Binding SQL Database to WebScheduler Using ISDataSource

This walkthrough shows you how to bind SQL database to WebScheduler using ISDataSource.

During this walkthrough, you will learn how to do the following:

- Use ISDataSource.
- Use SmartTag to set DataSource.
- Use Data Source Configuration Wizard to set the Database and table.

Prerequisites

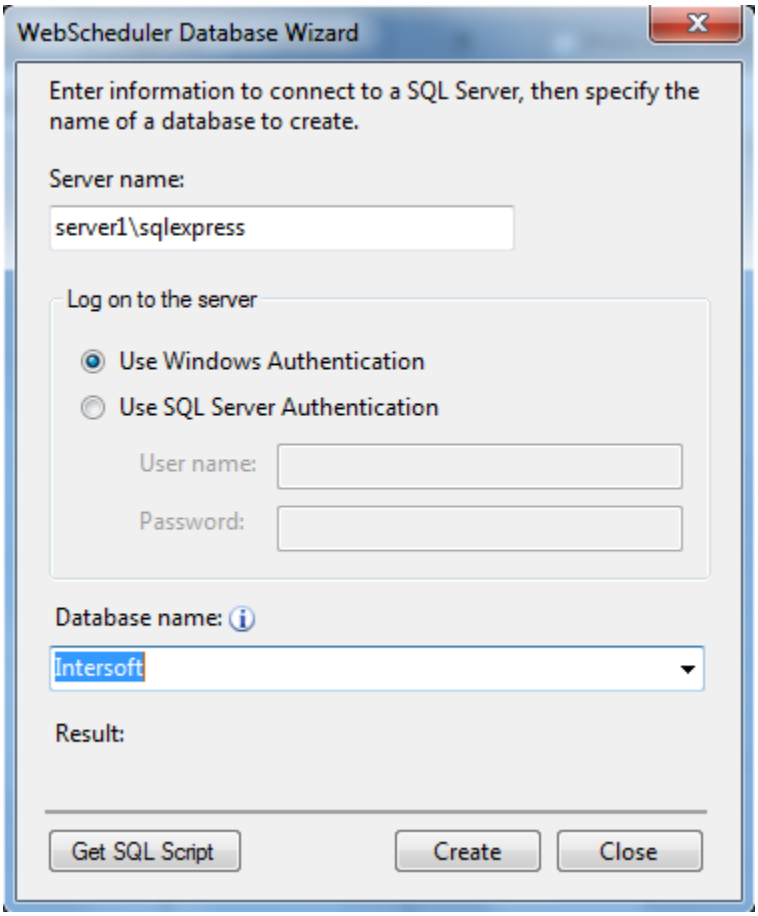
In order to complete this walkthrough, you will need the following:

- Access to the Intersoft SQL database.
- Visual Studio 2008 Application.

