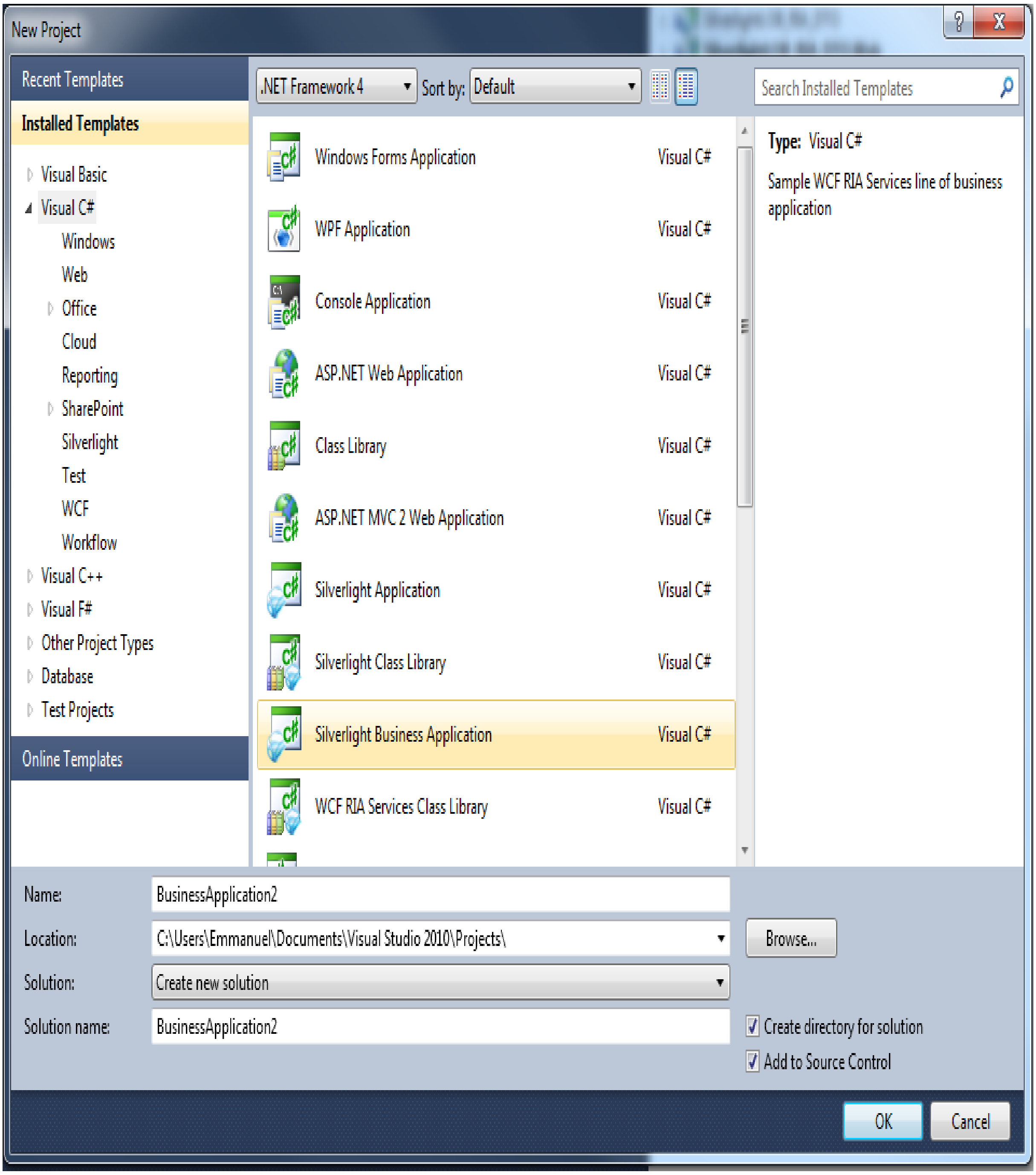
Using SilverLight 4 Line Of Business Application Development Template (LOB)

1. Create en empty SilverLight Business Template Solution


New Project
Recent Templates
Installed Templates
Visual Basic
Visual C#
Windows
Web
Office
Cloud
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
Visual C++
Visual F#
Other Project Types
Database
Test Projects
Online Templates
.NET Framework 4
Sort by: Default
Search Installed Templates
Windows Forms Application Visual C#
WPF Application Visual C#
Console Application Visual C#
ASP.NET Web Application Visual C#
Class Library Visual C#
ASP.NET MVC 2 Web Application Visual C#
Silverlight Application Visual C#
Silverlight Class Library Visual C#
Silverlight Business Application Visual C#
WCF RIA Services Class Library Visual C#
Browse...
OK
Cancel

Type: Visual C#

Sample WCF RIA Services line of business application

Name: BusinessApplication2

Location: C:\Users\Emmanuel\Documents\Visual Studio 2010\Projects\

Solution: Create new solution

Solution name: BusinessApplication2

☑ Create directory for solution

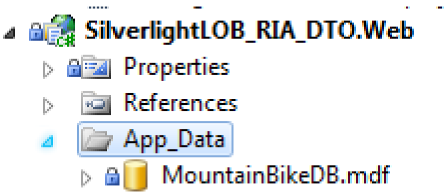
☑ Add to Source Control

Add a new Silverlight Business Application and call it **Silverlight_LOB_RIA_DTO.**

2. ServerSide

The Serverside contains the web-application which will ghost our SilverLight Client application. The is the **SilverLightLOB_RIA_DTO_Web** project.

