

How to add Barcode in Local Reports (RDLC) before report rendering stage

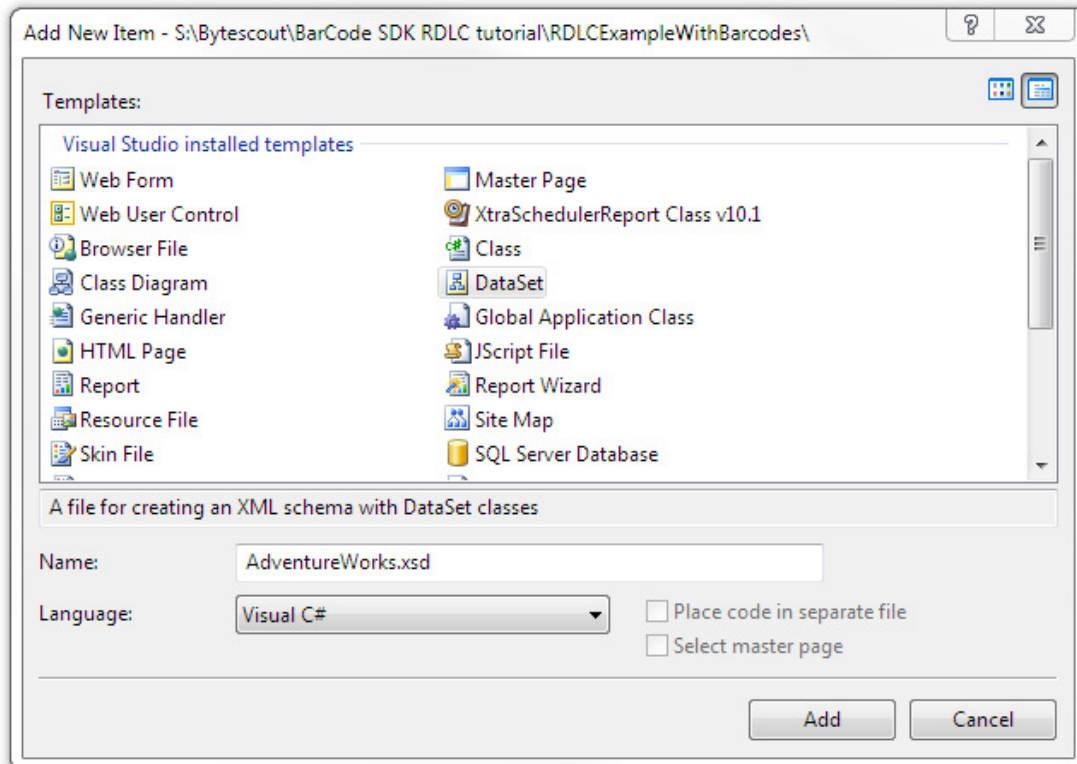
Prerequisites:

- Bytescout Barcode SDK.
- Microsoft .NET Framework 2.0 (or greater).
- Microsoft Visual Studio 2005 (or greater) or Visual Web Developer 2005 Express Edition (or greater).
- Microsoft SQL Server 2005 (any version) or greater with Adventure Works sample database installed.

In the following guide we'll create a local report (RDLC file) which features barcoding capabilities by using Bytescout Barcode SDK.

Follow these steps:

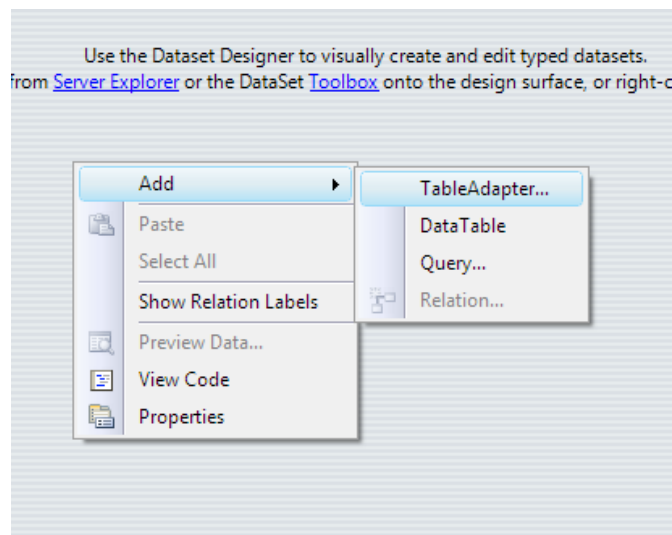
1. Open Visual Studio and create a new ASP.NET Website naming it "RDLCExampleWithBarcode".
2. Add new *DataSet* item to the project and name it "AdventureWorks.xsd".



