

Building a Report with a Barcode

Figure 40–1 Barcode JavaBean Web report output

SHIPPING INFORMATION	
FROM	Your Company Inc. 100 Evergreen Terrace Springfield, OH 34324
TO	Harrison Sutherland 6445 Bay H Indianapol , IN 46254 United Sta
Shipment Tracking Number	
1042354US	
	
1042354US	
Package Origin Scan	Package Destination Scan
34324-OH-US	46254-IN-US
	
34324-OH-US	46254-IN-US

Reports Builder enables you to create any type of report that displays barcodes. By using the Oracle Reports barcode JavaBean, you can build reports for the Web or for paper that display a barcode to make tasks like tracking shipping orders and employee identification numbers easier. In Reports 6i, you had to use a barcode font to generate the barcode. In Oracle Reports Builder, the JavaBean automatically generates the barcode for you.

To learn more about the barcode JavaBean, visit the Oracle Technology Network (<http://otn.oracle.com/products/reports/>), then click **Getting Started with Oracle Reports** and click **PL/SQL-Java Bridge** in the navigation bar.

Example Scenario

You will build two reports in this section, one for paper and one for the Web. The paper report shows an invoice for a single customer who has ordered multiple items from a company. The barcode indicates the tracking information for the order.

To build either of these reports, you must first refer to [Section 40.1, "Prerequisites for this example"](#).

Table 40–1 Features Demonstrated in this example

Feature	Location
Use the Java importer to add the barcode JavaBean for a paper report.	Section 40.2.1, "Import the Java classes into Reports Builder"
Use the Program Unit editor to create a PL/SQL package for a paper report.	Section 40.2.2, "Create a package to store your information"
Create a Before Report trigger to set up your barcode JavaBean for a paper report.	Section 40.2.3, "Create a Before Report trigger"
Use the Data Model view and toolbar to create a data model with a formula column for a paper report.	Section 40.2.4, "Create a data model with two formula columns"
Create a simple JSP-based Web report.	Section 40.3.1, "Create a query in an existing HTML file"
Create formula columns to call the barcode data for your Web report.	Section 40.3.2, "Create three formula columns in your data model"
Edit the JSP code in the Web source view.	Section 40.3.3, "Initialize the barcode JavaBean and set its properties"
View your JSP-based Web report in a browser.	Section 40.3.4, "Run your report to the Web"

40.1 Prerequisites for this example

To build the examples in this manual, you must have the example files we've provided, as well as access to the sample schema that is shipped with the Oracle9i database.

40.1.1 Example files

If you haven't already done so, you can download the files you'll need to complete this example from the Oracle Technology network and install them on your machine.

To download and install the example files:

1. Go to the Oracle Technology Network Web site (<http://otn.oracle.com/product/reports/>).
2. Click **Getting Started with Oracle Reports**.
3. Click **Index**, then find the "Building a Paper Report with a Barcode JavaBean" example and "Building a Web Report with a Barcode JavaBean" example. To complete this chapter, you need both sets of files. Note that this chapter covers how to build both a paper report and a Web report with a barcode bean.
4. Download the files `BarcodePaper.zip` and `BarCodeWeb.zip` into a temporary directory on your machine (e.g., `d:\temp`).
5. Unzip the contents of the file, maintaining the directory structure, into an examples directory on your machine (e.g., `d:\orawin90\examples`).

This zip file contains the following files:

Table 40–2 Files necessary for building the barcode JavaBean sample reports

File	Description
<code>Examples\BarCodeBeanPaper\result\ShippingManifest.pdf</code>	The final PDF version of the paper report, containing the barcode.
<code>Examples\BarCodeBeanPaper\scripts\oraclebarcode.jar</code>	The barcode JavaBean.
<code>Examples\BarCodeBeanPaper\scripts\barcode_code.txt</code>	All the code used in this chapter, so you can copy and paste the code from this file instead of typing it manually.
<code>Examples\BarCodeBeanPaper\source\ShippingManifest.rdf</code>	The source file for the sample paper report. Running this RDF in Reports Builder will display the final result of your paper report in the Paper Design view.
<code>Examples\BarCodeBeanWeb\result\ShippingManifestWeb.jsp</code>	The final JSP version of the Web report, containing the barcode.

