Java Control](image)

Figure 31  Selecting and Naming a Liquid Data Control

![Selecting and Naming a Liquid Data Control](image)
6. Click Next. If your server is remote, you will need to enter remote connection information. However, most likely Liquid Data samples are running on your local system; if so, select Local, the default option.

Figure 32 Creating a New Java Control Dialog

![New Java Control Extension - Liquid Data](image)

**STEP 1**
- New VCName: custOrdRpt

**STEP 2**
- **Local**
- **Server URL (e.g., localhost):**
- **Username:**
- **Password:**

Click Connection

7. Click Create.

8. From the list of available queries select custOrdRpt.

Figure 33 Adding a Deployed Query to a Liquid Data Control

![Select Liquid Data Queries](image)
Step 10. Creating a Liquid Data-Powered Web Application

9. Click the Add button.

10. When the query moves to the right-hand column, click Finish. A graphical depiction of the newly created Liquid Data control is shown in Figure 34.

Figure 34  Design View of Liquid Data Control

The final section takes you through the steps required to build a web application that display query results in automatically-generated java server pages (JSP).

Step 10. Creating a Liquid Data-Powered Web Application

Once a Liquid Data Control is available, its query elements can be built into JSP pages through creation of a WebLogic Workshop application.

Generating a Page Flow File

WebLogic Workshop applications are driven by a single page flow file, identified by its .jpf extension. When you generate a page flow, WebLogic Workshop creates the page flow file and several other files. (For details see “Generating a Page Flow From a Control” in the chapter Using Workshop Controls to Develop Liquid Data Applications in the Application Developer’s Guide.)

Perform the following steps to generate a page file (custOrdRpt.jpf) from a Liquid Data java control.

2. Select Generate Page Flow (Figure 35).

**Figure 35 Generating a Page Flow Document From a Liquid Data Java Control**

3. In the Page Flow Wizard, enter `custOrdRpt` as the page flow name.
4. Click Next.
5. In the Page Flow Wizard - Select Actions dialog box check the query methods you want available to your new application. In the case of `custOrdReport`, there only one method. Click the checkbox to the left of the method (Figure 36).
6. Select Create.
Step 10. Creating a Liquid Data-Powered Web Application

**Figure 36 Identifying the Liquid Data Methods for the Page Flow**

WebLogic Workshop generates:

- An application page flow file (`custOrdRptController.jpf`)
- A start page (`index.jsp`)
- A JSP file for method(s) you specified through the Liquid Data control (`custOrdRpt.jsp`).
Figure 37  WebLogic Workshop View After a Page Flow is Generated

The application Flow View graphically depicts the relationship between the various pages of your application.

7. Delete the *success* link that connects the *begin* element to *index.jsp* by selecting the link (click on it) and then choose Right-click —> Delete (Figure 38). The change is needed because *index.jsp* will not be the application's start-up page.

Figure 38  Removing a Page Flow Link
8. Create a new link from begin to custOrdRpt.jsp; click adjacent to begin. Then, holding your mouse button down, draw a link to custOrdRpt.jsp (Figure 39). It will automatically be labeled success.

Based on the XMLBean that was created from the custOrdRpt deployed query, a customer ID parameter is required by the web application. In WebLogic Workshop the new initial application page (custOrdRpt.jsp) requests a Customer ID. (See “Create a Query Parameter” on page 19.)

**Figure 39  Creating a New Page Flow Link**

Obtaining Preliminary Query Results

You can now test your application for data access.

1. From the WebLogic Workshop menu Choose Debug —> Start Without Debugging.

2. After some moments the custOrdRptController page appears. Enter a query parameter such as CUSTOMER_3 (Figure 40).

   **Note:** Case matters; customer_3 will not succeed.
3. Click Submit to test your query. After a few moments raw results such as is shown in Figure 41 appears.
Modifying Page Flow Source

While the application's Java Page Flow (JPF) file is automatically generated, a few lines of Java code are needed to add XMLBean variables to your page flow. (For details see the topics “Adding XMLBean Variables to the Page Flow” and “To Initialize the Variable in the Page Flow” in the chapter Using Workshop Controls to Develop Liquid Data Applications in the Application Developer's Guide.)

