B

Implementing RSS and Atom Feed

In today's scenario, Internet users frequently download Web pages of their favorite websites and search the content for any updates. These websites range from news portals to job portals. Accessing websites in this way, however, consumes a lot of time since the surfers have to open the websites and browse their favorite pages for the updates. A much simpler approach is to access the updated content through the use of Web syndication formats, such as Really Simple Syndication (RSS) and Atom.

Syndication of a Web page is a quick and easy way to create and use distributed or scattered content, such as breaking news. Nowadays, RSS and Atom are used for many purposes, such as marketing, bug reporting, consistent updating, and publications. It is very common to see RSS or Atom on various blogs and community websites. RSS enables a website to send article previews and alerts to customers about new products and upcoming events. In this appendix, you learn about RSS feed and Atom feed in detail.

RSS Feed

RSS is one of the Web feed formats used to keep you updated of the changes brought about in chosen website. A Web feed is used to provide frequently updated content of a Web page. It is a document (often XML-based) that contains content items with Web links. Web feeds are designed to be machine-readable (computer) rather than human-readable. RSS too uses an XML code that constantly scans the content of a website for new updates and broadcasts them to users through the feed. When an update is sent, it includes a headline and a small amount of text, either a summary or a link to the whole text. Table B.1 lists the various versions of RSS:

Table B.1: RSS Versions

RSS Version	Description
`RSS 0.90`	Designed by Netscape for building portals of headlines to news websites. This version is not used nowadays.
`RSS 0.91`	Proposed by Netscape, but developed by UserLand. Developers use this version for developing Web-based software. This version is called Rich Site Summary and is relatively easy to use. It is commonly used for basic syndication and offers an easy migration path to developers.
`RSS 0.92, RSS 0.93, RSS 0.94`	Developed by UserLand, this version allows richer metadata than 0.91, but has now been replaced by 2.0.

Table B.1: RSS Versions

RSS Version	Description
`RSS 1.0`	Developed by RSS-DEV Working Group, this format was based on Resource Description Framework (RDF) and is used in applications where the use of advance RDF-specific modules is desired. This ensures extensibility of a website, which can be managed by more than one vendor.
`RSS 2.0`	Developed by UserLand, this version offers extensibility by use of modules and has an easy migration path from the 0.9x branch. It is used for general-purpose and metadata-rich syndication.

It is also possible to get an RSS update automatically without any user intervention. This task is often performed by some special programs known as `Feedreader` or `aggregator`. These are programs or Web applications that aggregate syndicated Web content, such as news headlines and blogs, and thereby reduce the time and effort by checking websites for updates. After it is subscribed to a feed, an aggregator checks for new content at user-determined intervals and also retrieves the update. RSS readers or aggregators enable a website to provide with a combined view of its content in a single browser display or desktop application. An RSS document conforms to `XML 1.0` specifications, as published on the `World Wide Web Consortium (W3C)` website.