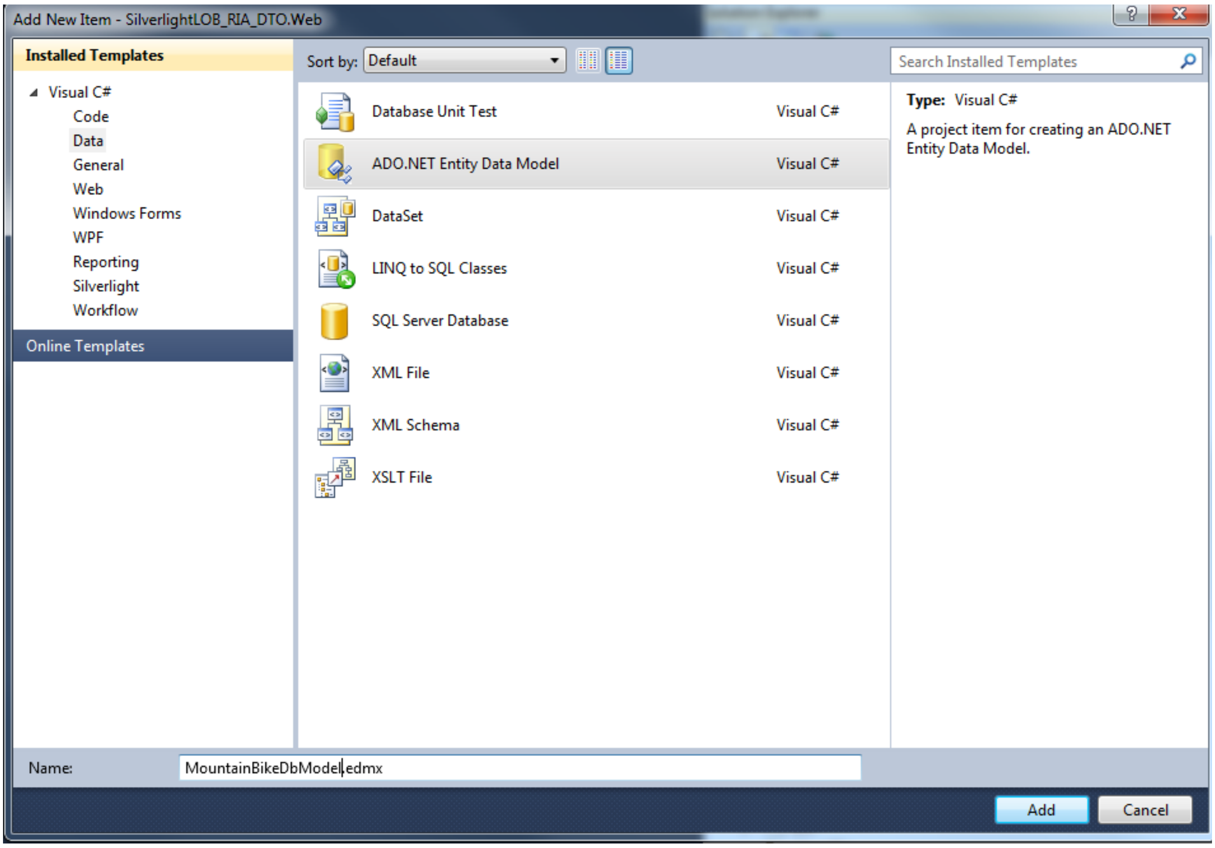
2.1. Adding the Database



First you have to add the **MountainBike.mdf** database file, these is included in the project source. Just make sur that the **App_Data** folder is the current one, next **right click** the folder, select **add existing item** and choose the database.

2.2. Adding the Entity Model

From the Database we will generate the EntityModel. First create a new folder called **EntityModel** on the WebSite project. Next right-click the folder, choose **add new item** from the list, and choose **ADO.NET Entity Data Model** from the installed templates.

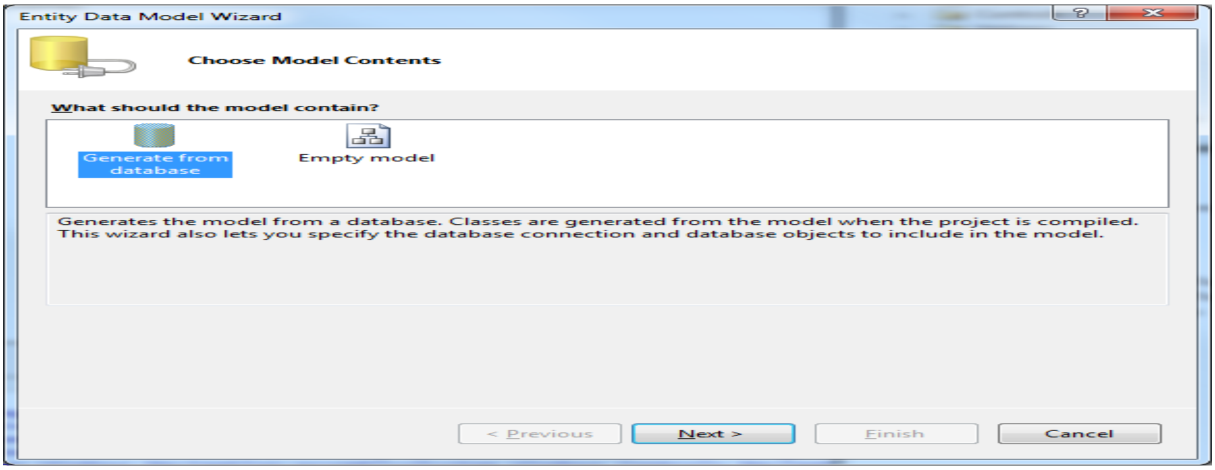


Sort by: Default

Type: Visual C#

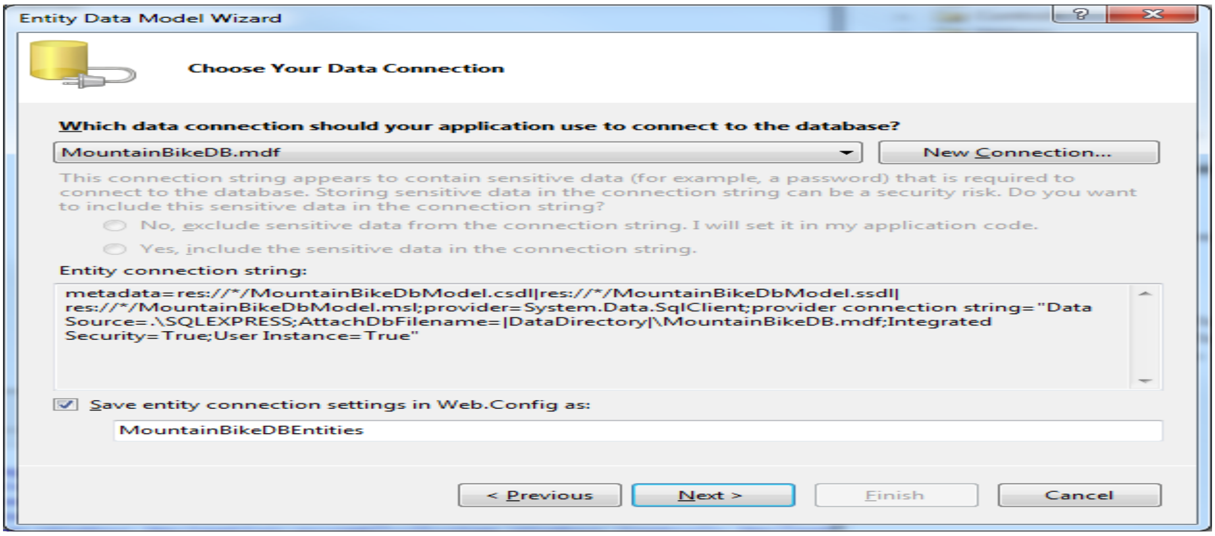
A project item for creating an ADO.NET Entity Data Model.