Table 40–2 Files necessary for building the barcode JavaBean sample reports

File	Description
<i>Examples</i> \BarCodeBean Web\result\ShippingM anifestWeb.rdf	The final RDF version of the Web report, containing the barcode.
<i>Examples</i> \BarCodeBean Web\result\assets	The images that Oracle Reports generated when the JSP was run.
<i>Examples</i> \BarCodeBean Web\scripts\SQL.txt	The SQL for the query you need to enter.
<i>Examples</i> \BarCodeBean Web\scripts\barcode_ code.txt	All the code used in this chapter, so you can copy and paste the code from this file instead of typing it manually.
<i>Examples</i> \BarCodeBean Web\source\ShippingL abel.html	The HTML page that you will use as a basis for the Web report.
<i>Examples</i> \BarCodeBean Web\source\ShippingM anifestWeb.rdf	The source file for the sample Web report. Running this RDF in Reports Builder to the Web will display the final result of your Web report in your browser.
<i>Examples</i> \BarCodeBean Web\source\assets*	The images and other files that your JSP-based Web report will require to display properly on the Web.
<i>Examples</i> \BarCodeBean Web\source\assets\BL AFbeige_logo.gif	The image you will use in your JSP-based Web report.

40.1.2 Access to the sample schema

If you don't know if you have access to the sample schema provided with the Oracle9i database, contact your database administrator. You should have access to the "Order Entry" portion of the schema to complete this example.

40.1.3 Update the REPORTS_CLASSPATH environment variable

Before you use a Reports JavaBean (for paper or the Web), you need to perform several steps. You first need to set up your environment to use the correct classpath for the bean. For a paper report, you must then use the Java Importer to import the JavaBean into Reports Builder. For a Web report, you must call the JavaBean from your JSP-based (JavaServer Page) report.

In this section, you will update the Reports class path with the location of the JavaBean. When you launch Reports Builder, it will use this new class path to recognize the location of the barcode bean.

Note: You must follow the steps in this section before you can even run the finished report we've provided, called `ShippingManifestPaper.rdf` and `ShippingManifestWeb.rdf`.

1. Find the class path:
 - In Windows, open the registry using `regedit` (you may want to export a backup before you modify your registry) and update the `REPORTS_CLASSPATH` environment variable.
 - On UNIX, update the `REPORTS_CLASSPATH` environment variable.
2. Modify the class path to reflect the location of the `oraclebarcode.jar` file, for example:

```
ORACLE_HOME/Examples/BarcodeBeanPaper/Scripts/oraclebarcode.jar;  
or
```

```
\ora9ids\reports\j2ee\reports_  
ids\web\examples\BarcodeBeanPaper\scripts\oracleb  
arcode.jar
```

Note: The path depends on where the files are located. If you downloaded the files from the Oracle Technology Network (<http://otn.oracle.com/products/reports/>), you may have to further edit this path. Also, when you edit the Windows registry, make sure you separate your entry with a semicolon (;), with no spaces on either side of the semicolon.

3. Save the new environment variable or exit the Windows registry.

You are now ready to begin building your report.

40.2 Create a barcode report for paper

In this section, you will create a paper-based report that shows the invoice for a particular customer. This invoice will display the address of the customer, his order, and a barcode that represents the tracking number for the order. The company can scan this barcode to find out the status of the order.

Next, you will import the JavaBean, then create a barcode report for paper (Acrobat PDF). If you want to learn how to create a barcode JSP-based report for the Web, skip to [Section 40.3](#).

40.2.1 Import the Java classes into Reports Builder

To create a paper report using the barcode JavaBean, you must first import two Java classes into Reports Builder. When you import these Java classes, Reports Builder automatically creates the packages you need to build the report.

Note: You do not need to perform this task if you're creating a Web report, as you will write JSP that calls the JavaBean.

To import the Java classes:

1. Launch Reports Builder

Note: You must launch Reports Builder now so that the new classpath is used.

2. Close the Welcome dialog box by clicking **Cancel**.
3. Choose **Program > Import Java Classes** to display the Import Java Classes dialog box.
4. Under Select Java Classes, navigate to:

```
oracle.apps.barcode.util.BarCodeConstants.
```

Note: If you do not see this class listed, try exiting Reports Builder and make sure the `REPORTS_CLASSPATH` reads correctly. Then, launch Reports Builder again.

5. Select the class, then click **Import**.
6. Once the packages have been created, import the second JavaBean: `oracle.apps.barcode.BarCodeMaker`.
7. Click **Close**.
8. In the Object Navigator, under the report named **MODULE 1**, click the **Program Units** node. You'll notice that Reports Builder created two package specs and two package bodies named **BARCODECONSTANTS** and **BARCODEMAKER**.

40.2.2 Create a package to store your information

In this report, you want to create a package where the information will be stored.