The syntax of RSS 2.0 is as follows:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rss version="2.0">
<channel>
  <title>Welcome to MyNewsSite1</title>
  <link>http:// MyNewsSite1.com </link>
  <description>News Update </description>
  <item>
  <title>Update 1</title>
  <link>http:// MyNewsSite1.com/update1</link>
  <description>Check out for todays business scoop </description>
  </item>
</channel>
</rss>
```

In the preceding syntax, the first line is the XML declaration that specifies the XML version of a document. Some elements and attributes used in the preceding syntax are explained as follows:

- `<rss>`—Uses the `version` attribute to specify the version of RSS to which a document conforms. In the preceding syntax, the `version` attribute is set to 2.0 to specify that the RSS document conforms to RSS 2.0.
- `<channel>`—Contains the metadata and acts like the headline of the channel. This element is nested within the `<rss>` element. There can be only one `<channel>` element in the RSS document. As shown in the preceding code, the `<channel>` element contains the following three attributes:
  - `<title>`—Specifies the name of the channel or the website corresponding to the channel.
  - `<link>`—Specifies the link of the home page of the website that corresponds to the channel.
  - `<description>`—Contains a phrase or sentence to describe the RSS feed.

The <channel> element supports a number of sub-elements, as listed in Table B.2:

Table B.2: Sub-Elements of the <channel> Element

Element	Description
<category>	Specifies the category of the channel.
<cloud>	Contains a notification for content updates.
<copyright>	Contains information about the copyright material of a website.
<docs>	Specifies the Uniform Resource Locator (URL) of the Web page containing the documentation on the RSS version used by the channel.

Table B.2: Sub-Elements of the <channel> Element

Element	Description
<generator>	Defines the automatic generator that indicates the program used to generate the channel.
<image>	Inserts graphics in the channel.
<language>	Specifies the language used by a channel. This element is used to group websites based on the language used by RSS aggregators.
<lastBuildDate>	Specifies the date on which a channel was last modified.
<managingEditor>	Specifies the e-mail address of the editor of a website.
<pubDate>	Specifies the publication date for the content in a channel.
<rating>	Specifies the parental control rating of a Web page.
<skipDays>	Specifies the number of days for which the feed is not updated.
<skipHours>	Specifies the number of hours for which the feed is not updated.
<textInput>	Creates a text input box that can be displayed with the channel.
<ttl>	Specifies the minutes for which the channel remains cached without refreshing.
<webMaster>	Specifies the e-mail address of the webmaster of a feed.
<title>	Specifies the title of the channel

- `<Item>`—Contains information, such as title, link, and description, which you intend to display on your RSS feed. It enables a developer to provide links to the content (including the updated content) of a website. The <Item> element supports a number of attributes. These attributes were introduced with RSS 2.0, which provides greater flexibility to work with the files of earlier version of RSS, such as RSS 0.91 and RSS 1.0. The sub-elements of the <Item> element are listed in Table B.3:

Table B.3: Sub-Elements of the <item> Element

Element	Description
<author>	Specifies the name of the author who has written the content included in an item
<category>	Specifies the category of a channel
<comments>	Contains the URL of the Web page that contains the comments related to an item
<enclosure>	Specifies the media object, such as mp3 files, that are enclosed with the content of an item
<guid>	Defines the Globally Unique Identifier (GUID), which is a unique value, for identifying an item
<pubDate>	Defines the last publication date for an item
<source>	Defines the third-party source used in an item
<title>	Specifies the title of the item
<link>	Specifies the URL of the item
<description>	Specifies synopsis for an item

Now, let’s discuss the Atom feed.

Atom Feed

Atom is a format based on XML language that is used for Web feeds. It was developed by Internet Engineering Task Force (IETF) Working Group. IETF is a large community of designers and researchers concerned with Internet architecture and the smooth operation of Internet Explorer. Web syndication formats, such as RSS and

Atom, are now used as an important tool to perform many tasks in various fields, such as marketing and news. Atom provides the user with the capability to export an entire blog or parts of a blog.

The development of the Atom format has provided a useful alternative to the RSS format. Atom resolves certain ambiguities of RSS, such as RSS incapability to distinguish plain text or HyperText Markup Language (HTML), by providing the capability to unambiguously label the type of content. Moreover, the syntax of Atom is designed in such a way that it can be reused outside the context of an Atom feed document. For example, it is possible to find `<atom:link>` elements, which are used to define a reference from an entry or feed to a website, used within RSS 2.0 feeds. The syntax of a sample Atom feed is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">

<title>Sample Feed</title>
  <link href="http://myWebsite.com/"/>
  <updated>2005-2-13T28:40:01Z</updated>
  <author>
       <name>Henry Wilkins</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>

  <entry>
       <title>demonstrating Atom Syntax</title>
       <link href="http://mywebsite.com/2005/2/13/atom03"/>
       <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
       <updated>2003-12-13T18:30:02Z</updated>
       <summary>descriptional text</summary>
  </entry>
</feed>
```