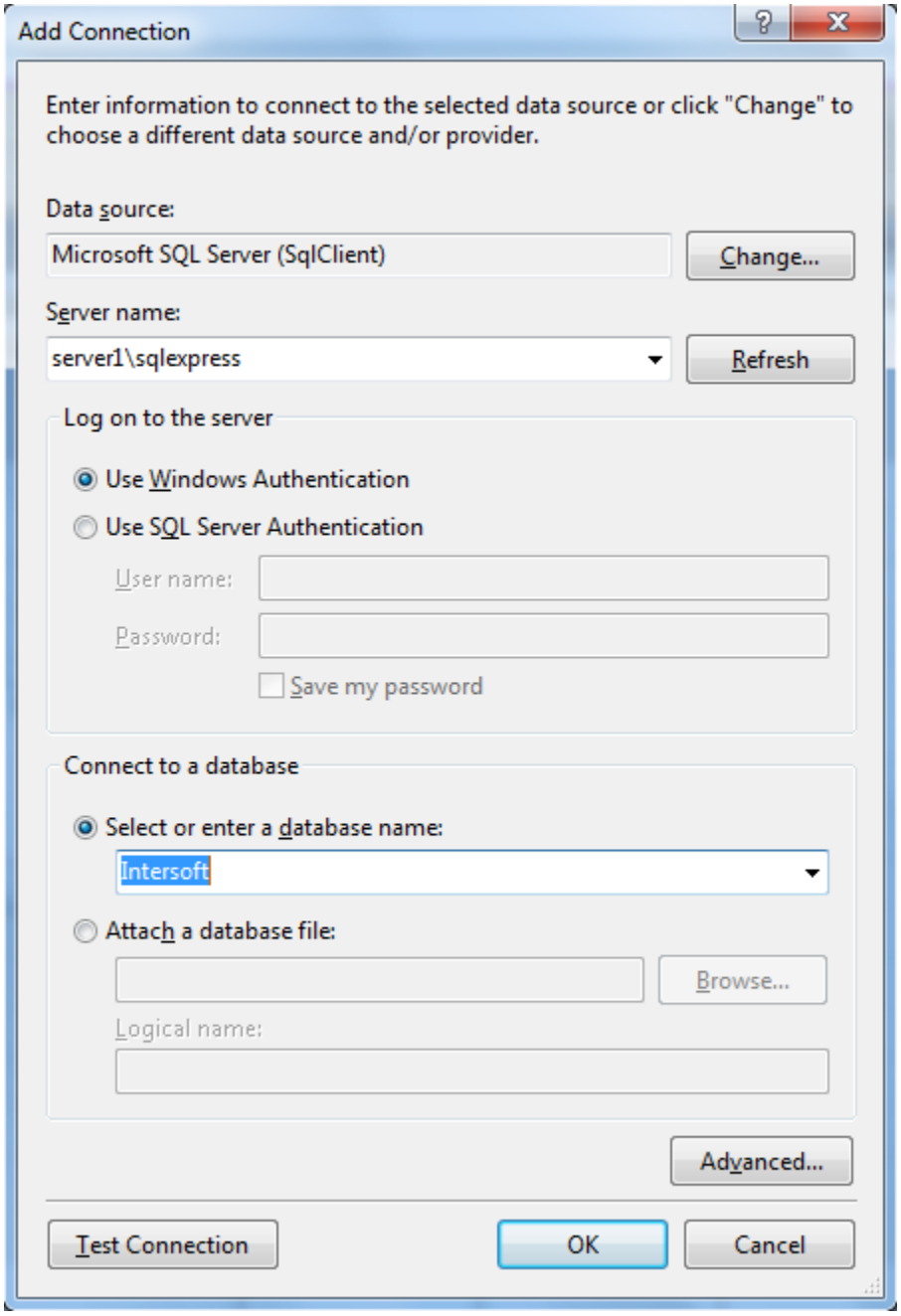
Step-By-Step Instructions

To create new web application and bind ISDataSource to WebScheduler

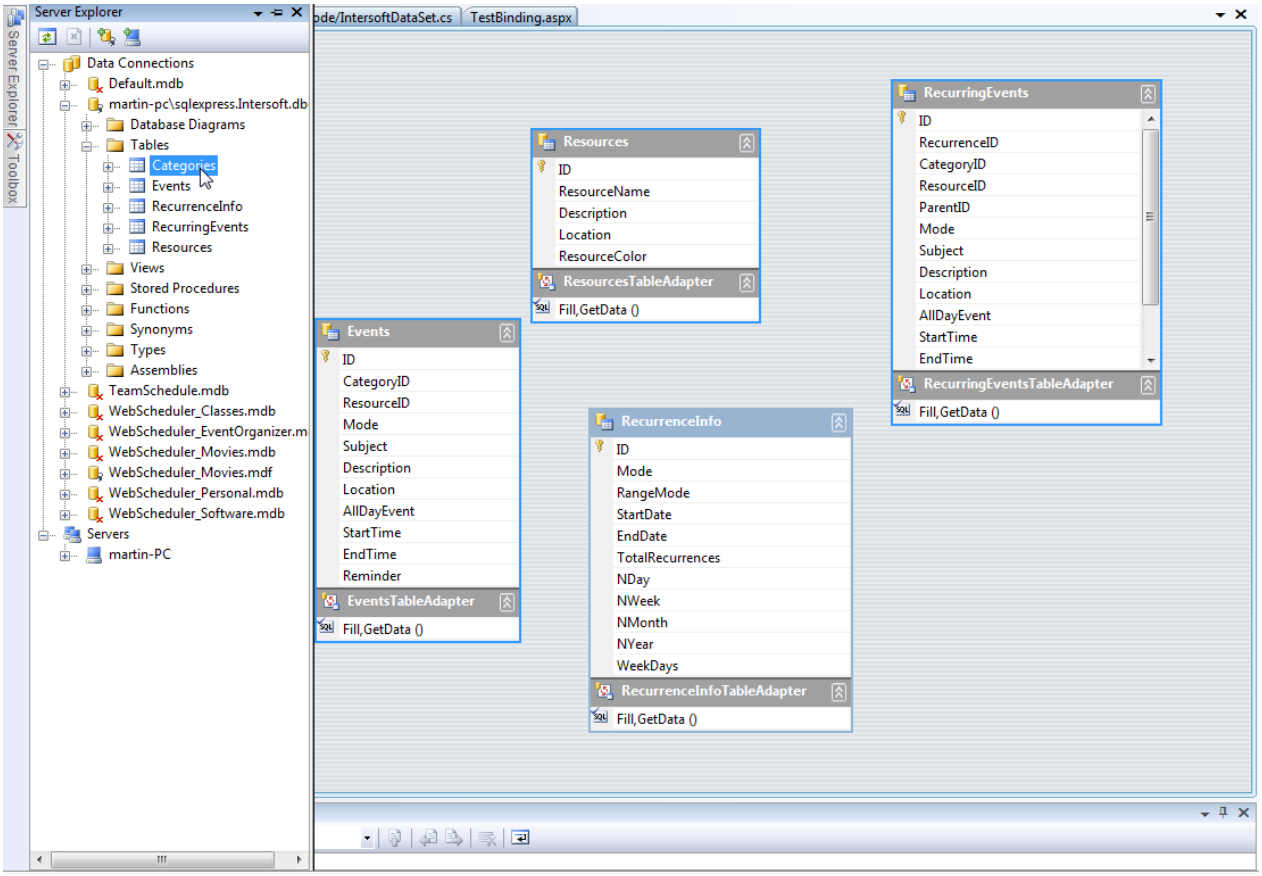
1. Launch **Visual Studio.NET 2008**.
2. Click on File menu, then select **New** and click **Web Site**.
3. Select **ASP.NET Web Site** in the Template box and set **Location** to **HTTP**.
4. Named the Web Site and click **OK**.
5. Right-click on Project's name and select **Add New Item**.
6. Select **Intersoft AppForm** in the My Templates box and named it as **Walkthrough.aspx**.
7. Drag and drop WebScheduler and ISDataSource instances from toolbar to WebForm.
8. Right click on WebScheduler control and choose **WebScheduler Database Wizard**.
9. Provide **Server name** and create **Database name**. In this case, Server name is set to **server1\sqlexpress** and Database name is set to **Intersoft**. Click **Create** button afterwards.



10. Open **Server Explorer** and choose **Connect to Database**. A dialog box will appear.
11. Set Server name to **server1\sqlexpress** and Database name to **Intersoft**. Click **OK** to continue.



12. In the Solution Explorer, right-click on **App_Code** and select **Add New Item**.
13. Add a **Dataset** and named it **Intersoft_DataSet.xsd**.
14. Drag and drop the tables from server explorer to the dataset then save the dataset.



15. In the Solution Explorer, right-click on **App_Code** and select **Add New Item**.
16. Add a new class and named it **Intersoft_DataSet.cs**.
17. Here is the IntersoftDataSet.cs content.