Name: MountainBikeDbModel.edmx



Generates the model from a database. Classes are generated from the model when the project is compiled. This wizard also lets you specify the database connection and database objects to include in the model.

Let the default option **Genereate from database** and click **“Next”.**



Which data connection should your application use to connect to the database?

MountainBikeDB.mdf

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

○ No, exclude sensitive data from the connection string. I will set it in my application code.

○ Yes, include the sensitive data in the connection string.

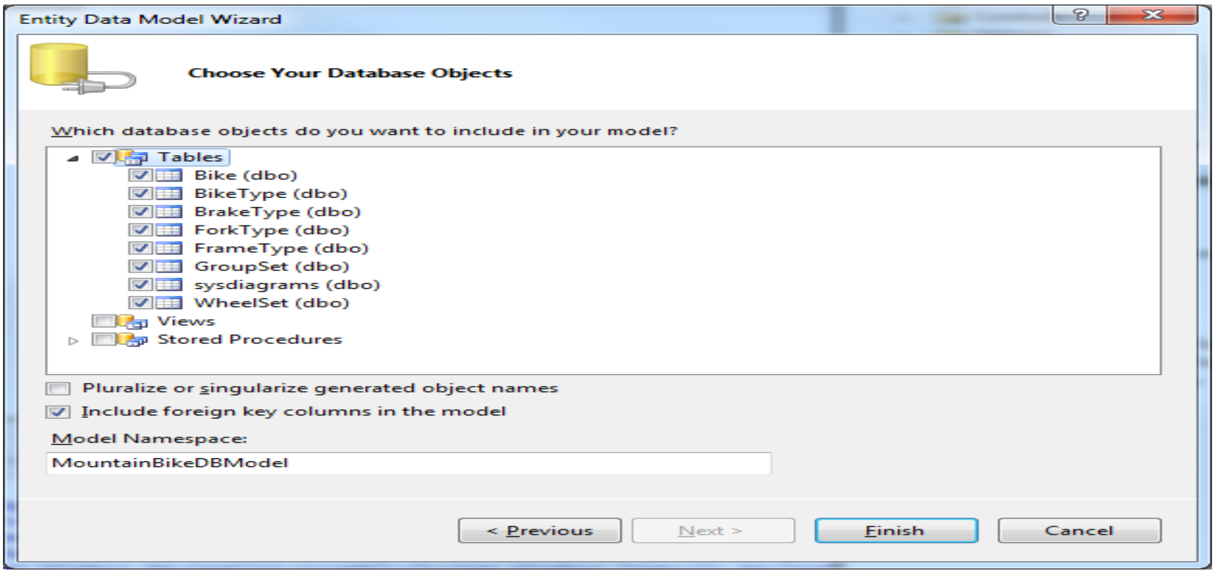
Entity connection string:

```
metadata=res://*/MountainBikeDbModel.csdl|res://*/MountainBikeDbModel.ssdl|
res://*/MountainBikeDbModel.msl;provider=System.Data.SqlClient;provider connection string="Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\MountainBikeDB.mdf;Integrated
Security=True;User Instance=True"
```

☑ Save entity connection settings in Web.Config as:

MountainBikeDBEntities

As we added our **MountainbikeDB** database to the solution, it will be selected by default, juist click **“Next”** to proceed.



Which database objects do you want to include in your model?

- ☑ Tables
  - ☑ Bike (dbo)
  - ☑ BikeType (dbo)
  - ☑ BrakeType (dbo)
  - ☑ ForkType (dbo)
  - ☑ FrameType (dbo)
  - ☑ GroupSet (dbo)
  - ☑ sysdiagrams (dbo)
  - ☑ WheelSet (dbo)
- □ Views
- □ Stored Procedures

□ Pluralize or singularize generated object names

☑ Include foreign key columns in the model

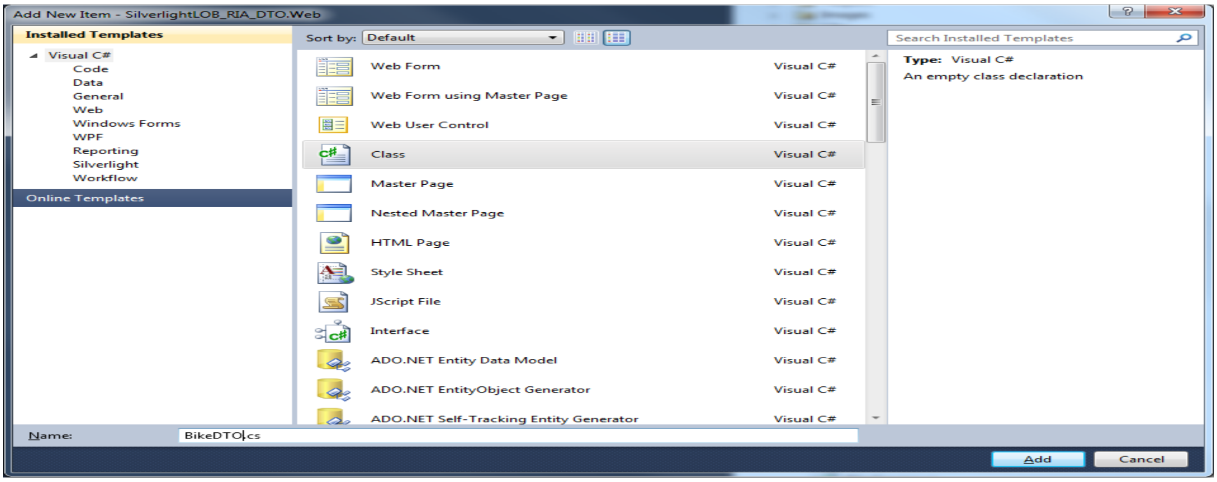
Model Namespace:

MountainBikeDBModel

Check the **Tables** item from the treeview and click **“Finish”.**

2.3. Creating the DTO Class

Now our database en entitymodel is in place, we will first create our DTO (Data Transfert Objects) class which will hold the bike properties to be bound to our ListBox on the SilverLight client application. First select the **/Models** folder, right-click and select **Class** from the installed templates (as shown here under). The class should be called **BikeDTO.**



Sort by: Default

Type: Visual C#

An empty class declaration

Name: BikeDTO.cs

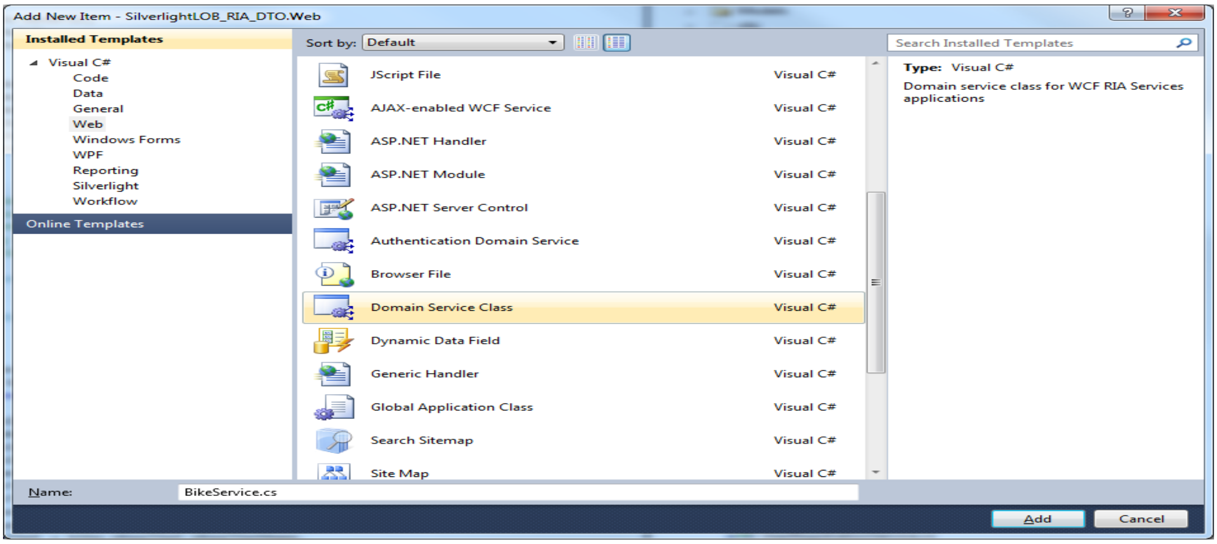
Open the class and add the properties, the final version of the class should look as next:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;

namespace SilverlightLOB_RIA_DTO.Web.Models
{
    public partial class BikeDTO
    {
        [Key]
        [Editable(false)]
        public int BikeID { get; set; }
        public string BikeName { get; set; }
        public byte[] BikeImage { get; set; }
        public decimal ModelNumber { get; set; }
        public string FrameType { get; set; }
        public string BikeType { get; set; }
        public string ForkType { get; set; }
        public string BrakeType { get; set; }
        public string GroupSet { get; set; }
        public string WheelSet { get; set; }
        public int? InStock { get; set; }
    }
}
```

You will notice the, while adding the **[Key]** attribute, you will have to add the **DataAnnotations** namespace. This can easly be done by right-clicking the Key attribute and select the namespace from the list.

2.4. Creating the DomainService

The DomainService class will be responsible for quering our entitymodel bike data and mapping them to the DTO class. First select the **/Services** folder in the Web project, right-click the folder and select **“Add – New Item”.**


Add New Item - SilverlightLOB_RIA_DTO.Web
Installed Templates
Visual C#
Code
Data
General
Web
Windows Forms
WPF
Reporting
Silverlight
Workflow
Online Templates
Sort by: Default
Search Installed Templates
JScript File Visual C#
AJAX-enabled WCF Service Visual C#
ASP.NET Handler Visual C#
ASP.NET Module Visual C#
ASP.NET Server Control Visual C#
Authentication Domain Service Visual C#
Browser File Visual C#
Domain Service Class Visual C#
Dynamic Data Field Visual C#
Generic Handler Visual C#
Global Application Class Visual C#
Search Sitemap Visual C#
Site Map Visual C#
Add
Cancel

Type: Visual C#
Domain service class for WCF RIA Services applications

Name: BikeService.cs

Select **Domain Service Class,** name it **BikeService** and click the **“Add”** button. The final implementation of the class should look like next (you have to create the code yourself here!:

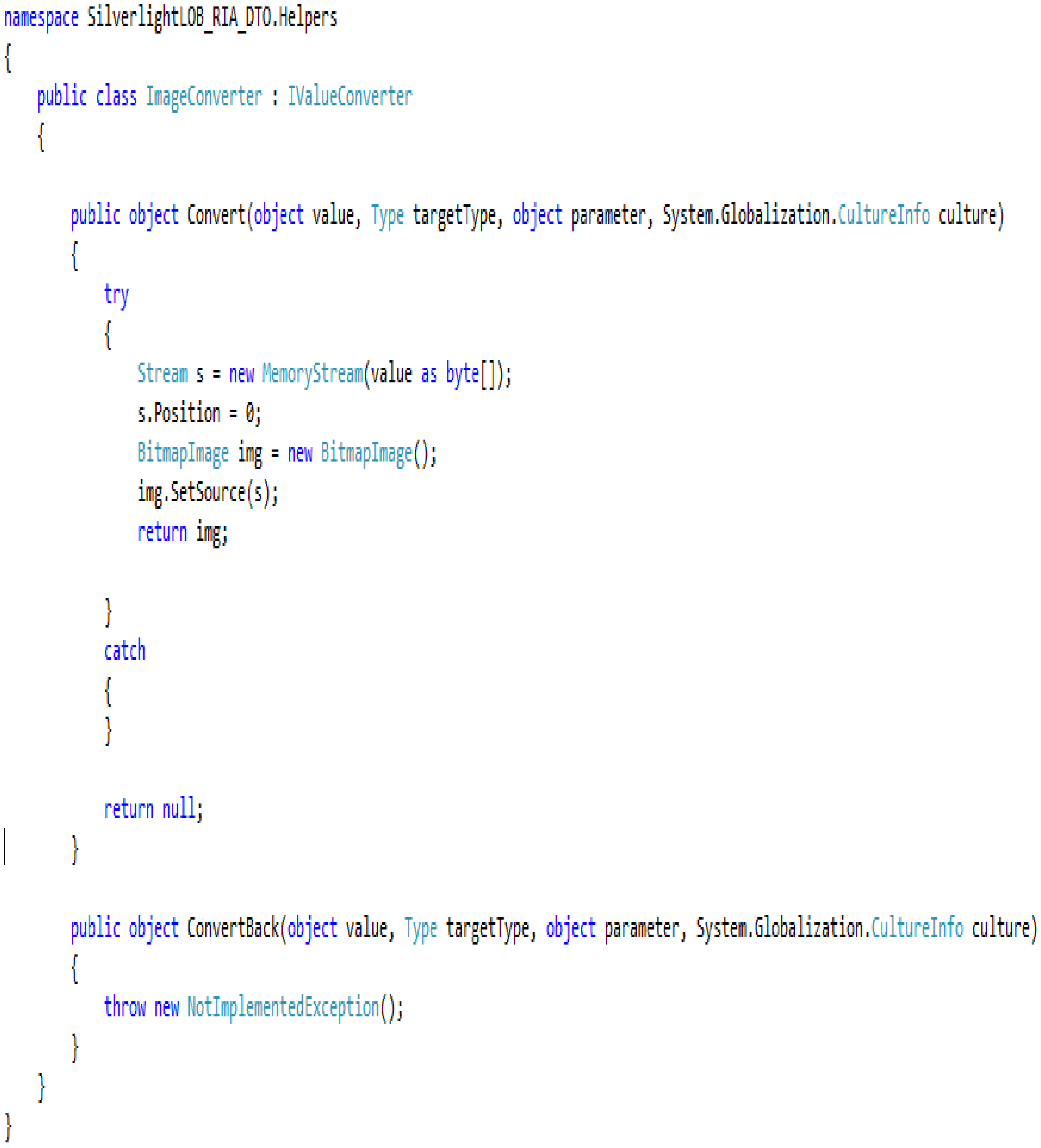
```
[EnableClientAccess()]
public class BikeService : LinqToEntitiesDomainService<MountainBikeDBEntities>
{
    public IQueryable<BikeDTO> GetBikeDTOList()
    {