In the preceding syntax, the first line is the XML declaration, which specifies the XML version of a document. According to W3C, all the Atom feed files must conform to the XML 1.0 specifications. Some elements and attributes used in the preceding syntax are explained as follows:

- `<feed>`—Acts as a container for metadata and associated data of the feed. This element is a top-level element of an Atom feed document. Some noteworthy attributes of the <feed> element are:
  - `id`—Identifies the feed with the help of a unique Uniform Resource Identifier (URI) and specifies a globally unique identifier for the entry, which should not change over time.
  - `title`—Specifies a readable and striking title for the feed. This name is usually the same as the title of the associated website. The link may have a `<title>` attribute, whose value must be a `String`.
  - `updated`—Specifies the time when the feed was last modified. It is the last updated timestamp.
  - `category`—Indicates the category to which the feed belongs.
  - `contributor`—Indicates a single contributor to the feed. Multiple contributor elements can also be provided in a feed.
  - `generator`—Specifies the software used for generating and debugging the feed.
  - `icon`—Provides an iconic visual identification in the form of a small image for the feed.
  - `logo`—Provides visual identification in the form of a large image (as compared to an icon element) for the feed.
  - `rights`—Provides information about rights, such as copyrights, in a feed.
- `<entry>`—Behaves similar to the `<item>` element in RSS. It has the following attributes:
  - `id`—Identifies an entry by using a universally unique and permanent URI.
  - `title`—Contains a title for an entry. This value should not be blank.
  - `updated`—Indicates the time when noteworthy changes were last made to an entry.
  - `category`—Indicates the category to which an entry belongs.
  - `contributor`—Indicates a single contributor to the entry.
  - `published`–Specifies the time when the entry is created.

- `source`—Specifies that, if an entry is copied from one feed to another feed, then the child elements of the feed other than the <entry> elements should be preserved.
- `rights`—Provides information about copyrights held over an entry.
- `content`—Contains either the content or the links to the content for an entry.
- `summary`—Contains an abstract of the data contained in an entry.

After knowing about RSS and Atom feeds, let's learn how to create and use an RSS feed in a website.

Creating an RSS Feed

Perform the following steps to create an RSS feed for a website:

1. Create a website and name it `CreateRSSFeed`. You can find this website as CreateRSSFeedCS (in C#) and CreateRSSFeedVB (in VB) in the CD-ROM.
2. Now, create a new database named `rsssample`. Next, execute the following code, given in Listing B.1 of the SQL script (also available as `SQLQuery1` in the CD-ROM):

Listing B.1: Showing the Code for the `rsssample` Database

```
CREATE TABLE [newsdata] (
   [newsid] [smallint] IDENTITY (1, 1) NOT NULL ,
   [newstitle] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
   [newsdescription] [varchar] (5000) COLLATE SQL_Latin1_General_CP1_CI_AS
   NULL ,
   [newsauthor] [varchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
   [newsdate] [datetime] NULL ,
    PRIMARY KEY  CLUSTERED
   (
          [newsid]
   )  ON [PRIMARY]
) ON [PRIMARY]
GO
```

3. Now, create a table named as newsdata in the rsssample. Specify values for all the fields in the table.
4. After creating the `newsdata` table, insert some records to create an RSS feed. Now, replace the code for the `Default.aspx` file of the `CreateRSSFeed` website with the following code to add the controls required for the `CreateRSSFeed` website. You can find the code for the `Default.aspx` page in Listing B.2:

Listing B.2: Showing the Code for the `Default.aspx` Page

In VB

```
<%@ Page Title="Home Page" Language="VB" MasterPageFile="~/Site.Master"
   AutoEventWireup="false"
     CodeFile="Default.aspx.vb" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
          ASP.NET 4.0 Black BOOK
          </h2>
           <p>
          &nbsp;</p>
          <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
           &nbsp;<div id="footer">
      </asp:Content>
```