Click **Add** button. You will be asked if you want to place the AdventureWorks.xsd file in the **App_Code** directory. **Answer yes.**

VS2005: After that, the **TableAdapter Configuration Wizard** is automatically launched so please follow its steps.

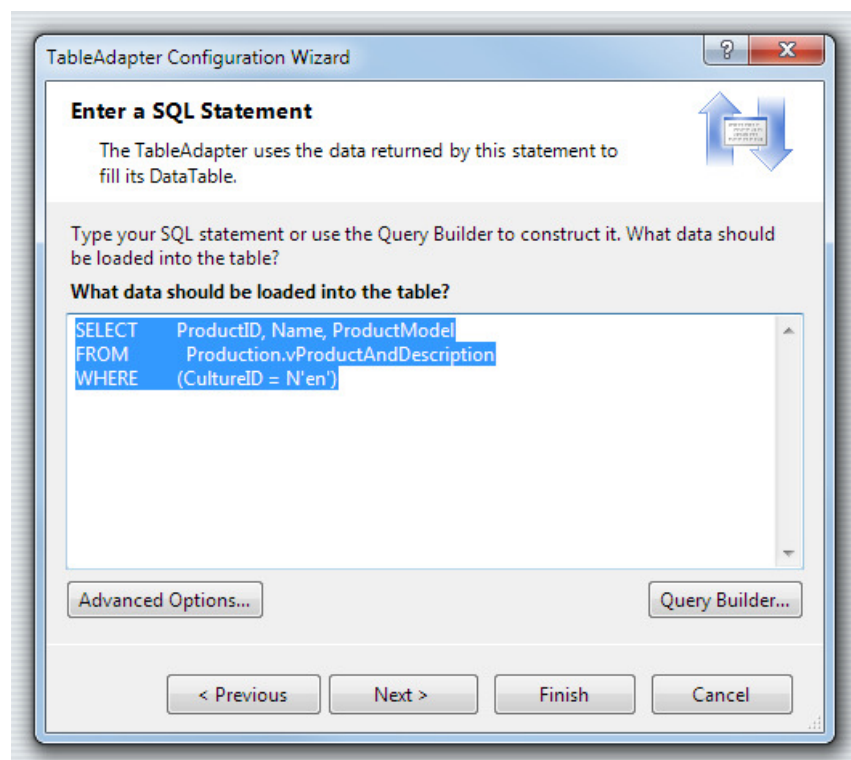
VS2008 and later: Right click the design surface and select Add > TableAdapter...



This will also run **TableAdapter Configuration Wizard**. Follow its steps.

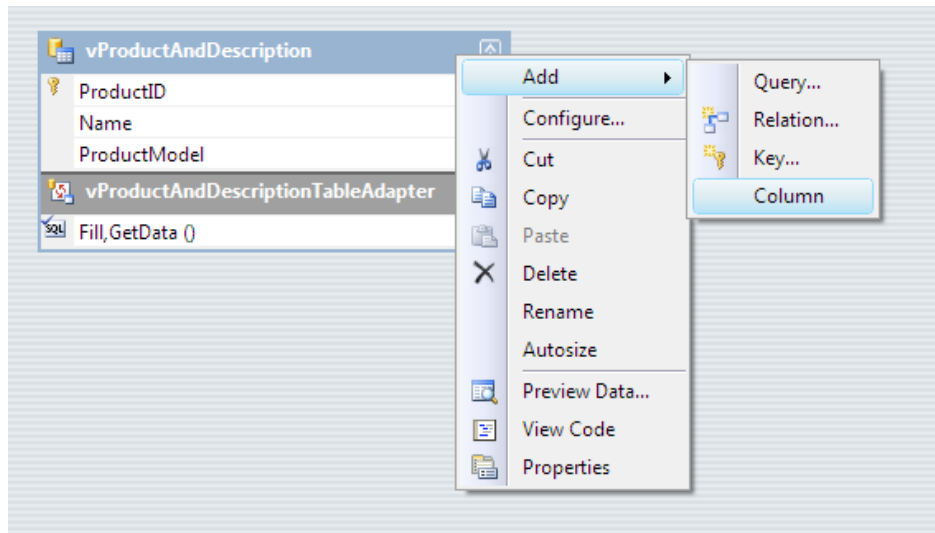
In the first step create a connection to the AdventureWorks SQL Server Database sample and click Next. In the second step, choose "Use SQL statements" and click Next. After that, please enter the following SQL Statement:

```
SELECT ProductID, Name, ProductModel FROM Production.vProductAndDescription WHERE (CultureID = N'en')
```