Follow these steps to copy several lines of Java code into the correct section of custOrdRptController.jpf:

1. Open the custOrdRptController.jpf file in WebLogic Workshop.
2. Click the Source View tab in WebLogic Workshop.
3. Immediately after the following line (near the beginning of the file):
   
   ```java
   private custOrdRpt myControl;
   ```

   insert the following three lines of java code:

   ```java
   ```

   These instructions create three variables. Two represent the CUSTOMERORDER[] arrays for BroadBandOrders and WirelessOrders data sources, respectively. The third, Customers, represents data associated with a particular customer.

4. A few lines down locate the section of code that ties the query to the custOrdRpt form. An easy way to do this is to search. From the WebLogic Workshop menu select Edit —> Find (or Control-F). Search for a section of code beginning with the string:

   ```java
   Forward custOrdRpt
   ```

   After the line:

   ```java
   getRequest().setAttribute( "results", var )
   ```

   enter (or copy/paste) the following:
Getting Started with Liquid Data

Wireless = var.getCustomerOrderReport().getCustomerOrderArray(0).getWirelessOrders().getCUSTOMERORDERArray();
BroadBand = var.getCustomerOrderReport().getCustomerOrderArray(0).getBroadbandOrders().getCUSTOMERORDERArray();
Customers = var.getCustomerOrderReport().getCustomerOrderArray(0);

These instructions make the three public variables available to the application. The relevant code is shown in Figure 7 (newly entered lines are in boldface).


package custOrdRpt;
import com.bea.wlw.netui.pageflow.FormData;
import com.bea.wlw.netui.pageflow.Forward;
import com.bea.wlw.netui.pageflow.PageFlowController;
import custOrdRpt.custOrdRpt;
... 
public class custOrdRptController extends PageFlowController 
{
  /**
   * This is the control used to generate this pageflow
   * @common:control
   */
  private custOrdRpt myControl;

  public

  public

  public

  // Uncomment this declaration to access Global.app.
  //
Step 10. Creating a Liquid Data-Powered Web Application

protected global.Global globalApp;

// For an example of page flow exception handling see the example "catch" and "exception-handler"
// annotations in {project}/WEB-INF/src/global/Global.app

/**
 * This method represents the point of entry into the pageflow
 * @jpf:action
 */
protected Forward begin()
{
    return new Forward( "success" );
}

/**
 * Action encapsulating the control method :custOrdRpt
 * @jpf:action
 * @jpf:forward name="success" path="index.jsp"
 * @jpf:catch method="exceptionHandler" type="Exception"
 */
public Forward custOrdRpt( CustOrdRptForm aForm )
throws Exception
{
    schemasBeaComLdCustorder.CustomerOrderReportDocument var = myControl.custOrdRpt( aForm.myCustomer );
    getRequest().setAttribute( "results", var );

    Wireless = var.getCustOrderReport().getCustomerOrderArray(0).getWirelessOrders().getCUSTOMERORDERArray();
    BroadBand = var.getCustOrderReport().getCustomerOrderArray(0).getBroadbandOrders().getCUSTOMERORDERArray();
    Customers = var.getCustOrderReport().getCustomerOrderArray(0);

    return new Forward( "success" );
}

/**
 * @jpf:action
 * @jpf:forward name="success" path="custOrdRpt.jsp"
 */
public Forward custOrdRptLink()
throws Exception
{
    return new Forward( "success" );
}
Getting Started with Liquid Data

```java
/**
 * @jpf:exception-handler
 * @jpf:forward name="errorPage" path="/error.jsp"
 */
protected Forward exceptionHandler( Exception ex, String actionName, String message, FormData form )
{
    String displayMessage = "An exception occurred in the action " + actionName;
    getRequest().setAttribute("errorMessage", displayMessage);
    return new Forward("errorPage");
}
/**
 * FormData class CustOrdRptForm
 * FormData get and set methods may be overwritten by the Form Bean editor.
 */
public static class CustOrdRptForm extends FormData
{
    private java.lang.String myCustomer;
    public void setMyCustomer( java.lang.String myCustomer )
    {
        this.myCustomer = myCustomer;
    }
    public java.lang.String getMyCustomer()
    {
        return myCustomer;
    }
}
```