To create a package for storing your information:

1. In the Object Navigator, under your new report, click the **Program Units** node.
2. Click the Create button in the toolbar to display the New Program Unit dialog box.
3. In the New Program Unit dialog box, type `globals`.
4. Select **Package Spec**, then click **OK** to display the PL/SQL Editor:
5. In the PL/SQL Editor, type the following code:

```
PACKAGE globals IS
    bcoobj ora_java.jobobject;
    barcode_to_use varchar2(256);
    tempdir varchar2(100);
    directory_sep varchar2(2);
END;
```

Note: You can also enter this code by copying and pasting it from the provided text file called `barcode_code.txt`.

6. Click **Compile** to make sure there are no errors in your code.

Note: If your code does not compile, make sure you've typed in exactly the code we've provided.

7. Once the code is compiled, click **Close**.
8. In the Object Navigator, click your report name (e.g., **MODULE 1**).
9. Choose **File > Save**.
10. Name the file `shippingmanifest_<your initials>.rdf` (e.g., `shippingmanifest_vw`) and make sure you save it to a new directory (e.g., `My Examples`), where your new files are stored. Make sure you save the file in RDF format.
11. Click **Save**.

You have created a package that will contain the global information for your report.

40.2.3 Create a Before Report trigger

You can use the Before Report trigger to initialize specific tasks that will run before the report runs. Here, you will define the type of barcode you want to use in your report, as well as the temporary directory where your barcode images will be stored.

To create a Before Report trigger:

1. In the Object Navigator, under **SHIPPINGMANIFEST_<your initials>**, expand the **Report Triggers** node, then double-click the icon next to **BEFORE REPORT** to display the PL/SQL Editor
2. In the PL/SQL Editor, use the template to enter the following PL/SQL code:

```
function BeforeReport return boolean is
begin
    globals.barcode_to_use := BarCodeConstants.BAR_CODE_128;
    globals.bcobj := barcodemaker.new();
return (TRUE);
end;
```

To modify the type of barcode you want to use, you can change the value `BarCodeConstants.BAR_CODE_128` to any other valid value. To determine which values are valid, check the contents of the package by opening the `BarCodeConstants` package spec in the Object Navigator, under the **Program Units** node.

Note: You can enter this code by copying and pasting it from the provided text file called `barcode_code.txt`.

3. Click **Compile** to make sure there aren't any errors.

Note: If you have errors, make sure you've imported the necessary Java classes and that your code matches the code above. If you change the code, be sure to compile it again

4. When the code is compiled, click **Close**. Notice how the node icon next to the **BEFORE REPORT** trigger has changed.
5. Save your report.

You have created a trigger that will set up the barcode type for you when you run the report.

40.2.4 Create a data model with two formula columns

In this section, you will manually create the query that the report will use to retrieve data from the sample schema. You will also create a formula column that will communicate with the JavaBean to create the barcode, then return the file name of the generated image.

To create the query:

1. In the Object Navigator, under **SHIPPINGMANIFEST_<your initials>**, double-click the view icon next to the **Data Model** node to display the Data Model view for your report.
2. In the the Data Model view, click the SQL Query tool in the tool palette, then click in an open area of the Data Model view to display the SQL Query Statement dialog box.
3. In the **SQL Query Statement** field, type (or paste) the following code:

```
SELECT ALL CUSTOMERS_A1.CUST_FIRST_NAME,
        CUSTOMERS_A1.CUSTOMER_ID, CUSTOMERS_A1.CUST_LAST_NAME,
        CUSTOMERS_A1.CUST_ADDRESS.STREET_ADDRESS,
        CUSTOMERS_A1.CUST_ADDRESS.POSTAL_CODE,
        CUSTOMERS_A1.CUST_ADDRESS.CITY,
        CUSTOMERS_A1.CUST_ADDRESS.STATE_PROVINCE,
        CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID, ORDERS.ORDER_ID,
        ORDERS.ORDER_DATE,
        ORDERS.ORDER_TOTAL, ORDER_ITEMS.LINE_ITEM_ID,
        PRODUCTS.PRODUCT_NAME,
        ORDER_ITEMS.UNIT_PRICE, ORDER_ITEMS.QUANTITY,
```

```
COUNTRIES.COUNTRY_NAME
FROM CUSTOMERS CUSTOMERS_A1, ORDER_ITEMS, ORDERS,
PRODUCTS, HR.COUNTRIES
WHERE ((ORDER_ITEMS.ORDER_ID = ORDERS.ORDER_ID)
AND (ORDERS.CUSTOMER_ID = CUSTOMERS_A1.CUSTOMER_ID)
AND (ORDER_ITEMS.PRODUCT_ID = PRODUCTS.PRODUCT_ID)
AND (CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID =
HR.COUNTRIES.COUNTRY_ID))
AND ORDERS.ORDER_ID = :P_ORDER_ID
ORDER BY order_ID, line_item_ID
```