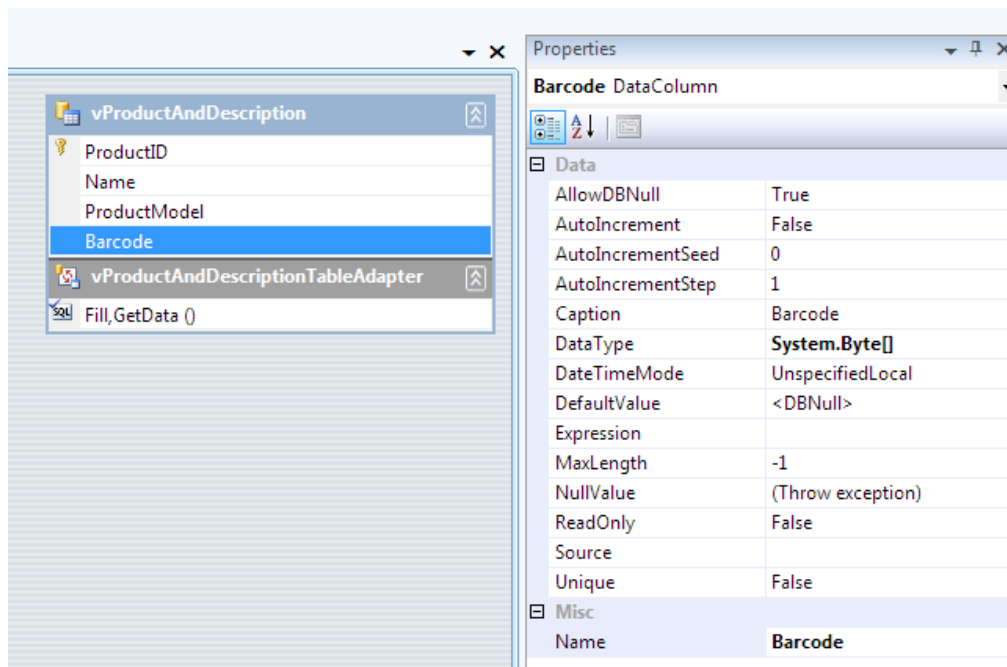


Click Finish to close the wizard dialog box.