Adding Data to the Application

Now that variables are available to hold query results, the last step is to modify `index.jsp` to display your results. In the `custOrdRpt` folder double-click on `index.jsp` (Figure 42).
Three things need to be added to `index.jsp`:

- Field labels.
- Data fields, to display information about the current customer.
- NetUI `repeater` fields, to contain data from the BroadBand and Wireless data sources.

The following sections walk you through the `index.jsp` editing process.

**Adding Field Labels**

First, several labels need to be added to `index.jsp`.

Go to the source view. Open the `index.jsp` file. Near the end of that file find and replace the long line beginning with:

```jsp
<%Object res = request.getAttribute ...
```

With the following:

```jsp
<b>CUSTOMER</b>
<br>
<b>BroadBand Orders</b>
<br>
<b>Wireless Orders</b>
<br>
```
Adding Data Fields

Next, a HTML table needs to be added to the page to hold customer information (in an RDBMS application, customer information would represent the *master* and customer orders the *detail*).

1. Return to Design View.
2. Using your mouse, click on CUSTOMER (Figure 43).

**Figure 43 Java Server Page (index.jsp) After Adding Field Labels**

3. From the WebLogic Workshop menu select Insert —> HTML —> Table.
4. When the TableWizard appears, set the number of rows to 4 and the number of columns to 2.

**Figure 44 Creating a HTML Table in WebLogic Workshop**
5. Click Ok. Notice that the table is placed in the Results Area below the CUSTOMER label.

6. Double-click in the upper-left field of the generated grid (Figure 45). This will open to the relevant section of index.jsp in Source View.

**Figure 45 Selecting a Cell in WebLogic Workshop Table Grid**

7. Replace the first table detail of each row with the following descriptions: First name, Last name, Customer ID, and Email Address. Replacement text is highlighted (Figure 48).

```
<table border="1">
<tr>
  <td>First Name</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>Last Name</td>
  <td>&nbsp;</td>
</tr>
<tr>
  <td>Customer ID</td>
  <td>&nbsp;</td>
</tr>
```

Getting Started  47
<table>
  <tr>
    <td>Email Address</td>
    <td>&nbsp;</td>
  </tr>
</table>

8. Switch back to Design View.

9. In the WebLogic Workshop Data Palette, expand PageFlow —> Properties —> Customers —> CUSTOMER to see the available Customer elements (Figure 46).

**Figure 46 Data Palette Page Flow Data Components**

10. Drag-and-drop data elements to their corresponding text labels (Figure 47).
Figure 47  Populating WebLogic Workshop HTML Table with Customer Data Element

Figure 48 shows a fully configured customer information table.
Add Repeater Fields For Order Arrays

WebLogic Workshop repeater fields make it very easy to insert arrays of data such as Wireless and BroadBand order information. (For details see the topic “To Add a Repeater to a JSP File” in the chapter Using Workshop Controls to Develop Liquid Data Applications in the Application Developer’s Guide.)

Using your cursor, highlight the return after BroadBand (Figure 49).

