Lab 4: .NET 3.5 Silverlight 2 Client for caBIO caGrid Service

caBIG 2009 Annual Meeting Hack-a-thon

Creative Information Technology, Inc.
Joe Richardson, .NET Architect

Overview: Create a basic Silverlight 2 test client for the caBIO caGrid Service. A graphical wizard built into Visual Studio is used to build client-side proxy objects for the caBIO caGrid Service. Because of minor interoperability issues, we will create an adapter service that wraps the caBIO caGRID service and host this adapter service in the same ASP.NET site that hosts the Silverlight app. The adapter service will contain a predefined query to search for genes with their symbol starting with "ABC". In Part 1 of this lab, we create the shell solution containing both an ASP.NET web project and a Silverlight project and will create a WCF Adapter Service for the caBIO caGRID service. In Part 2, we update the Silverlight project's default page to query the adapter service and bind the search results to a grid.
Expected duration: 30 minutes

Part 1: Create ASP.NET host site and caBIO Adapter Service

1) Run Microsoft Visual Web Developer 2008 Express Edition.
   a) Create a new Silverlight project: (File → New Project → Visual C# → Silverlight Application). Name it caBIOSLClient and hit "OK" in this "New Project" window (keeping the option to create a solution directory).
   b) You will be prompted to add a new ASP.NET web project to host the Silverlight application. Click OK to keep the defaults which will create an ASP.NET project named caBIOSLClient.Web in the solution.
2) Right-click on the caBIOSLClient.Web project and select Add Web Reference. Type into the "Address" field: http://cabiogrid40.nci.nih.gov/wsrf/services/cagrid/CaBIO40GridSvc?wsdl and hit "Go". Keep the Namespace = gov.nih.nci.cabiogrid40 and click Add Reference.
3) Create caBIO Adapter Service:
   a) right-click on the caBIOSLClient.Web project
   b) select Add New Item
   c) Choose Silverlight-enabled WCF Service under Silverlight
   d) Leave name Service1.svc
4) Add GeneData DataContract:
   a) Open the Service1.cs code behind file for Service1.svc
   b) Add the GeneData class to the caBIOSLClient.Web namespace:

```
[DataContract]
 public class GeneData
 {
     [DataMember]
     public string Symbol { get; set; }

     [DataMember]
     public string FullName { get; set; }

     [DataMember]
     public string ClusterId { get; set; }

     [DataMember]
     public string Id { get; set; }
 }
```

5) Update Service1 Adapter service to call caBIO service:
   a) Open Service1.cs

b) Add `using caBIOSLClient.Web.gov.nih.nci.cabiogrid40;`
c) Replace DoWork() method with Query:

```
public List<GeneData> Query()
{
      CaBIO40GridSvcService svc = new CaBIO40GridSvcService();
      QueryRequestCqlQuery q = new QueryRequestCqlQuery();
      q.CQLQuery = new CQLQuery();
      q.CQLQuery.Target = new caBIOSLClient.Web.gov.nih.nci.cabiogrid40.Object();
      q.CQLQuery.Target.name = "gov.nih.nci.cabio.domain.Gene";
      q.CQLQuery.Target.Item = new caBIOSLClient.Web.gov.nih.nci.cabiogrid40.Attribute()
      {
            name = "symbol",
            predicate = Predicate.LIKE,
            value = "ABC%"
      };
      CQLQueryResults queryResults = svc.query(q);

      List<GeneData> results = new List<GeneData>();
      foreach (CQLObjectResult item in queryResults.Items)
      {
            results.Add(new GeneData()
            {
                  ClusterId = item.Any.Attributes["clusterId"].Value,
                  Id = item.Any.Attributes["id"].Value,
                  FullName = item.Any.Attributes["fullName"].Value,
                  Symbol = item.Any.Attributes["symbol"].Value
            });
      }

      return results;
}
```

6) Press F6 to build the solution.

Part 2: Create Silverlight caBIO Client

1) Right-click on the caBIOSLClient Silverlight project and select Add Service Reference. Click Discover which should find the Service1.svc service developed in Part 1. Keep the Namespace ServiceReference1.

2) Open Page.xaml and change the dimensions of the root UserControl to 600 x 600.

3) Configure the LayoutRoot Grid control to have 2 rows and add a Button to the first row named QueryButton and a DataGrid to the second row named GeneDataGrid. Setup the row definitions first, and then drag the 2 controls from the Toolbar onto the Page.xaml markup inside the Grid element. The markup should look like this:

```
<UserControl xmlns:data="clr-namespace:System.Windows.Controls;
    assembly=System.Windows.Controls.Data"  x:Class="caBIOSLClient.Page"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Width="600" Height="600">
    <Grid x:Name="LayoutRoot" Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="25"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>
        <Button Name="QueryButton" Content="Click Me!" Grid.Row="0"></Button>
```

```
        <data:DataGrid Name="GeneDataGrid" Grid.Row="1"></data:DataGrid>
    </Grid>
</UserControl>
```

4) In the Page.xaml.cs file (the code behind file for Page.xaml), add an event handler for the QueryButton's click event in the constructor for the Page. Visual Studio will assist in this process by stubbing out an event handler for you with the proper parameters when you type "QueryButton.Click +=" which will bring up a prompt asking whether you want to create a new RoutedEventHandler. Click Tab twice to complete the operation.

5) Implement the event handler stub created in the previous step. Here, we'll create a web service proxy object for the caBIO adapter service and will execute a query. Silverlight requires that all calls to web services be executed asynchronously. So, we will need to create an event handler for the QueryCompleted event of the service proxy (here, the svc object). Visual Studio will again stub out the event handler for you after you type "svc.QueryCompleted +=" and again, click Tab twice to complete the operation:

```
void QueryButton_Click(object sender, RoutedEventArgs e)
{
  ServiceReference1.Service1Client svc = new ServiceReference1.Service1Client();
      svc.QueryCompleted += new
      EventHandler<ServiceReference1.QueryCompletedEventArgs>(svc_QueryCompleted);
      svc.QueryAsync();
}

void svc_QueryCompleted(object sender,ServiceReference1.QueryCompletedEventArgs e)
{
      throw new NotImplementedException();
}
```

6) Implement the QueryCompleted event handler to bind the query search results to the GeneDataGrid. The DataGrid control will auto-generate the columns for you:

```
void svc_QueryCompleted(object sender,ServiceReference1.QueryCompletedEventArgs e)
{
      GeneDataGrid.ItemsSource = e.Result;
}
```

7) You are now ready to run the sample app. Press F5 to start the app in debug mode. Click the QueryButton when the page loads to execute the search. It will take a few seconds to get the data from the caBIO service which should return 97 records. The DataGrid will be populated with the results from the query.