In C#

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master"
   AutoEventWireup="true"
     CodeFile="Default.aspx.cs" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
```

```
        <h2>
            ASP.NET 4.0 Black BOOK
            </h2>
    <p>
            &nbsp;</p>
            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
             &nbsp;<div id="footer">
        </asp:Content>
```

5. Add the code in the code-behind file of the Default.aspx page, as shown in Listing B.3:

Listing B.3: Showing the Code for the Code-Behind File of the `Default.aspx Page`

In VB

```
    Imports System
    Imports System.Configuration
    Imports System.Data
    Imports System.Linq
    Imports System.Web
    Imports System.Web.Security
    Imports System.Web.UI
    Imports System.Web.UI.HtmlControls
    Imports System.Web.UI.WebControls
    Imports System.Web.UI.WebControls.WebParts
    Imports System.Xml.Linq
    Imports System.Data.SqlClient
    Imports System.Text
    Imports System.Xml

    Partial Class _Default
        Inherits System.Web.UI.Page
        Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs)
            Try

                Response.Clear()
                Response.ContentType = "text/xml"
                Dim XMLTextObject As New XmlTextWriter(Response.OutputStream, Encoding.UTF8)
                XMLTextObject.WriteStartDocument()
                XMLTextObject.WriteStartElement("rss")
                XMLTextObject.WriteAttributeString("version", "2.0")
                XMLTextObject.WriteStartElement("channel")
                XMLTextObject.WriteElementString("title", "Creating RSS Feed On a Website")
                XMLTextObject.WriteElementString("link",
                     "http://www.somedomain.com/news.aspx")
                XMLTextObject.WriteElementString("description", "This is a sample RSS 2.0
                     Feed")
                XMLTextObject.WriteElementString("copyright", "(c) 2006,  All rights
                     reserved.")
                XMLTextObject.WriteElementString("ttl", "5")
                Dim connectionstring As String = "Data Source=.\sqlexpress;Initial
                     Catalog=rsssample;Integrated Security=True"
                Dim objConnection As New SqlConnection(connectionstring)
                objConnection.Open()
                Dim sql As String = "SELECT TOP 3 newstitle, newsdescription, newsdate FROM
                     newsdata ORDER BY newsdate DESC"
                Dim objCommand As New SqlCommand(sql, objConnection)
                Dim objReader As SqlDataReader = objCommand.ExecuteReader()
                Do While objReader.Read()
                    XMLTextObject.WriteStartElement("item")
                    XMLTextObject.WriteElementString("title", objReader.GetString(0))
                    XMLTextObject.WriteElementString("description", objReader.GetString(1))
                    XMLTextObject.WriteElementString("link",
                    "http://www.somedomain.com/GetArticle.aspx?id=0")
                    XMLTextObject.WriteElementString("pubDate",
                    objReader.GetDateTime(2).ToString("R"))
                    XMLTextObject.WriteEndElement()
                Loop
                objReader.Close()
                objConnection.Close()
                XMLTextObject.WriteEndElement()
                XMLTextObject.WriteEndElement()
                XMLTextObject.WriteEndDocument()
                XMLTextObject.Flush()
                XMLTextObject.Close()
                Response.End()
            Catch ex As Exception
                Label1.Text = ex.Message
            End Try
        End Sub
    End Class
```

In C#

```
    using System;
    using System.Configuration;
    using System.Data;
    using System.Linq;
    using System.Web;
    using System.Web.Security;
    using System.Web.UI;
    using System.Web.UI.HtmlControls;
    using System.Web.UI.WebControls;
    using System.Web.UI.WebControls.WebParts;
    using System.Xml.Linq;
    using System.Data.SqlClient;
    using System.Text;
    using System.Xml;

    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            try
             {