3. Add a new custom **Column** to the **DataTable** just created and name it "**Barcode**".

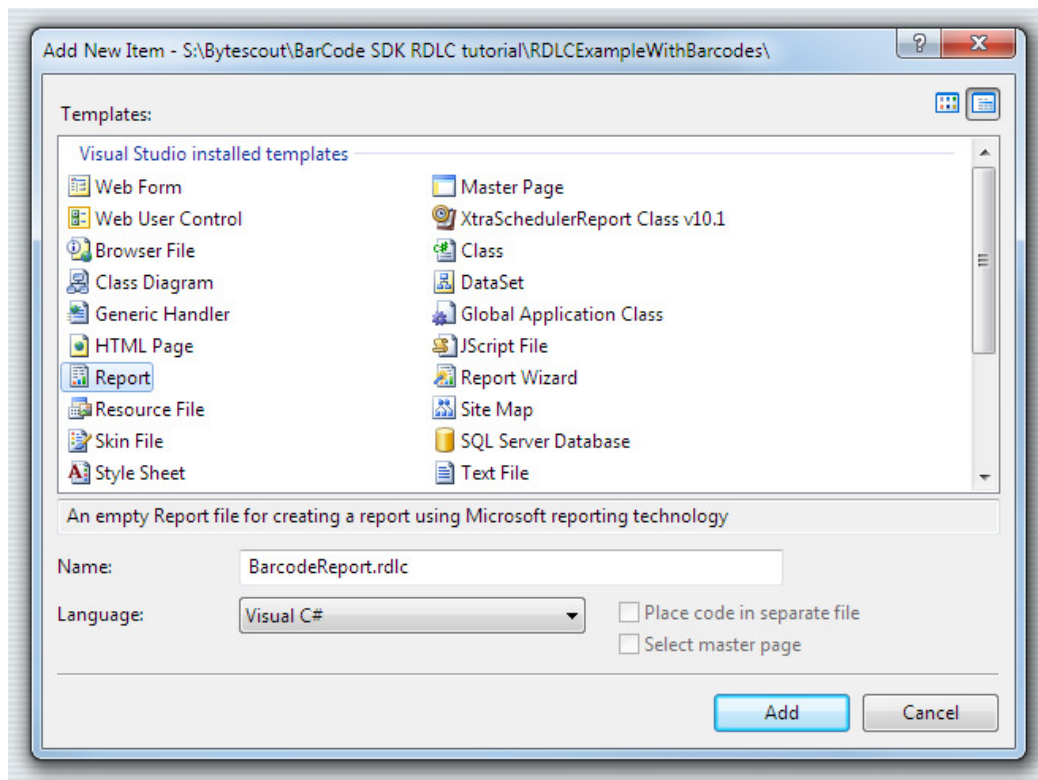


4. Change the data type of the **Barcode** column to **System.Byte[] (Array of Byte)**. If the **System.Byte[]** data type is not listed you must type it manually.

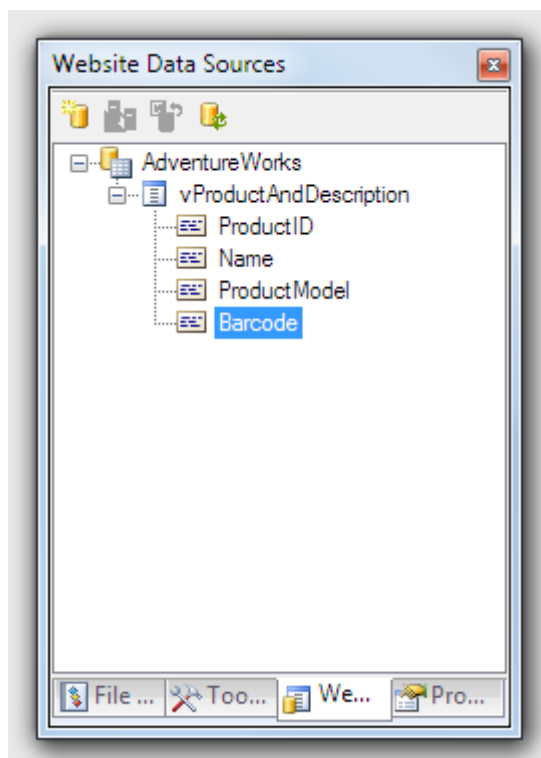


Save the **AdventureWorks.xsd** file.