C#

```
using System.Data.SqlClient;

namespace Intersoft_DataSetTableAdapters
{
   public partial class EventsTableAdapter :
global::System.ComponentModel.Component
    {
        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
er")]
```

```
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compon
entModel.DataObjectMethodType.Insert, true)]
     public virtual int Insert(int EventID, string CategoryID,
global::System.Nullable<int> ResourceID, global::System.Nullable<int> Mode,
string Subject,
       string Description, string Location, bool AllDayEvent,
global::System.Nullable<global::System.DateTime> StartTime,
           global::System.Nullable<global::System.DateTime> EndTime,
global::System.Nullable<int> ReminderTimeSpan)
        {
            this.Connection.Open();

            try
            {
                this.Insert(CategoryID, ResourceID, Mode, Subject,
Description, Location, AllDayEvent, StartTime, EndTime, ReminderTimeSpan);

                SqlCommand command = new SqlCommand();
                 command.CommandText = "SELECT @@IDENTITY";
                 command.Connection = this.Connection;

                return int.Parse(command.ExecuteScalar().ToString());
            }
           finally
            {
                 this.Connection.Close();
            }
        }
    }

    public partial class RecurringEventsTableAdapter :
global::System.ComponentModel.Component
    {
    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
er")]

[global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compon
entModel.DataObjectMethodType.Insert, true)]
   public virtual int Insert(int EventID, global::System.Nullable<int>
RecurrenceID, string CategoryID, global::System.Nullable<int> ResourceID,
         global::System.Nullable<int> ParentID, global::System.Nullable<int>
Mode, string Subject, string Description, string Location, bool AllDayEvent,
         global::System.Nullable<global::System.DateTime> StartTime,
global::System.Nullable<global::System.DateTime> EndTime,
         global::System.Nullable<global::System.DateTime> ExceptionDate,
global::System.Nullable<int> ExceptionInfo, global::System.Nullable<int>
ReminderTimeSpan)
    {
     this.Connection.Open();

     try
        {
            this.Insert(RecurrenceID, CategoryID, ResourceID, ParentID, Mode,
Subject, Description, Location, AllDayEvent, StartTime, EndTime,
ExceptionDate, ExceptionInfo, ReminderTimeSpan);
```

```
            SqlCommand command = new SqlCommand();
             command.CommandText = "SELECT @@IDENTITY";
             command.Connection = this.Connection;

            return int.Parse(command.ExecuteScalar().ToString());
        }
       finally
        {
            this.Connection.Close();
        }
    }
}
public partial class RecurrenceInfoTableAdapter :
global::System.ComponentModel.Component
{
   [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
er")]

[global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compon
entModel.DataObjectMethodType.Insert, true)]
  public virtual int Insert(int RecurrenceID, global::System.Nullable<int>
Mode, global::System.Nullable<int> RangeMode,
         global::System.Nullable<global::System.DateTime> StartDate,
global::System.Nullable<global::System.DateTime> EndDate,
         global::System.Nullable<int> TotalRecurrences,
global::System.Nullable<int> NDay, global::System.Nullable<int> NWeek,
         global::System.Nullable<int> NMonth, global::System.Nullable<int>
NYear, string WeekDays)
  {
     this.Connection.Open();

     try
        {
          this.Insert(Mode, RangeMode, StartDate, EndDate, TotalRecurrences,
NDay, NWeek, NMonth, NYear, WeekDays);

             SqlCommand command = new SqlCommand();
              command.CommandText = "SELECT @@IDENTITY";
              command.Connection = this.Connection;

             return int.Parse(command.ExecuteScalar().ToString());
        }
    finally
        {
            this.Connection.Close();
        }
   }
}

public partial class CategoriesTableAdapter :
global::System.ComponentModel.Component
{
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
er")]
[global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compon
entModel.DataObjectMethodType.Insert, true)]
```

```
 public virtual int Insert(int CategoryID, string CategoryName, string
Description, string CategoryColor)
 {
   this.Connection.Open();

     try
     {
       this.Insert(CategoryName, Description, CategoryColor);

        SqlCommand command = new SqlCommand();
         command.CommandText = "SELECT @@IDENTITY";
         command.Connection = this.Connection;

        return int.Parse(command.ExecuteScalar().ToString());
     }
   finally
    {
        this.Connection.Close();
    }
  }
}

 public partial class ResourcesTableAdapter :
global::System.ComponentModel.Component
{
     [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]

[global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapt
er")]

[global::System.ComponentModel.DataObjectMethodAttribute(global::System.Compon
entModel.DataObjectMethodType.Insert, true)]
   public virtual int Insert(int ResourceID, string ResourceName, string
Description, string Location, string ResourceColor)
    {
         this.Connection.Open();

      try
         {
             this.Insert(ResourceName, Description, Location, ResourceColor);

             SqlCommand command = new OleDbCommand();
              command.CommandText = "SELECT @@IDENTITY";
              command.Connection = this.Connection;

             return int.Parse(command.ExecuteScalar().ToString());
         }
        finally
         {
              this.Connection.Close();
         }
```

```
    }
  }
}
```