Note: You can enter this query in any of the following ways:

- Copy and paste the code from the provided text file called `barcode_code.txt` into the **SQL Query Statement** field.
 - Click **Query Builder** to build the query without entering any code manually.
 - Type the code in the **SQL Query Statement** field.
-
-

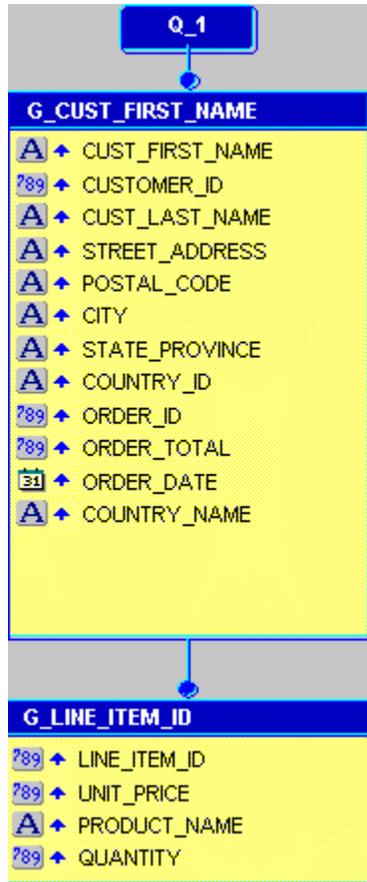
4. Click **OK**.

If you are not connected to a database that contains the sample schema we've provided, you must log in now. If you're not sure what your connection string is, contact your database administrator. Note that this example uses the "Order Entry" portion of the sample schema.

5. When a message displays indicating that the bind parameter `p_order_id` was created, click **OK**.
6. In the data model you just created, select all of the following columns using Shift-click, then drag them below the current query into a detail group:
- **LINE_ITEM_ID**
 - **PRODUCT_NAME**
 - **UNIT_PRICE**
 - **QUANTITY**

The resulting data model should look like this:

Figure 40–2 Data Model for the query



40.2.5 Create a formula column to retrieve the barcode image

To create a formula column:

1. In the Data Model view, click the Formula Column tool in the tool palette.

2. Click in the master group (the main group that still contains most of the column names) to create a new formula column.
3. Double-click the new formula column object (**CF_1**) to display the Property Inspector, and set properties:
 - Under **Column**, set the Datatype property to Character.
 - Under **Placeholder/Formula**, click the PL/SQL Formula property field to display the PL/SQL Editor.
4. In the PL/SQL Editor, use the template to enter the following PL/SQL code:

```
function CF_1Formula return VARCHAR2 is
  myFileName varchar2(500);
  result varchar2(500);
  barcodeData VARCHAR2(50) := :customer_ID || :order_ID;
begin
myFileName := srw.create_temporary_filename;
  barcodemaker.setBarWidthInch(globals.bcobj, 0.005);
  barcodemaker.setBaseCodeData(globals.bcobj,barcodeData);
  barcodemaker.setBarCodeType(globals.bcobj,globals.barcode_to_use);
  barcodemaker.setFullPath(globals.bcobj, myFileName);
  barcodemaker.renderBarCode(globals.bcobj);
  return(myFileName);
end;
```

Note: You can enter this code by copying and pasting it from the provided text file called `barcode_code.txt`.

5. Click **Compile** to make sure there aren't any errors.

Note: If you have errors, make sure you've imported the necessary Java classes and that your code matches the code above. If you change the code, be sure to compile it again.

6. When the code is compiled, click **Close**.
7. Double-click the new formula column object (**CF_1**) to display the Property Inspector, and set properties:
 - Under **Column**, set the Read from File property to Yes, and set the File Format property to Image.

40.2.6 Create a formula column that returns the order total

To create the second formula column:

1. In the Data Model view, create a formula column in the detail group **G_LINE_ITEM_ID**.
2. Open the Property Inspector for the formula column.
3. Under **General Information**, set the Name property to `LineTotal`.
 - Under **Column**, make sure the Datatype property is set to Number.
 - Under **Placeholder/Formula**, click the PL/SQL Formula property field to display the PL/SQL Editor.
4. In the PL/SQL Editor, use the template to enter the following PL/SQL code:

```
function LineTotalFormula return Number is
begin
    return (:quantity * :unit_price);
end;
```

Note: You can enter this code by copying and pasting it from the provided text file called `barcode_code.txt`.