                     Response.Clear();
                     Response.ContentType = "text/xml";
                     XmlTextWriter XMLTextObject = new XmlTextWriter(Response.OutputStream,
                     Encoding.UTF8);
                     XMLTextObject.WriteStartDocument();
                     XMLTextObject.WriteStartElement("rss");
                     XMLTextObject.WriteAttributeString("version", "2.0");
                     XMLTextObject.WriteStartElement("channel");
                     XMLTextObject.WriteElementString("title", "Sample News Channel");
                     XMLTextObject.WriteElementString("link",
                     "http://www.somedomain.com/news.aspx");
                     XMLTextObject.WriteElementString("description", "This is a sample RSS 2.0
                     Feed");
                     XMLTextObject.WriteElementString("copyright", "(c) 2006,  All rights
                     reserved.");
                     XMLTextObject.WriteElementString("ttl", "5");
                     string connectionstring = "Data Source=.\\sqlexpress;Initial
                     Catalog=rsssample;Integrated Security=True";
                     SqlConnection objConnection = new SqlConnection(connectionstring);
                     objConnection.Open();
                     string sql = "SELECT TOP 3 newstitle, newsdescription, newsdate FROM
                     newsdata ORDER BY newsdate DESC";
                     SqlCommand objCommand = new SqlCommand(sql, objConnection);
                     SqlDataReader objReader = objCommand.ExecuteReader();
                     while (objReader.Read())
                     {
                             XMLTextObject.WriteStartElement("item");
                             XMLTextObject.WriteElementString("title",
                             objReader.GetString(0));
```

```
                              XMLTextObject.WriteElementString("description",
                              objReader.GetString(1));
                              XMLTextObject.WriteElementString("link",
                              "http://www.somedomain.com/GetArticle.aspx?id=0");
                              XMLTextObject.WriteElementString("pubDate",
                              objReader.GetDateTime(2).ToString("R"));
                              XMLTextObject.WriteEndElement();
                      }
                      objReader.Close();
                      objConnection.Close();

                      XMLTextObject.WriteEndElement();
                      XMLTextObject.WriteEndElement();
                      XMLTextObject.WriteEndDocument();
                      XMLTextObject.Flush();
                      XMLTextObject.Close();
                      Response.End();
              }
              catch (Exception ex)
              {
                      Label1.Text = ex.Message;
              }
      }
    }
```

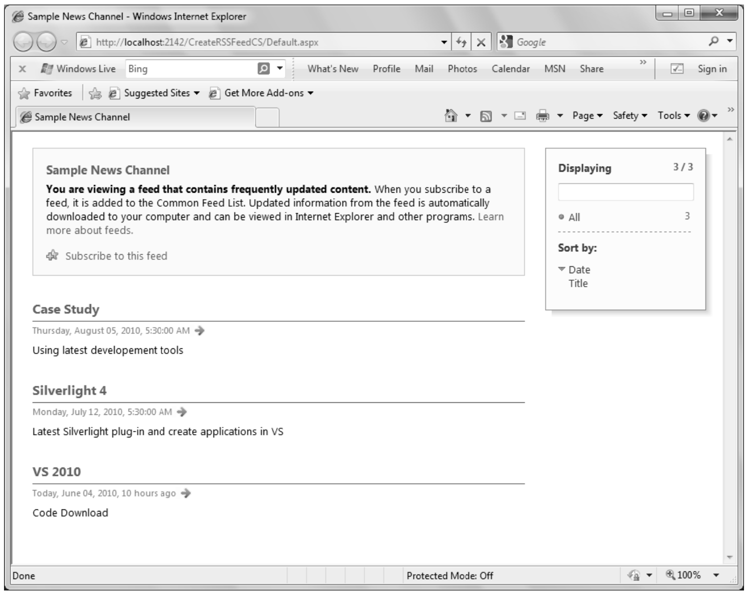
6. In Listing B.3, the following syntax is used to generate a query to retrieve the top three rows of data from the `newsdata` table:

```
    string sql = "SELECT TOP 3 newstitle, newsdescription, newsdate FROM newsdata ORDER BY
      newsdate DESC";
```

In case of the CreateRSSFeedVB website, we have also added the following connection string in the `web.config` file of the website:

```
    <connectionStrings>
      <add name="rsssampleConnectionString" connectionstring = "Data
      Source=.\\sqlexpress;Initial Catalog=rsssample;Integrated Security=True"/>
    </connectionStrings>
```

7. Now, press the F5 key on the keyboard to run the CreateRSSFeed website. The output is shown in Figure B.1:



Sample News Channel

You are viewing a feed that contains frequently updated content. When you subscribe to a feed, it is added to the Common Feed List. Updated information from the feed is automatically downloaded to your computer and can be viewed in Internet Explorer and other programs. Learn more about feeds.

Subscribe to this feed

Displaying 3 / 3

All 3

Sort by:

Date

Title

Case Study

Thursday, August 05, 2010, 5:30:00 AM

Using latest developement tools

Silverlight 4

Monday, July 12, 2010, 5:30:00 AM

Latest Silverlight plug-in and create applications in VS

VS 2010

Today, June 04, 2010, 10 hours ago

Code Download

Figure B.1: Output of the CreateRSSFeed Example

NOTE

The data displayed on the webpage is based on the data in the newsdata table

Consuming an RSS Feed

Numerous websites and blogs share their content through RSS feeds over the Internet. Now, let's understand how a website consumes an RSS feed.

Perform the following steps to see how a website can consume an RSS feed:

1. Create a website and name it ConsumeRSSFeed. You can find this website as ConsumeRSSFeedCS (in C#) and ConsumeRSSFeedVB (in VB) in the CD-ROM.
2. Next, replace the code for the `Default.aspx` file with the following code to add the controls required for the ConsumeRSSFeed website. You can find the code for the `Default.aspx` page in Listing B.4:

Listing B.4: Showing the Code for the `Default.aspx` Page

In VB

```
<%@ Page Title="Home Page" Language="VB" MasterPageFile="~/Site.Master"
  AutoEventWireup="false"
    CodeFile="Default.aspx.vb" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        ASP.NET 4.0 Black Book</h2>
        <asp:Label ID="Label1" runat="server" Font-Bold="True"
                                 Text="CNN.com RSS Feed"></asp:Label><br />
                                 <br />
                                 <asp:DataList ID="DataList1" runat="server"
                                 DataSourceID="XmlDataSource1" BackColor="White"
                                         BorderColor="#999999" BorderStyle="None"
                                         BorderWidth="1px" CellPadding="3"
                                         GridLines="Vertical"
                                         Font-Size="XX-Small" Width="308px">
                                         <ItemTemplate>
                                                 <%#XPath("title")%>
                                                 <%#XPath("description")%>
                                                 <%#XPath("link")%>
                                         </ItemTemplate>
                                         <FooterStyle BackColor="#CCCCCC"
                                         ForeColor="Black" />
                                         <SelectedItemStyle BackColor="#008A8C" Font-
                                         Bold="True" ForeColor="White" />
                                         <AlternatingItemStyle BackColor="Tan" />
                                         <ItemStyle BackColor="#EEEEEE" ForeColor="Black"
                                         />
                                         <HeaderStyle BackColor="#000084" Font-Bold="True"
                                         ForeColor="White" />
                                 </asp:DataList><br />
                                 <asp:XmlDataSource ID="XmlDataSource1" runat="server"
                                 DataFile="http://rss.cnn.com/rss/cnn_topstories.rss"
                                         XPath="rss/channel/item"></asp:XmlDataSource>
                                 &nbsp;