VB .NET

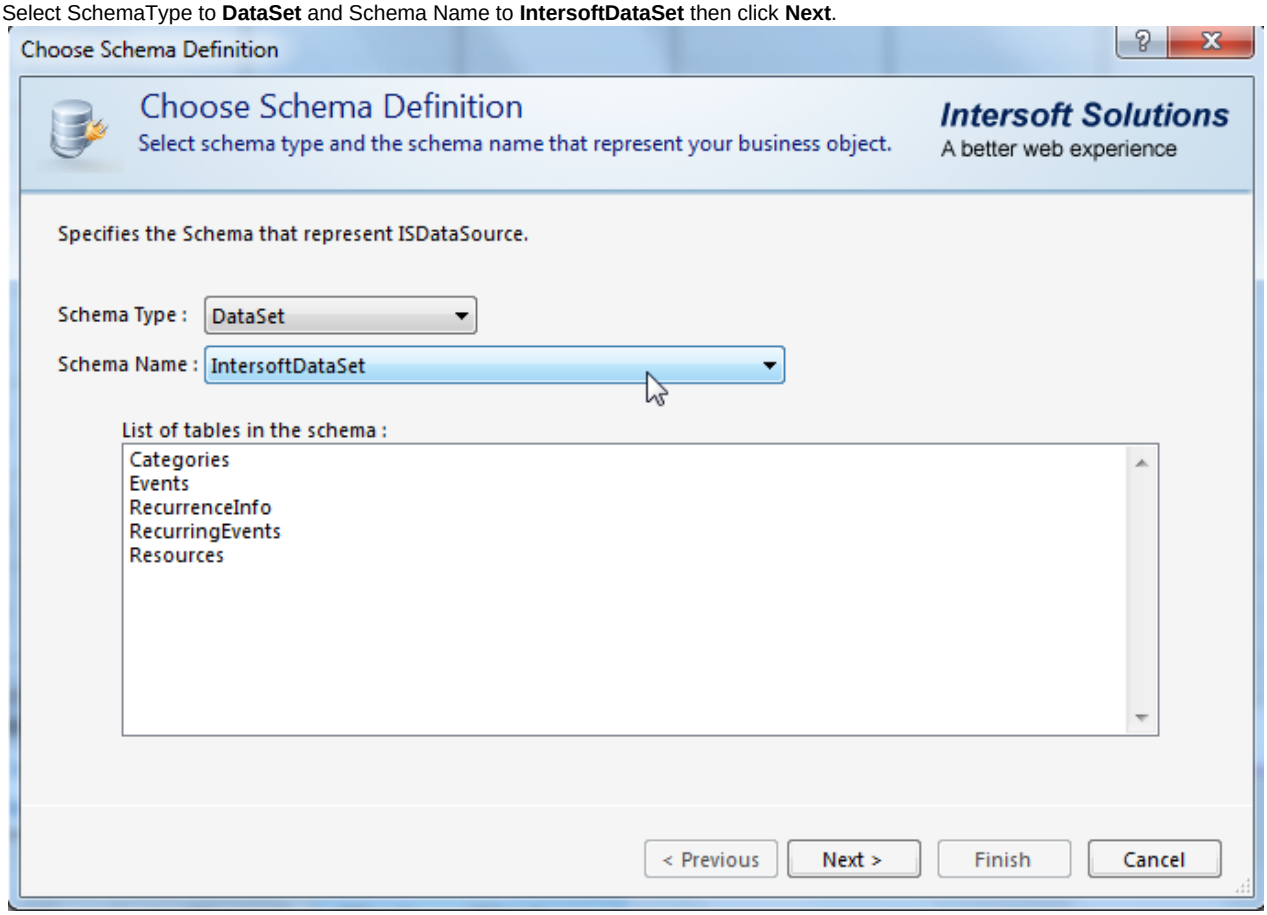
```
Imports System.Data.SqlClient

Namespace Intersoft_DataSetTableAdapters
 Public Partial Class EventsTableAdapter
  Inherits Global.System.ComponentModel.Component
  <System.Diagnostics.DebuggerNonUserCodeAttribute> _
  <System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")>
_

<System.ComponentModel.DataObjectMethodAttribute(Global.System.ComponentModel.
DataObjectMethodType.Insert, True)> _
  Public Overridable Function Insert(EventID As Integer, CategoryID As String,
ResourceID As Global.System.Nullable(Of Integer), Mode As
Global.System.Nullable(Of Integer), Subject As String, Description As String,
_
   Location As String, AllDayEvent As Boolean, StartTime As
Global.System.Nullable(Of Global.System.DateTime), EndTime As
Global.System.Nullable(Of Global.System.DateTime), ReminderTimeSpan As
Global.System.Nullable(Of Integer)) As Integer
   Me.Connection.Open()
   Try
    Me.Insert(CategoryID, ResourceID, Mode, Subject, Description, Location, _
     AllDayEvent, StartTime, EndTime, ReminderTimeSpan)
    Dim command As New SqlCommand()
    command.CommandText = "SELECT @@IDENTITY"
    command.Connection = Me.Connection
    Return Integer.Parse(command.ExecuteScalar().ToString())
   Finally
    Me.Connection.Close()
   End Try
  End Function
 End Class
 Public Partial Class RecurringEventsTableAdapter
  Inherits Global.System.ComponentModel.Component
  <System.Diagnostics.DebuggerNonUserCodeAttribute> _
  <System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")>
_

<System.ComponentModel.DataObjectMethodAttribute(Global.System.ComponentModel.
DataObjectMethodType.Insert, True)> _
  Public Overridable Function Insert(EventID As Integer, RecurrenceID As
Global.System.Nullable(Of Integer), CategoryID As String, ResourceID As
Global.System.Nullable(Of Integer), ParentID As Global.System.Nullable(Of
Integer), Mode As Global.System.Nullable(Of Integer), _
   Subject As String, Description As String, Location As String, AllDayEvent
As Boolean, StartTime As Global.System.Nullable(Of Global.System.DateTime),
EndTime As Global.System.Nullable(Of Global.System.DateTime), _
   ExceptionDate As Global.System.Nullable(Of Global.System.DateTime),
ExceptionInfo As Global.System.Nullable(Of Integer), ReminderTimeSpan As
Global.System.Nullable(Of Integer)) As Integer
```

```
   Me.Connection.Open()
   Try