5. Click **Compile** to make sure there aren't any errors.
6. When the code is compiled, click **Close**.
7. Save the report.

You have created the data model for your barcode report, which contains a formula column that retrieves the barcode information and displays the barcode image on your report, and another formula column that displays the order total.

Your data model and the PL/SQL for the formula column should look similar to this:

Figure 40–3 Data Model with two new formula columns



40.2.7 Create a layout for your report

Before you can run your report, you must create a layout.

To create a paper layout:

1. Under your report's node in the Object Navigator, right-click **Paper Layout**, then choose **Report Wizard**.
2. In the Report Wizard, on the **Report Type** page, select **Create both Web and Paper Layout**, then click **Next**.

3. On the **Style** page, select **Group Above**, then click **Next**.
4. On the **Data Source** page, click **Next**.
5. On the **Data** page, click **Next**.
6. On the **Groups** page, make sure the following fields are listed in the **Group Fields** list (if not, use the arrows to move the field to the appropriate list):
 - **ORDER_ID**
 - **ORDER_DATE**
 - **CUSTOMER_ID**
 - **CUST_FIRST_NAME**
 - **CUST_LAST_NAME**
 - **STREET_ADDRESS**
 - **POSTAL_CODE**
 - **COUNTRY_NAME**
 - **CITY**
 - **STATE_PROVINCE**
 - **COUNTRY_ID**
 - **ORDER_TOTAL**
 - **CF_1**
7. On the **Fields** page, click the double right arrows (>>) to move all of the fields to the **Displayed Fields** list, then click **Finish**.
8. In the Paper Layout view, click the Run Paper Layout button in the toolbar to run your report.
9. In the Runtime Parameter Form, next to **P_ORDER_ID**, type 2354.
10. Once your report displays in the Paper Design view, you can rearrange your layout objects in the Paper Layout view to make your report look something like this:

Figure 40–4 Paper Design view of the barcode paper report

Shipping Details

Harrison Sutherland
6445 Bay Harbor Ln
Indianapolis IN 46254
United States of America

Tracking Details

1042354

Order Details

Order ID 2354
Order Date 14-JUL-00

46,257.

ItemNo.	Product Name	Quantity	Unit Price	
1	KB 101/EN	61	48.00	2,9
2	MB - \$900/650+	43	96.80	4,1
3	PS 220V /D	47	79.00	3,7
4	Sound Card STD	47	41.00	1,9
5	Screws <S.16.S>	48	21.00	1,0

Note: If you aren't sure whether you produced the desired results, you can always open the file we provided, called `ShippingManifest.pdf` in Acrobat Reader. Or, you can run `ShippingManifest.rdf` to paper and the report will display in the Paper Design view.

11. Save the report. You have now finished building a barcode report for paper.

40.3 Create a barcode report for the Web

The steps in this section show you how to build a Web report using JavaServer Pages (JSPs), using the barcode JavaBean you imported in [Section 40.2.1, "Import the Java classes into Reports Builder"](#). If you want to build a paper report with a barcode, see [Section 40.2, "Create a barcode report for paper"](#).

If you are not familiar with creating a JSP-based Web report and would like to learn how to create one, refer to the *Oracle Reports Tutorial*, located in the **Getting Started with Oracle Reports** Web site on the Oracle Technology Network (<http://otn.oracle.com/reports/>).

The report you will create in this section is the same as the one you created for paper. You will create a report that displays the invoice for a particular customer. This invoice will display the address of the customer, his order, and a barcode that represents the tracking number for the order. The company can use this barcode to find out the status of the order.

You can run the final version of the JSP report we've provided to see what you'll build in these steps, but please note that you will need to update the location of the images in the source code (see [Section 40.3.3, "Initialize the barcode JavaBean and set its properties"](#)) before you can run the report to the Web.

Note: Before you begin this section, make sure you have all the necessary files, and that you've imported the Java classes, and set up the class path. See [Section 40.1, "Prerequisites for this example"](#) and [Section 40.2.1, "Import the Java classes into Reports Builder"](#).

40.3.1 Create a query in an existing HTML file

When you create a JSP-based Web report, you can use an existing HTML file as a template. The steps in this section will show you how to open an HTML file in Reports Builder and add data to it.