1. From the Data Palette, just above the Public Controls section, drag the BroadBand array symbol to the highlighted area (also Figure 49).
Figure 49 Populating a JSP With an Array Containing BroadBand Order Data Elements

2. Accept the defaults (meaning all the elements in the array) by clicking Next. No title is needed.
Figure 50 Selecting Fields for NetUI Repeater Element

3. Click Create to accept the default repeater data format of Table.

Figure 51 Setting Format for NetUI Repeater Element

4. Repeat Steps 1 and 2 for Wireless, dragging the Wireless array icon just to the right of the Wireless label. As necessary, drag-and-drop text labels so they appear in the right location.
Testing Your Results

Congratulations. Your application is completed. To test it:

1. Run your application using the Start Without Debugging command.
2. When the initial screen appears enter, in uppercase, a valid customer ID such as CUSTOMER_3.
   When finished, your results page contain the same data as is shown in Figure 52.

Figure 52 Results Page After Query is Run

Summary

This concludes the Liquid Data Quick Start tutorial. If you followed along with the instructions, you successfully:

- identified disparate data sources
- Created a parameterized query
**Getting Started with Liquid Data**

- Deployed your query to the Liquid Data server
- Created a Liquid Data control
- Used a query from the control in a web application

**Generated Files and Code**

All the files associated with the Liquid Data query and web application are generated automatically. Table 8 provides some details. Notice that of the nearly 500 lines of code required to produce this rather simple application, only 13 lines were created by hand.

<table>
<thead>
<tr>
<th>Generated File</th>
<th>Source / Purpose</th>
<th>Total lines of code</th>
<th>Hand coded</th>
</tr>
</thead>
<tbody>
<tr>
<td>custOrdRpt.xq</td>
<td>Data View Builder</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>custOrdRpt.jcx</td>
<td>Liquid Data control, includes target schema</td>
<td>96</td>
<td>0</td>
</tr>
<tr>
<td>custOrdRptController.jspf</td>
<td>WebLogic Workshop Page Flow Control File</td>
<td>180</td>
<td>6</td>
</tr>
<tr>
<td>index.jsp</td>
<td>Web application home page</td>
<td>121</td>
<td>7</td>
</tr>
<tr>
<td>custOrdRpt.jsp</td>
<td>Web application input parameter page</td>
<td>23</td>
<td>0</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td>469</td>
<td>13</td>
</tr>
</tbody>
</table>

**Topics of Further Interest**

Now that you are somewhat familiar with the basics of creating Liquid Data queries using the Data View Builder, the following may be of increased interest:

- *Liquid Data by Example* contains numerous samples and examples beginning starting with a comprehensive survey of Liquid Data Retail Sample Application. That Sample illustrates the data retrieval capabilities of Liquid Data with WebLogic Workshop to create a customer self-service web application.
Step 10. Creating a Liquid Data-Powered Web Application

- Liquid Data Application Developer's Guide provides detailed instructions on using results from Liquid Data queries in WebLogic Workshop, through Enterprise Java Beans (EJBs), and in JSP clients.

- Liquid Data Concepts Guide provides an end-to-end overview of the product including a survey of the product's capabilities, architecture and role in a WebLogic-centered application development environment.

- Building Queries and Data Views provides a detailed GUI overview and description of Data View Builder features.

- The Liquid Data Administration Guide provides information about how to configure the data source types supported by Liquid Data, how to configure security, how to set up monitoring and reporting.
Index

A
Administration Console
logging in 4
starting 3
URL 3
where to find more information 55

C
conditions, creating 18
customer support contact information viii

D
Data View Builder
starting 10
URL 10
demo on how to construct query 12
Design tab 11
documentation, where to find it vii

G
generated XQuery, viewing on the Test tab 23

L
logging in to the Administration Console 4

P
printing product documentation vii

R
related information viii
relational database
configuring as Liquid Data data source 5
running a query 25

S
Samples server, starting 2
source schemas
mapping elements to target schema 14
opening/adding to a project 12
starting
Administration Console 3
Data View Builder 10
Samples server 2
support
technical viii

T
target schema
configuring 14
mapping 14
testing a query 23

U
URL
Administration Console 3
Data View Builder 10
XML file, configuring as Liquid Data data source 8
XML query result, viewing on Test tab 27
XQuery, viewing on Test tab 23