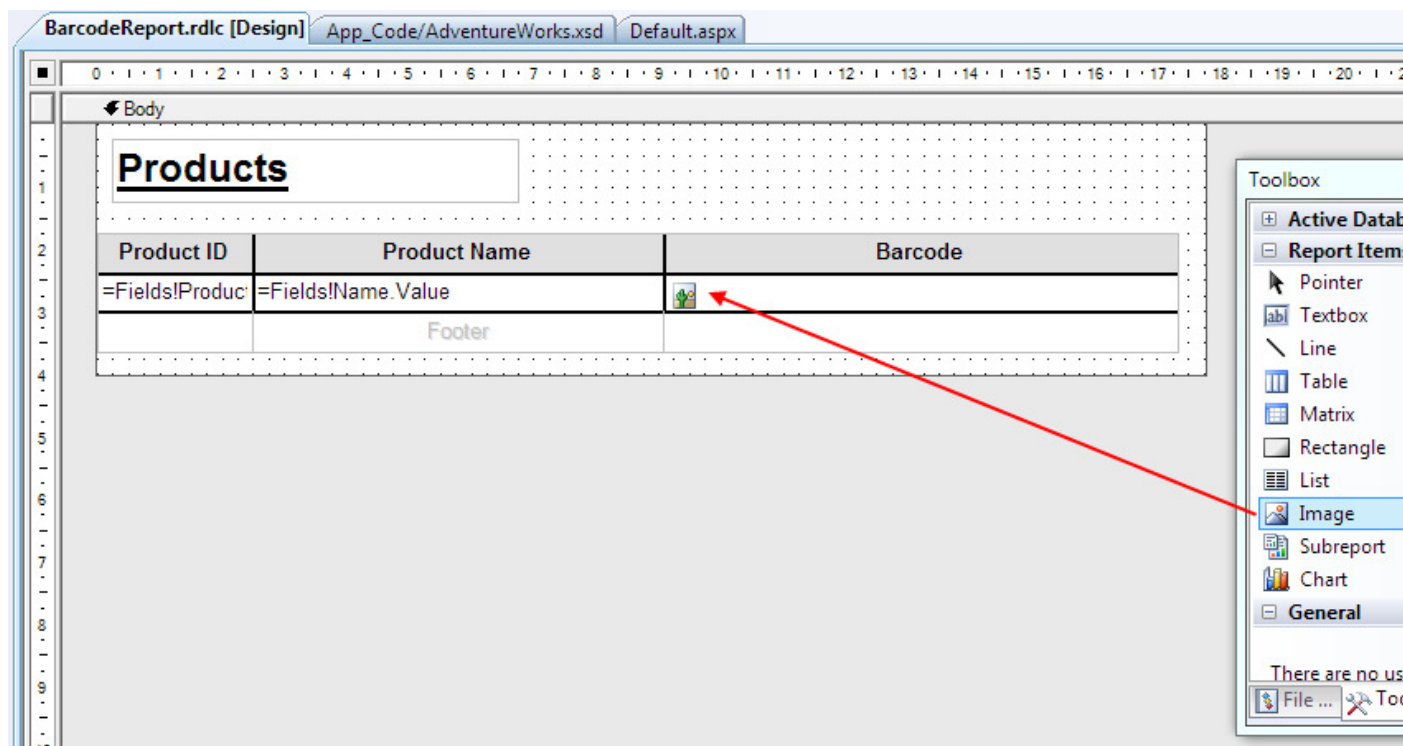
5. Now add a new **Report** item to the project and name it **BarcodeReport.rdlc**.



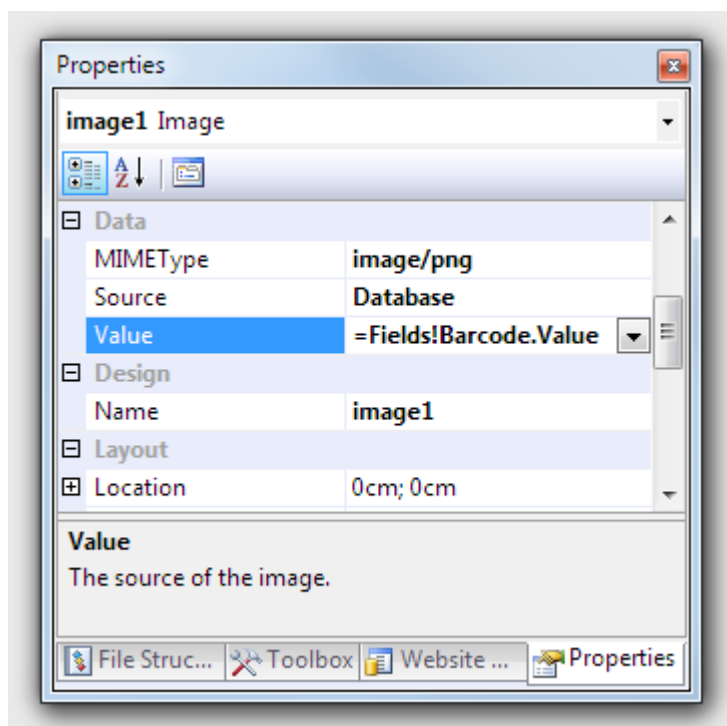
The data source for the report should look like the following figure. NOTE: You can display Data Sources Window by selecting Data menu and then Show Data Sources (Shift+Alt+D).