To create a query in an existing HTML file:

1. In Reports Builder, choose **File > Open** and open the file
Examples\BarCodeBeanWeb\source\ShippingLabel.html.
2. In the Object Navigator, under **SHIPPINGLABEL**, double-click the view icon next to the **Data Model** node to display the Data Model view for the report.
3. In the Data Model view, click the SQL Query tool in the tool palette, then click in an open area of the Data Model view to display the SQL Query Statement dialog box.
4. In the **SQL Query Statement** field, type (or paste) the following code:

```
SELECT ALL CUSTOMERS_A1.CUST_FIRST_NAME,  
       CUSTOMERS_A1.CUSTOMER_ID, CUSTOMERS_A1.CUST_LAST_NAME,  
       CUSTOMERS_A1.CUST_ADDRESS.STREET_ADDRESS,  
       CUSTOMERS_A1.CUST_ADDRESS.POSTAL_CODE,  
       CUSTOMERS_A1.CUST_ADDRESS.CITY,  
       CUSTOMERS_A1.CUST_ADDRESS.STATE_PROVINCE,  
       CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID,  
       ORDERS.ORDER_ID, ORDERS.ORDER_DATE, ORDERS.ORDER_TOTAL,
```

```
COUNTRIES.COUNTRY_NAME FROM CUSTOMERS CUSTOMERS_A1, ORDERS,  
HR.COUNTRIES  
WHERE ((ORDERS.CUSTOMER_ID = CUSTOMERS_A1.CUSTOMER_ID)  
AND (CUSTOMERS_A1.CUST_ADDRESS.COUNTRY_ID = HR.COUNTRIES.COUNTRY_ID))  
AND ORDERS.ORDER_ID = :P_ORDER_ID ORDER BY order_ID
```

Note: You can enter this query in any of the following ways:

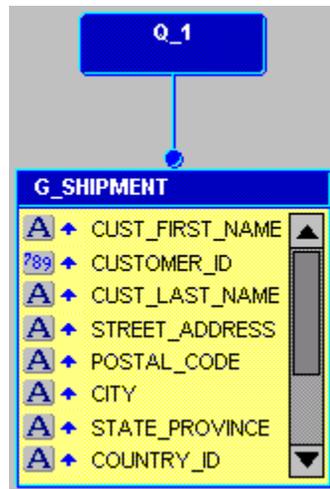
- Copy and paste the code from the provided text file called `barcode_code.txt` into the **SQL Query Statement** field.
 - Click **Query Builder** to build the query without entering any code manually.
 - Type the code in the **SQL Query Statement** field.
-
-

5. Click **OK**.

If you are not connected to a database that contains the sample schema we've provided, you must log in now. If you're not sure what your connection string is, contact your database administrator. Note that this example uses the "Order Entry" portion of the sample schema.

6. When a message displays indicating that the bind parameter was created, click **OK**.
7. In the Data Model view, double-click the group object to display the Property Inspector, and set properties:
- Under **General Information**, set the Name property to `G_SHIPMENT`.

Your data model should look something like this:

Figure 40-5 Data Model for the JSP-based Web report query

8. Save your report as ShippingLabel_<your initials>.jsp to create the JSP-based Web source for this report.

You have now created the query that will pull in the data for your report.

40.3.2 Create three formula columns in your data model

You will need to create three formula columns in your report to retrieve the tracking number for the order, the origin of the order, and the destination for the order.

To create the TrackingNumber formula column:

1. In the Data Model view, click the Formula Column tool in the tool palette.
2. Click in the G_SHIPMENT group to create a new formula column.
3. Double-click the new formula column object (CF_1) to display the Property Inspector, and set properties:
 - Under **General Information**, set the Name property to TrackingNumber.
 - Under **Column**, set the Datatype property to Character.
 - Under **Placeholder/Formula**, click the PL/SQL Formula property field to display the PL/SQL Editor.

4. In the PL/SQL Editor, use the template to enter the following PL/SQL code:

```
function TrackingNumberFormula return char is
begin
    return (:Customer_id||:Order_ID||:country_ID);
end;
```

Note: You can enter this code by copying and pasting it from the provided text file called `barcode_code.txt`.

5. Click **Compile**.

Note: If your code does not compile, make sure you've typed in exactly the code we've provided.