        return from bike in this.ObjectContext.Bikes
               orderby bike.BikeName
               select new BikeDTO
               {
                   BikeID = bike.BikeID,
                   BikeName = bike.BikeName,
                   BikeImage = bike.Photo,
                   ModelNumber = bike.ModelNumber,
                   FrameType = bike.FrameType.FrameTypeName,
                   BikeType = bike.BikeType.BikeTypeName,
                   ForkType = bike.ForkType.ForkTypeName,
                   BrakeType = bike.BrakeType.BrakeTypeName,
                   GroupSet = bike.GroupSet.GroupSetName,
                   WheelSet = bike.WheelSet.WheelSetName,
                   InStock = bike.InStock

               };

    }
}
```

This service method will query our entitymodel and create and will map each bike-entity to a DTO object. Finaly a list of Bike DTO objects will be returned to the client. Our serverside part is configured now, lets move to the client part !

3. ClientSide

3.1. Add Image Converter

The first thing we do is adding an **ImageConverter** class to our **/Helpers** section. This converter is necessary to make binding of the BikeImage (returned as Byte[]) possible. This class implements the **IValueConverter** interface and will take a **byte[]** as input parameter and return an **Image** as output parameter. This image can then be used to bind in the XAML code of our page.

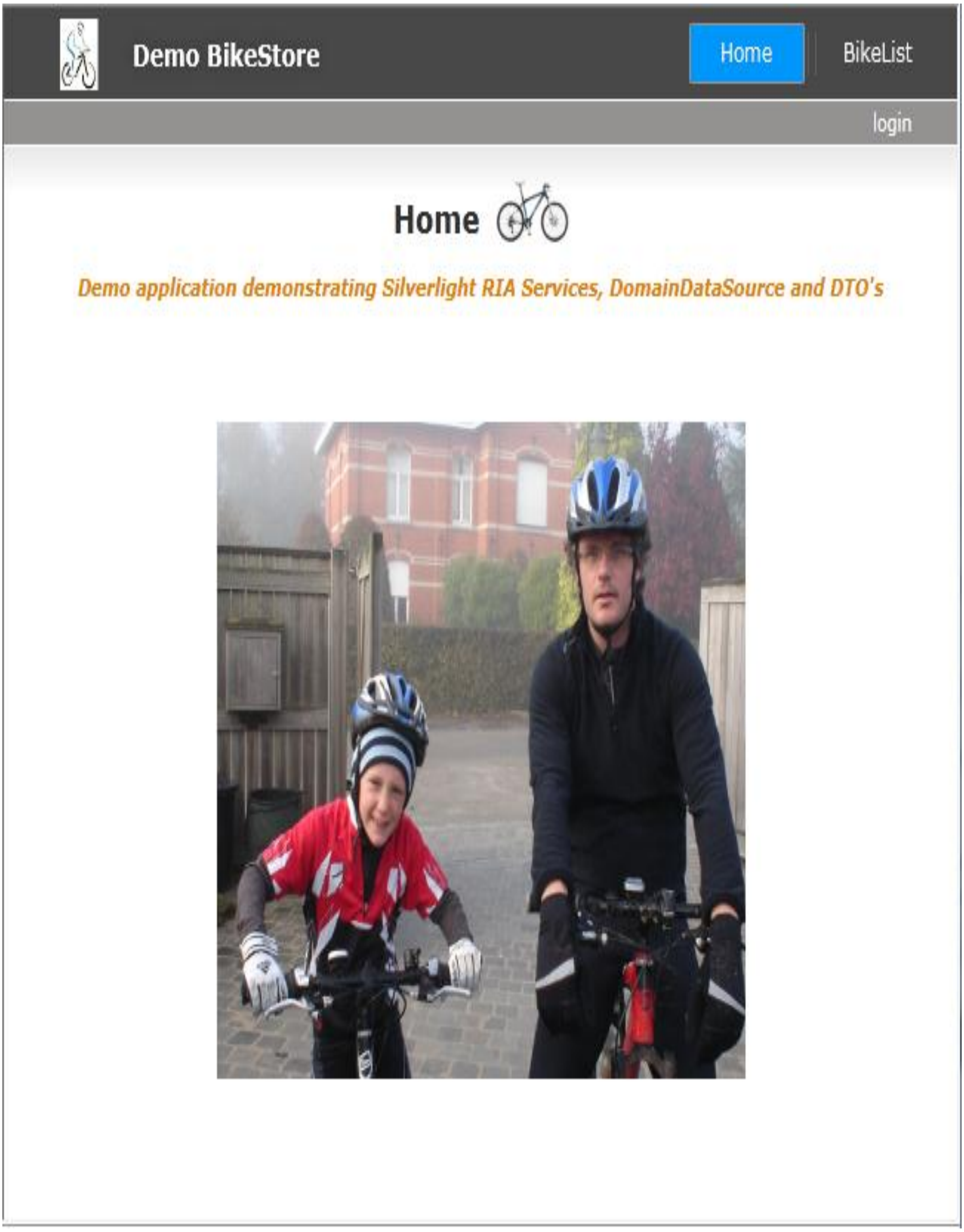