    </asp:Content>
```

In C#

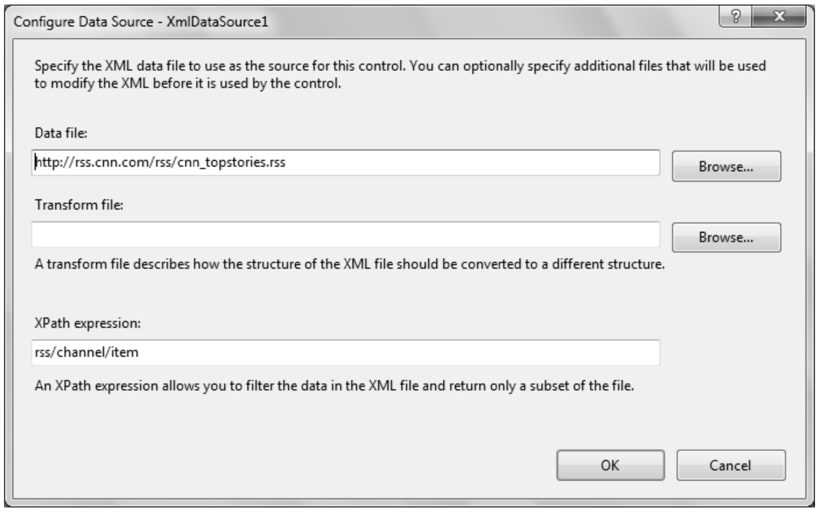
```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master"
  AutoEventWireup="true"
    CodeFile="Default.aspx.cs" Inherits="_Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
```

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        ASP.NET 4.0 Black Book</h2>
        <asp:Label ID="Label1" runat="server" Font-Bold="True"
                                 Text="CNN.com RSS Feed"></asp:Label><br />
                                 <br />
                                 <asp:DataList ID="DataList1" runat="server"
                                 DataSourceID="XmlDataSource1" BackColor="White"
                                         BorderColor="#999999" BorderStyle="None"
                                         BorderWidth="1px" CellPadding="3"
                                         GridLines="Vertical"
                                         Font-Size="XX-Small" Width="308px">
                                         <ItemTemplate>
                                                 <%#XPath("title")%>
                                                 <%#XPath("description")%>
                                                 <%#XPath("link")%>
                                         </ItemTemplate>
                                         <FooterStyle BackColor="#CCCCCC"
                                         ForeColor="Black" />
                                         <SelectedItemStyle BackColor="#008A8C" Font-
                                         Bold="True" ForeColor="White" />
                                         <AlternatingItemStyle BackColor="Tan" />
                                         <ItemStyle BackColor="#EEEEEE" ForeColor="Black"
                                         />
                                         <HeaderStyle BackColor="#000084" Font-Bold="True"
                                         ForeColor="White" />
                                 </asp:DataList><br />
                                 <asp:XmlDataSource ID="XmlDataSource1" runat="server"
                                 DataFile="http://rss.cnn.com/rss/cnn_topstories.rss"
                                         XPath="rss/channel/item"></asp:XmlDataSource>
                                 &nbsp;

    </asp:Content>
```

3. Next, specify the path of the RSS feed in the Data file text box in the Configure Data Source dialog box while configuring the data source to display the data in the ConsumeRSSFeed website, as shown in Figure B.2:


Configure Data Source - XmlDataSource1

Specify the XML data file to use as the source for this control. You can optionally specify additional files that will be used to modify the XML before it is used by the control.

Data file: http://rss.cnn.com/rss/cnn_topstories.rss

Transform file:

A transform file describes how the structure of the XML file should be converted to a different structure.