6. When the code is compiled, click **Close**.

To create the OriginScan formula column:

1. Create another formula column in the **G_SHIPMENT** group.
2. Double-click the new formula column object (**CF_1**) to display the Property Inspector, and set properties:
 - Under **General Information**, set the Name property to OriginScan.
 - Under **Column**, set the Datatype property to Character.
 - Under **Placeholder/Formula**, click the PL/SQL Formula property field to display the PL/SQL Editor.
3. In the PL/SQL Editor, use the template to enter the following PL/SQL code:

```
function OriginScanFormula return char is
begin
    return ('34324-OH-US');
end;
```

Note: You can enter this code by copying and pasting it from the provided text file called `barcode_code.txt`.

4. Click **Compile**.

Note: If your code does not compile, make sure you've typed in exactly the code we've provided.

5. When the code is compiled, click **Close**.

To create the **DestinationScan** formula column:

1. Create a third formula column in the **G_SHIPMENT** group.
2. Double-click the new formula column object (**CF_1**) to display the Property Inspector, and set properties:
 - Under **General Information**, set the Name property to DestinationScan.
 - Under **Column**, set the Datatype property to Character.
 - Under **Placeholder/Formula**, click the PL/SQL Formula property field to display the PL/SQL Editor.
3. In the PL/SQL Editor, use the template to enter the following PL/SQL code:

```
function DestinationScanFormula return char is
begin
    return (:postal_code || '-' || :state_province || '-' || :country_ID);
end;
```

Note: You can enter this code by copying and pasting it from the provided text file called `barcode_code.txt`.

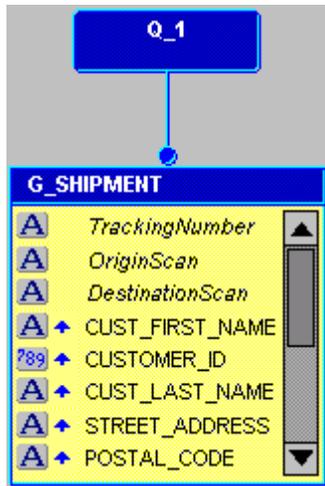
4. Click **Compile**.

Note: If your code does not compile, make sure you've typed in exactly the code we've provided.

5. When the code is compiled, click **Close**.
6. Save your report as a JSP.

You have created the three formula columns that will hold the values for the tracking number, the origin, and the destination of the order. Your data model should look something like this:

Figure 40–6 Data Model with the three formula columns



40.3.3 Initialize the barcode JavaBean and set its properties

To enable your JSP-based Web report to communicate with the JavaBean, you need to initialize it in the JSP. Unlike using the JavaBean with a paper report, you do *not* need to use the Java importer to import the Java classes.

In this section, you will also learn how to set the properties for the bean so that the correct data is being used to produce the barcode.

If you don't want to bother with typing the code in yourself, you can always open the source file

(*Examples/BarCodeBeanWeb/source/ShippingManifestWeb.rdf*) and copy the appropriate pieces of the Web source into your report.

To ensure that the JavaBean references the correct barcode images, you must first update your report with the correct path.

To update the paths:

1. In the Object Navigator, double-click the view icon next to the **Web Source** node for your JSP-based Web report (**ShippingLabel_<your initials>**) to display the Web Source view.

2. In the Web Source view, look for the text: Define Path information for your barcode images.
3. Under this text, update the paths to make sure they point to the directories where your images will be generated (e.g., d:\\temp\\docroot\\images\\). Be sure to maintain the integrity of the paths we've provided.

Note: The directory where these files are located must be accessible by the Web server. This directory is typically located somewhere below the directory where you keep your JSPs. Make sure that this directory exists on your computer before you run the report to the Web.

To initialize the JavaBean:

1. In the Web Source view, look for the text: Initialize the JavaBeans.
2. Under this text, type the following code:

```
<jsp:useBean id="BC" scope="page"
class="oracle.apps.barcode.util.BarCodeConstants" />
<jsp:useBean id="BM" scope="page" class="oracle.apps.barcode.BarCodeMaker"
/>
```

Note: You can enter all the code in this section by copying and pasting it from the provided text file called `barcode_code.txt`.

To set the barcode JavaBean properties:

1. In the Web Source view, look for the text: Setting the barcodes properties.
2. Under this text, type the following code:

```
<jsp:setProperty name="BM" property="BarCodeType" value="<%= BC.BAR_CODE_128
%>" />
<jsp:setProperty name="BM" property="BarWidthInch" value="0.01"/>
<jsp:setProperty name="BM" property="Directory" value="<%=
BarcodePhysicalPath %>"/>
```

To define the barcode variables:

1. In the Web Source view, look for the text:
Define variables to hold the data for the three barcodes.
2. Under this text, type the following code:

```
<%! private String BarCodeData1 = "12345-XX-XX"; %>  
<%! private String BarCodeData2 = "12345-XX-XX"; %>  
<%! private String BarCodeData3 = "12345-XX-XX"; %>
```