```
namespace SilverlightLOB_RIA_DTO.Helpers
{
    public class ImageConverter : IValueConverter
    {

        public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
        {
            try
            {
                Stream s = new MemoryStream(value as byte[]);
                s.Position = 0;
                BitmapImage img = new BitmapImage();
                img.SetSource(s);
                return img;

            }
            catch
            {
            }

            return null;
        }

        public object ConvertBack(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
        {
            throw new NotImplementedException();
        }
    }
}
```

When you will look (see later …) add the XAML code of our **BikeListView** page, you will see how the valueconverter is included as **static resource** of the page and is used in the Binding of the Image.

3.2. Home Page

When the application start's up, the **"Home Page"** is shown, I've changed the page a bit so it mimicks a startup page for our demo webshop. (you see a picture of my youngest son and I).


Demo BikeStore
Home
BikeList
login
Home
Demo application demonstrating Silverlight RIA Services, DomainDataSource and DTO's

The XAML-code for the **home** page:

```
<navigation:Page
  x:Class="SilverlightLOB_RIA_DTO.Home"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:navigation="clr-
namespace:System.Windows.Controls;assembly=System.Windows.Controls.Navigation"
  mc:Ignorable="d" d:DesignWidth="640" d:DesignHeight="480"
  Style="{StaticResource PageStyle}">

  <Grid x:Name="LayoutRoot">

        <ScrollViewer x:Name="PageScrollViewer" Style="{StaticResource
PageScrollViewerStyle}" >

      <StackPanel x:Name="ContentStackPanel" Style="{StaticResource
ContentStackPanelStyle}">

                <StackPanel Orientation="Horizontal" HorizontalAlignment="Center">
                    <TextBlock x:Name="HeaderText" Style="{StaticResource
HeaderTextStyle}"
                           Text="{Binding Path=ApplicationStrings.HomePageTitle,
Source={StaticResource ResourceWrapper}}" FontSize="18" />
                    <TextBlock Width="10"/>
                    <Image Width="50" Height="50" Stretch="Fill"
Source="/SilverlightLOB_RIA_DTO;component/Images/Scale%2080.png" />
                </StackPanel>
                <TextBlock x:Name="ContentText" Style="{StaticResource
ContentTextStyle}"
                           Text="Demo application demonstrating Silverlight RIA
Services, DomainDataSource and DTO's" FontWeight="Bold" FontStyle="Italic"
Foreground="#FFE08100" FontSize="11" HorizontalAlignment="Center" />
       <StackPanel Orientation="Horizontal" Height="50"/>
                <StackPanel Orientation="Horizontal" Height="306" Width="357">
                    <Image Width="357" Height="303"
Source="/SilverlightLOB_RIA_DTO;component/Images/DSC00002.JPG" Stretch="Fill" />
                </StackPanel>
        </StackPanel>
     </ScrollViewer>
  </Grid>

</navigation:Page>
```

3.3. Adding the BikeList view

The **BikeListView** page is a new page, so add a new Page to the **/Views** folder and call it **BikeListView.** The full XAML-Code is :

```
<navigation:Page x:Class="SilverlightLOB_RIA_DTO.Views.BikeListView"
           xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
           xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
           xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
           xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
           mc:Ignorable="d"
           xmlns:navigation="clr-
namespace:System.Windows.Controls;assembly=System.Windows.Controls.Navigation"
           d:DesignWidth="640" d:DesignHeight="682"
           Title="BikeListXAMLView Page"
           xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk"
           xmlns:riaControls="clr-
namespace:System.Windows.Controls;assembly=System.Windows.Controls.DomainServices"
           xmlns:my="clr-namespace:SilverlightLOB_RIA_DTO.Web.Models"
           xmlns:my1="clr-namespace:SilverlightLOB_RIA_DTO.Web.Services"
           xmlns:helpers="clr-namespace:SilverlightLOB_RIA_DTO.Helpers"
           xmlns:localControls="clr-namespace:SilverlightLOB_RIA_DTO.Controls">