6. Please design the report so it looks like the following figure.



The report features a **Table** item with 3 columns: **Product ID**, **Product Name** and **Barcode**. **Barcode** column holds an image. Drag & drop an **Image** item into it and set its properties as is shown in the following figure. Notice that **Value** property of the **Image** item is bound to the **Barcode** column.



Save the report.

7. Create/Open an ASP.NET WebForm at design time and drag & drop a **ReportViewer** control onto it. NOTE: DO NOT select any report to display. We'll set up it by code next.

8. Now, from the Solution Explorer, add a reference to Bytescout Barcode SDK assembly (**Bytescout.BarCode.dll**). Look for it in the installation directory.
9. Write the following code in the **Page_Load** event procedure:

C#:

```
using System;

using Bytescout.BarCode;
using Microsoft.Reporting.WebForms;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        // Fill the datasource from DB
        AdventureWorksTableAdapters.vProductAndDescriptionTableAdapter ta =
            new AdventureWorksTableAdapters.vProductAndDescriptionTableAdapter();
        AdventureWorks.vProductAndDescriptionDataTable dt =
            new AdventureWorks.vProductAndDescriptionDataTable();
        ta.Fill(dt);

        // Create and setup an instance of Bytescout Barcode SDK
        Barcode bc = new Barcode(SymbologyType.Code128);
        bc.RegistrationName = "demo";
        bc.RegistrationKey = "demo";
        bc.DrawCaption = false;

        // Update DataTable with barcode image
        foreach (AdventureWorks.vProductAndDescriptionRow row in dt.Rows)
        {
            // Set the value to encode
            bc.Value = row.ProductID.ToString();
            // Generate the barcode image and store it into the Barcode Column
            row.Barcode = bc.GetImageBytesPNG();
        }

        // Create Report Data Source
        Microsoft.Reporting.WebForms.ReportDataSource rptDataSource =
            new ReportDataSource("AdventureWorks_vProductAndDescription", dt);

        ReportViewer1.LocalReport.DataSources.Add(rptDataSource);
        ReportViewer1.LocalReport.ReportPath = Server.MapPath("BarcodeReport.rdlc");
        ReportViewer1.LocalReport.Refresh();
    }
}
```

Visual Basic:

```
Imports Bytescout.BarCode

Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles form1.Load

        ' Fill the datasource from DB
        Dim ta As New AdventureWorksTableAdapters.vProductAndDescriptionTableAdapter()
        Dim dt As New AdventureWorks.vProductAndDescriptionDataTable()
        ta.Fill(dt)

        ' Create and setup an instance of Bytescout Barcode SDK
        Dim bc As New Barcode(SymbologyType.Code128)
        bc.RegistrationName = "demo"
        bc.RegistrationKey = "demo"
        bc.DrawCaption = False

        ' Update DataTable with barcode image
        Dim row As AdventureWorks.vProductAndDescriptionRow

        For Each row In dt.Rows
            ' Set the value to encode
            bc.Value = row.ProductID.ToString()
        Next
    End Sub
End Class
```