To create a For Each loop:

1. In the Web Source view, look for the text:
Replace this with your RW:FOREACH open tag.
2. Replace this line with the following code:

```
<rw:foreach id="R_G_SHIPMENT" src="G_SHIPMENT">
```

3. Look for the text:
Replace this with your RW:FOREACH close tag.
4. Replace this line with the following code:

```
</rw:foreach>
```

To code the formula columns to render the barcodes:

1. In the Web Source view, look for the text:
BARCODEShippingTrackingNumber.
2. Under this text, type the following code:

```
<!-- Get the value of the TrackingNumber and assign it to the variable -->  
<rw:getValue id="BarCodeData1" src="TrackingNumber"/>  
<!-- Set the data for the barcode and the filename -->  
<jsp:setProperty name="BM" property="BaseCodeData" value="<%= BarCodeData1  
%"/>  
<jsp:setProperty name="BM" property="FileName" value="<%= BarCodeData1 %"/>  
<!-- Render the barcode -->  
<% BM.renderBarCode(); %>  
<!-- View the image in the page -->  

```

Note: the lines beginning with "`<!--`" are comments. If you do not want comments in your code, you do not have to add these lines.

3. In the Web Source view, look for the text:

```
**BARCODEOriginScan**.
```

4. Under this text, type the following code:

```
<!-- Get the value of the OriginScan and assign it to the variable -->
<rw:getValue id="BarCodeData2" src="OriginScan"/>
<!-- Set the data for the barcode and the filename -->
<jsp:setProperty name="BM" property="BaseCodeData" value="<%= BarCodeData2
%>"/>
<jsp:setProperty name="BM" property="FileName" value="<%= BarCodeData2 %>"/>
<!-- Render the barcode -->
<% BM.renderBarCode(); %>
<!-- View the image in the page -->

```

5. In the Web Source view, look for the text:

```
**BARCODEDestinationScan**.
```

6. Under this text, type the following code:

```
<!-- Get the value of the DestinationScan and assign it to the variable -->
<rw:getValue id="BarCodeData3" src="DestinationScan"/>
<!-- Set the data for the barcode and the filename -->
<jsp:setProperty name="BM" property="BaseCodeData" value="<%= BarCodeData3
%>"/>
<jsp:setProperty name="BM" property="FileName" value="<%= BarCodeData3 %>"/>
<!-- Render the barcode -->
<% BM.renderBarCode(); %>
<!-- View the image in the page -->

```

7. Save your report as a JSP.

40.3.4 Run your report to the Web

Since you've created a Web report, you must run this report to the Web to see your results.

1. In the Object Navigator, click your report name, **ShippingManifestWeb_<your initials>**.

- Click the Run Web Layout button in the toolbar to run your report to a browser.

Note: If Netscape 7.0 is your default browser, the browser may not display. You can work around this bug by making a copy of the Netscape 7.0 executable, naming it netscape.exe; with this name, the browser will display as expected.

Your report displays in your Web browser, and should look something like this:

Figure 40-7 Snapshot of the final JSP-based Web report with barcode



Note: If you aren't sure whether you produced the desired results, you can always open the file we provided in the results directory, called ShippingManifestWeb.html in your Web browser. Or, once you've updated the images path, you can run ShippingManifestWeb.jsp to the Web, and the report will display in the your browser.

40.4 Summary

Congratulations! You have created a paper report and a JSP-based Web report that use the barcode JavaBean to generate barcode images.

You now know how to:

- use the Java importer to add Java classes to a paper report.
- use JavaServer Pages to call a JavaBean from within a report.
- create a PL/SQL package.
- use a Before Report trigger to tell Reports Builder what type of barcode image to use.
- manually build a data model with a SQL query and formula columns.
- edit the code in the Web Source view for a JSP-based Web report.
- set up a JavaBean in JSP.

For more information on any of the wizards, views, or properties used in this example, refer to the *Reports Builder Online Help*, which you can access in two ways:

- From the Oracle Technology Network (<http://otn.oracle.com/products/reports/>), click **Documentation** and navigate to the *Reports Builder Online Help* for the most recent, hosted online help.
- From Reports Builder, choose **Help > Help Contents**.

Note: You can replace your Reports Builder online help with the most recent update by downloading the latest online help set available on the Oracle Technology Network (<http://otn.oracle.com/docs/products/reports/>). Instructions for replacing your help file are included in the `readme.txt` in the download file.