    <navigation:Page.Resources>

        <helpers:ImageConverter x:Key="_imageConverter"/>

        <DataTemplate x:Key="_bikeItemTemplate">
            <Grid Height="250" HorizontalAlignment="Left" Margin="160,23,0,0"
Name="grid1" VerticalAlignment="Top" Width="433">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="152*" />
                    <ColumnDefinition Width="148*" />
                </Grid.ColumnDefinitions>
                <Grid.RowDefinitions>
                    <RowDefinition Height="25*" />
                    <RowDefinition Height="25*" />
                    <RowDefinition Height="25*" />
                    <RowDefinition Height="25*" />
                    <RowDefinition Height="25*" />
                    <RowDefinition Height="25*" />
                    <RowDefinition Height="25*" />
                    <RowDefinition Height="25*" />
                </Grid.RowDefinitions>
                <Image Grid.RowSpan="7"
                       Height="219"
                       HorizontalAlignment="Left"
                       Name="image1" Stretch="Fill"
                       VerticalAlignment="Top"
                       Width="219"
                       Source="{Binding Path=BikeImage,
Converter={StaticResource _imageConverter}}"
                       Margin="2" />
                <StackPanel Orientation="Horizontal" Grid.Column="1" Height="31"
HorizontalAlignment="Left" Name="stackPanel1" VerticalAlignment="Top" Width="214">
```

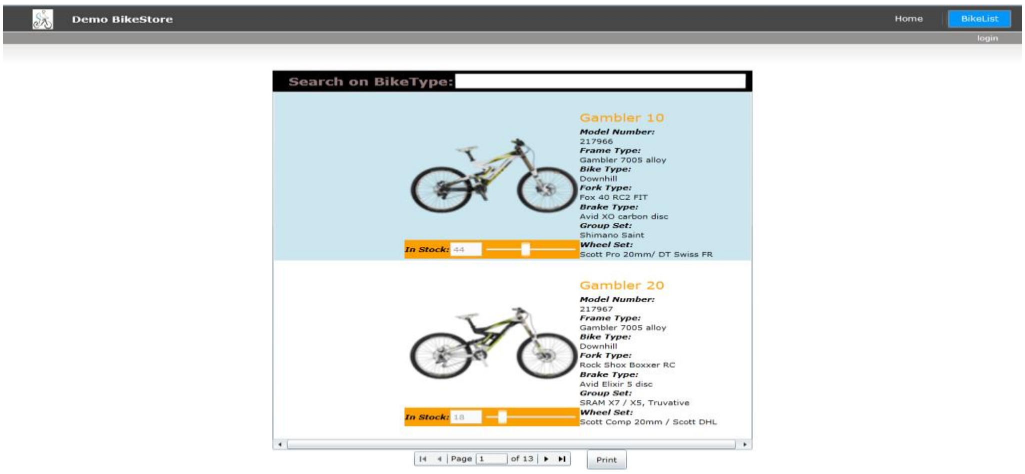
```
                    <TextBlock Text="{Binding Path=BikeName}" Width="148"
HorizontalAlignment="Center" VerticalAlignment="Center" Foreground="Orange"
FontSize="18" Name="_bikeName" />
                </StackPanel>
                <StackPanel Orientation="Vertical" Grid.Row="1" Grid.Column="1"
Height="31" HorizontalAlignment="Left" Name="stackPanel2" VerticalAlignment="Top"
Width="214">
                    <TextBlock Text="Model Number:" FontWeight="Bold"
FontStyle="Italic" />
                    <TextBlock Text="{Binding Path=ModelNumber}" Name="_modelNumber"
/>
                </StackPanel>
                <StackPanel Orientation="Vertical" Grid.Row="2" Grid.Column="1"
Height="31" HorizontalAlignment="Left" Name="stackPanel3" VerticalAlignment="Top"
Width="214">
                    <TextBlock Text="Frame Type:" FontWeight="Bold" FontStyle="Italic"
/>
                    <TextBlock Text="{Binding Path=FrameType}" Name="_frameType" />
                </StackPanel>
                <StackPanel Orientation="Vertical" Grid.Row="3" Grid.Column="1"
Height="31" HorizontalAlignment="Left" Name="stackPanel4" VerticalAlignment="Top"
Width="214">
                    <TextBlock Text="Bike Type:" FontWeight="Bold" FontStyle="Italic"
/>
                    <TextBlock Text="{Binding Path=BikeType}" Name="_bikeType" />
                </StackPanel>
                <StackPanel Orientation="Vertical" Grid.Row="4" Grid.Column="1"
Height="31" HorizontalAlignment="Left" Name="stackPanel5" VerticalAlignment="Top"
Width="214">
                    <TextBlock Text="Fork Type:" FontWeight="Bold" FontStyle="Italic"
/>
                    <TextBlock Text="{Binding Path=ForkType}" Name="_forkType" />
                </StackPanel>
                <StackPanel Orientation="Vertical" Grid.Row="5" Grid.Column="1"
Height="31" HorizontalAlignment="Left" Name="stackPanel6" VerticalAlignment="Top"
Width="214">
                    <TextBlock Text="Brake Type:" FontWeight="Bold" FontStyle="Italic"
/>
                    <TextBlock Text="{Binding Path=BrakeType}" Name="_brakeType" />
                </StackPanel>
                <StackPanel Orientation="Vertical" Grid.Row="6" Grid.Column="1"
Height="31" HorizontalAlignment="Left" Name="stackPanel7" VerticalAlignment="Top"
Width="214">
                    <TextBlock Text="Group Set:" FontWeight="Bold" FontStyle="Italic"
/>
                    <TextBlock Text="{Binding Path=GroupSet}" Name="_groupSet" />
                </StackPanel>
                <StackPanel Orientation="Vertical" Grid.Row="7" Grid.Column="1"
Height="31" HorizontalAlignment="Left" Name="stackPanel8" VerticalAlignment="Top"
Width="214">
                    <TextBlock Text="Wheel Set:" FontWeight="Bold" FontStyle="Italic"
/>
                    <TextBlock Text="{Binding Path=WheelSet}" Name="_wheelSet" />
                </StackPanel>
                <StackPanel Orientation="Horizontal" Grid.Row="7" Height="31"
HorizontalAlignment="Right" Name="stackPanel9" VerticalAlignment="Top" Width="219"
Background="#FFFCA000">
                    <TextBlock Text="In Stock:" FontWeight="Bold" FontStyle="Italic"
VerticalAlignment="Center" />
                    <TextBox IsEnabled="False" Height="23" Name="_inStock" Width="40"
Text="{Binding Path=InStock}" VerticalAlignment="Center" />
```

```
                    <Slider Height="23" Name="_inStockSlider" Width="122"
Value="{Binding Path=InStock}" Maximum="100" IsEnabled="False" />
                </StackPanel>
            </Grid>
        </DataTemplate>
    </navigation:Page.Resources>

    <Grid x:Name="LayoutRoot" Height="681">


        <riaControls:DomainDataSource
            AutoLoad="True"
            LoadSize="100"
            d:DesignData="{d:DesignInstance my:BikeDTO,
CreateList=true}"
            LoadedData="bikeDTODomainDataSource_LoadedData"
            Name="bikeDTODomainDataSource"
            QueryName="GetBikeDTOListQuery"
            Height="0"
            Width="0">
            <riaControls:DomainDataSource.DomainContext>
                <my1:BikeContext />
            </riaControls:DomainDataSource.DomainContext>