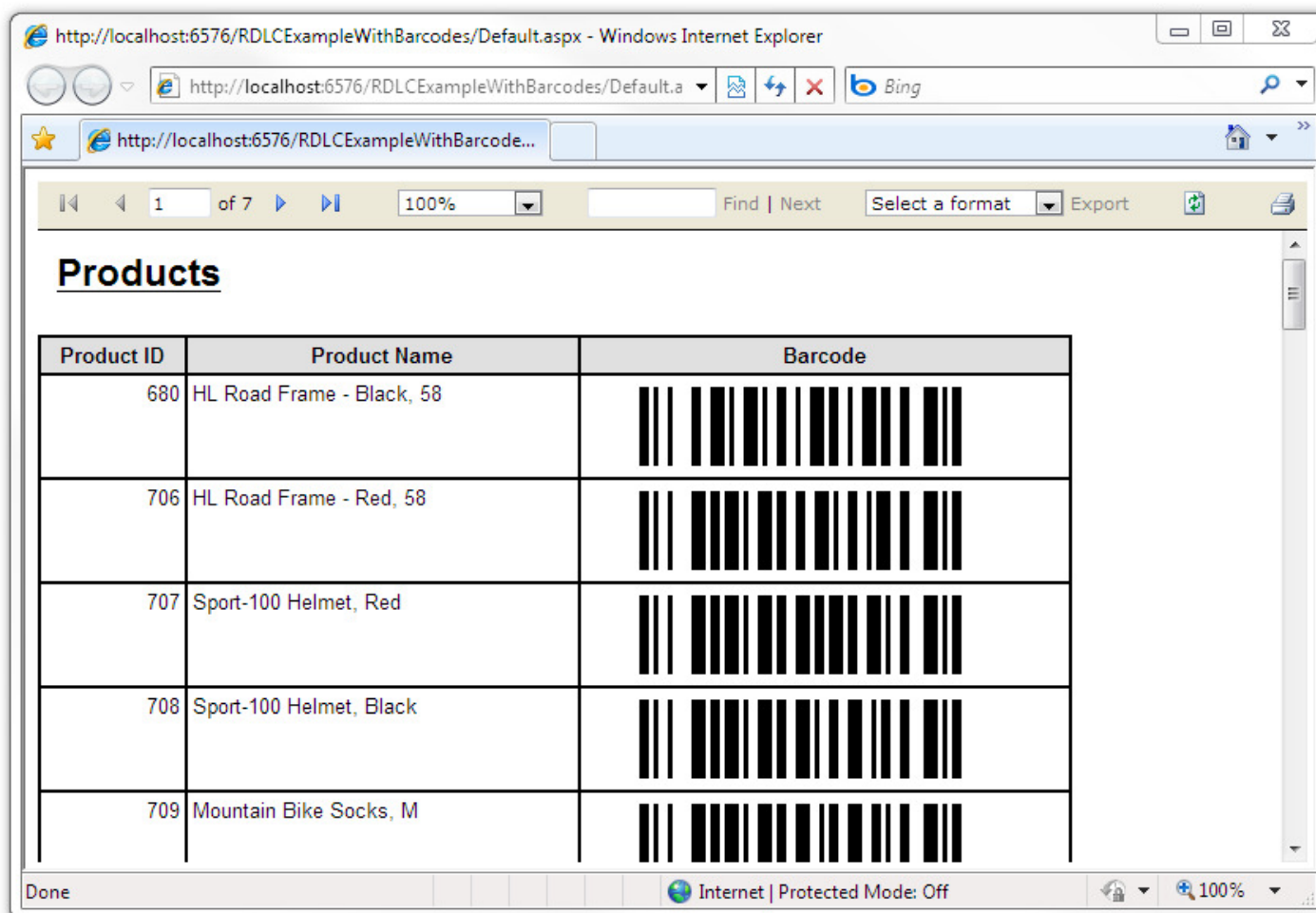
```

        'Generate the barcode image and store it into the Barcode Column
        row.Barcode = bc.GetImageBytesPNG()
    Next

    'Create Report Data Source
    Dim rptDataSource As New
Microsoft.Reporting.WebForms.ReportDataSource("AdventureWorks_vProductAndDescription", dt)
    Me.ReportViewer1.LocalReport.DataSources.Add(rptDataSource)
    Me.ReportViewer1.LocalReport.ReportPath = Server.MapPath("BarcodeReport.rdlc")
    Me.ReportViewer1.LocalReport.Refresh()
End Sub
End Class

```

10. Run your application. You should get the barcode images displayed on the report.



ReportViewer control lets you to export the displayed report to Acrobat PDF as well as Microsoft Excel XLS. In both cases the barcode images are maintained.