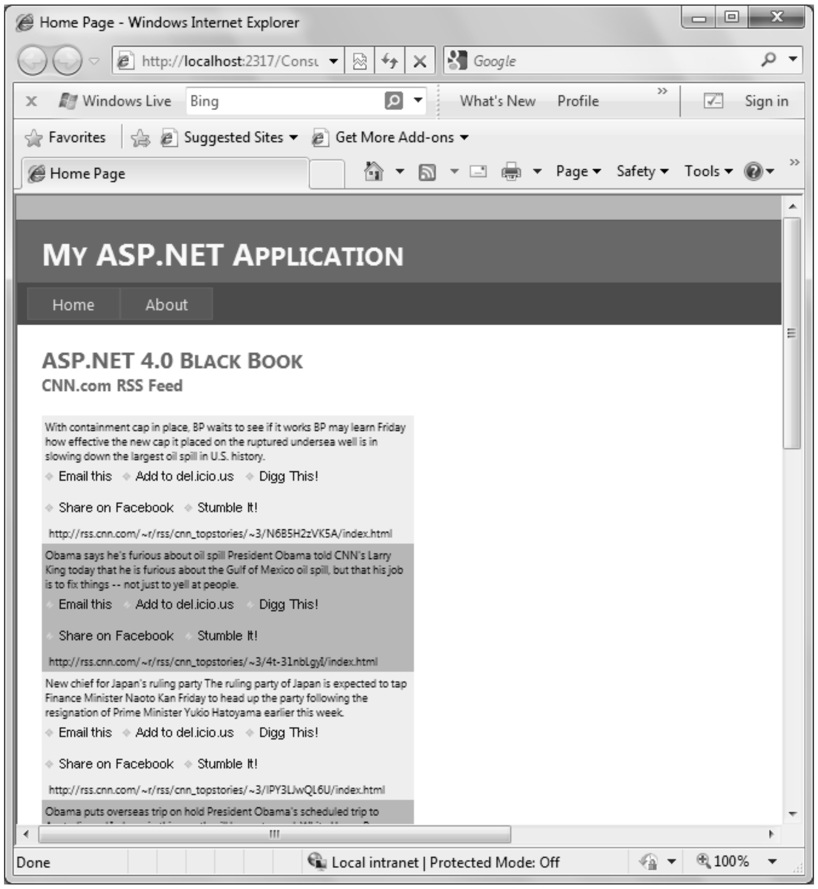
XPath expression: rss/channel/item

An XPath expression allows you to filter the data in the XML file and return only a subset of the file.

Figure B.2: Displaying the Configure Data Source Dialog Box

4. Similarly, in the `Transform file` text box of the Configure Data Source dialog box, you can provide the path of the `.xls` file (Figure B.2). With XSLT, you can transform an XML document into HTML to help you to provide a different structure to an XML file.

5. In the XPath expression text box, specify an optional XPath expression that allows you to filter the data in the XML file to return a subset of the file (Figure B.2).
6. Now, press the F5 key on the keyboard to run the ConsumeRSSFeed website. The output is shown in Figure B.3:



ASP.NET 4.0 BLACK BOOK

CNN.com RSS Feed

With containment cap in place, BP waits to see if it works BP may learn Friday how effective the new cap it placed on the ruptured undersea well is in slowing down the largest oil spill in U.S. history.

Email this   Add to del.icio.us   Digg This!

Share on Facebook   Stumble It!

http://rss.cnn.com/~r/rss/cnn_topstories/~3/N6B5H2zVK5A/index.html

Obama says he's furious about oil spill President Obama told CNN's Larry King today that he is furious about the Gulf of Mexico oil spill, but that his job is to fix things -- not just to yell at people.

Email this   Add to del.icio.us   Digg This!

Share on Facebook   Stumble It!

http://rss.cnn.com/~r/rss/cnn_topstories/~3/4t-31nbLgyI/index.html

New chief for Japan's ruling party The ruling party of Japan is expected to tap Finance Minister Naoto Kan Friday to head up the party following the resignation of Prime Minister Yukio Hatoyama earlier this week.

Email this   Add to del.icio.us   Digg This!

Share on Facebook   Stumble It!

http://rss.cnn.com/~r/rss/cnn_topstories/~3/IPY3LJwQL6U/index.html

Obama puts overseas trip on hold President Obama's scheduled trip to

Figure B.3: Output of the ConsumeRssFeed Website

In this way, you can implement RSS and Atom Feed in the ASP.NET 4.0 applications.