            <!-- Filter on BikeType -->
            <riaControls:DomainDataSource.FilterDescriptors>
                <riaControls:FilterDescriptor
PropertyPath="BikeType"
                                              Operator="StartsWith"
                                              Value="{Binding
ElementName=_bikeTypeSearch, Path=Text}"/>
            </riaControls:DomainDataSource.FilterDescriptors>

            <!-- Sort by default on BikeName -->
            <riaControls:DomainDataSource.SortDescriptors>
                <riaControls:SortDescriptor PropertyPath="BikeName"
Direction="Ascending"/>
            </riaControls:DomainDataSource.SortDescriptors>

        </riaControls:DomainDataSource>

        <localControls:BusyIndicator BusyContent="Retrieving Bike Information ..."
IsBusy="{Binding ElementName=bikeDTODomainDataSource, Path=IsBusy}">

            <Grid Margin="0,11,0,0">

                <Grid.RowDefinitions>
                    <RowDefinition Height="41"/>
                    <RowDefinition Height="596*"/>
                    <RowDefinition Height="33"/>
                </Grid.RowDefinitions>

                <StackPanel Orientation="Horizontal" Margin="20,6,20,0"
HorizontalAlignment="Center" Background="Black" Width="600">
```

```
                    <TextBlock Text="   " Height="25" Width="20" 
VerticalAlignment="Center"/>
                    <TextBlock Text="Search on BikeType:  " Height="25" 
VerticalAlignment="Center" Width="208" FontWeight="Bold" FontSize="18" 
Foreground="#FF937979" />
                    <TextBox x:Name="_bikeTypeSearch" Width="364" Height="25" 
FontWeight="Bold" FontSize="14" />
                    <TextBlock Text="   " Height="25" Width="20" 
VerticalAlignment="Center"/>
                </StackPanel>

                        <ListBox x:Name="_bikeListBox"
                                 ItemsSource="{Binding 
ElementName=bikeDTODomainDataSource, Path=Data}"
                         ItemTemplate="{StaticResource _bikeItemTemplate}" Width="600" 
HorizontalAlignment="Center"
                         Margin="12,0" Grid.Row="1" />

                <StackPanel Orientation="Horizontal" Grid.Row="2">
                    <sdk:DataPager PageSize="2"
                                   Source="{Binding Data, 
ElementName=bikeDTODomainDataSource}" Margin="20,0,20,6"
                                   IsTotalItemCountFixed="True"
                                   HorizontalAlignment="Center" />
                    <Button x:Name="PrintButton" Width="50" Content="Print" 
Click="PrintButton_Click"/>
                </StackPanel>

            </Grid>

        </localControls:BusyIndicator>

    </Grid>
</navigation:Page>
```

Notice the use of the imageconverter and datatemplate for our listbox, the implementation of the RIA-Services DomainDataSource and how the ListBox of the DomainDataSource is connected with our Bike ListBox (super-posed text).

3.4. The Bike List View in Windows Explorer



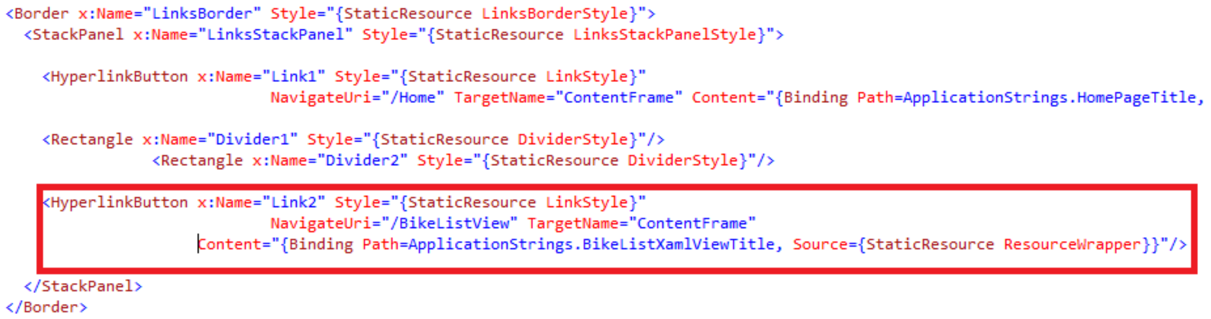
Search on BikeType:

Gambler 10
Model Number:
217966
Frame Type:
Gambler 7005 alloy
Bike Type:
Downhill
Fork Type:
Fox 40 RC2 FIT
Brake Type:
Avid XO carbon disc
Group Set:
Shimano Saint
Wheel Set:
Scott Pro 20mm/ DT Swiss FR
In Stock: 44

Gambler 20
Model Number:
217967
Frame Type:
Gambler 7005 alloy
Bike Type:
Downhill
Fork Type:
Rock Shox Boxxer RC
Brake Type:
Avid Elixir 5 disc
Group Set:
SRAM X7 / X5, Truvative
Wheel Set:
Scott Comp 20mm / Scott DHL
In Stock: 18

3.5. Make the Bike List View Accessible from the main page

We should add next code to the **MainPage.Xaml** (this is the startup page of our SilverLight client Application). This add's a button to view the BikeList page. Also add the title to the **ApplicationStrings** resources of the solution.



```
<Border x:Name="LinksBorder" Style="{StaticResource LinksBorderStyle}">
  <StackPanel x:Name="LinksStackPanel" Style="{StaticResource LinksStackPanelStyle}">

    <HyperlinkButton x:Name="Link1" Style="{StaticResource LinkStyle}"
                         NavigateUri="/Home" TargetName="ContentFrame" Content="{Binding Path=ApplicationStrings.HomePageTitle,

    <Rectangle x:Name="Divider1" Style="{StaticResource DividerStyle}"/>
                <Rectangle x:Name="Divider2" Style="{StaticResource DividerStyle}"/>

    <HyperlinkButton x:Name="Link2" Style="{StaticResource LinkStyle}"
                         NavigateUri="/BikeListView" TargetName="ContentFrame"
                 Content="{Binding Path=ApplicationStrings.BikeListXamlViewTitle, Source={StaticResource ResourceWrapper}}"/>

  </StackPanel>
</Border>
```