    Me.Insert(RecurrenceID, CategoryID, ResourceID, ParentID, Mode, Subject, _
     Description, Location, AllDayEvent, StartTime, EndTime, ExceptionDate, _
     ExceptionInfo, ReminderTimeSpan)
    Dim command As New SqlCommand()
    command.CommandText = "SELECT @@IDENTITY"
    command.Connection = Me.Connection
    Return Integer.Parse(command.ExecuteScalar().ToString())
   Finally
    Me.Connection.Close()
   End Try
  End Function
 End Class
 Public Partial Class RecurrenceInfoTableAdapter
  Inherits Global.System.ComponentModel.Component
  <System.Diagnostics.DebuggerNonUserCodeAttribute> _
  <System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")>
_

<System.ComponentModel.DataObjectMethodAttribute(Global.System.ComponentModel.
DataObjectMethodType.Insert, True)> _
  Public Overridable Function Insert(RecurrenceID As Integer, Mode As
Global.System.Nullable(Of Integer), RangeMode As Global.System.Nullable(Of
Integer), StartDate As Global.System.Nullable(Of Global.System.DateTime),
EndDate As Global.System.Nullable(Of Global.System.DateTime), TotalRecurrences
As Global.System.Nullable(Of Integer), _
   NDay As Global.System.Nullable(Of Integer), NWeek As
Global.System.Nullable(Of Integer), NMonth As Global.System.Nullable(Of
Integer), NYear As Global.System.Nullable(Of Integer), WeekDays As String) As
Integer
   Me.Connection.Open()
   Try
    Me.Insert(Mode, RangeMode, StartDate, EndDate, TotalRecurrences, NDay, _
     NWeek, NMonth, NYear, WeekDays)
    Dim command As New SqlCommand()
    command.CommandText = "SELECT @@IDENTITY"
    command.Connection = Me.Connection
    Return Integer.Parse(command.ExecuteScalar().ToString())
   Finally
    Me.Connection.Close()
   End Try
  End Function
 End Class
 Public Partial Class CategoriesTableAdapter
  Inherits Global.System.ComponentModel.Component
  <System.Diagnostics.DebuggerNonUserCodeAttribute> _
  <System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")>
_

<System.ComponentModel.DataObjectMethodAttribute(Global.System.ComponentModel.
DataObjectMethodType.Insert, True)> _
  Public Overridable Function Insert(CategoryID As Integer, CategoryName As
String, Description As String, CategoryColor As String) As Integer
   Me.Connection.Open()
   Try
    Me.Insert(CategoryName, Description, CategoryColor)
    Dim command As New SqlCommand()
    command.CommandText = "SELECT @@IDENTITY"
```

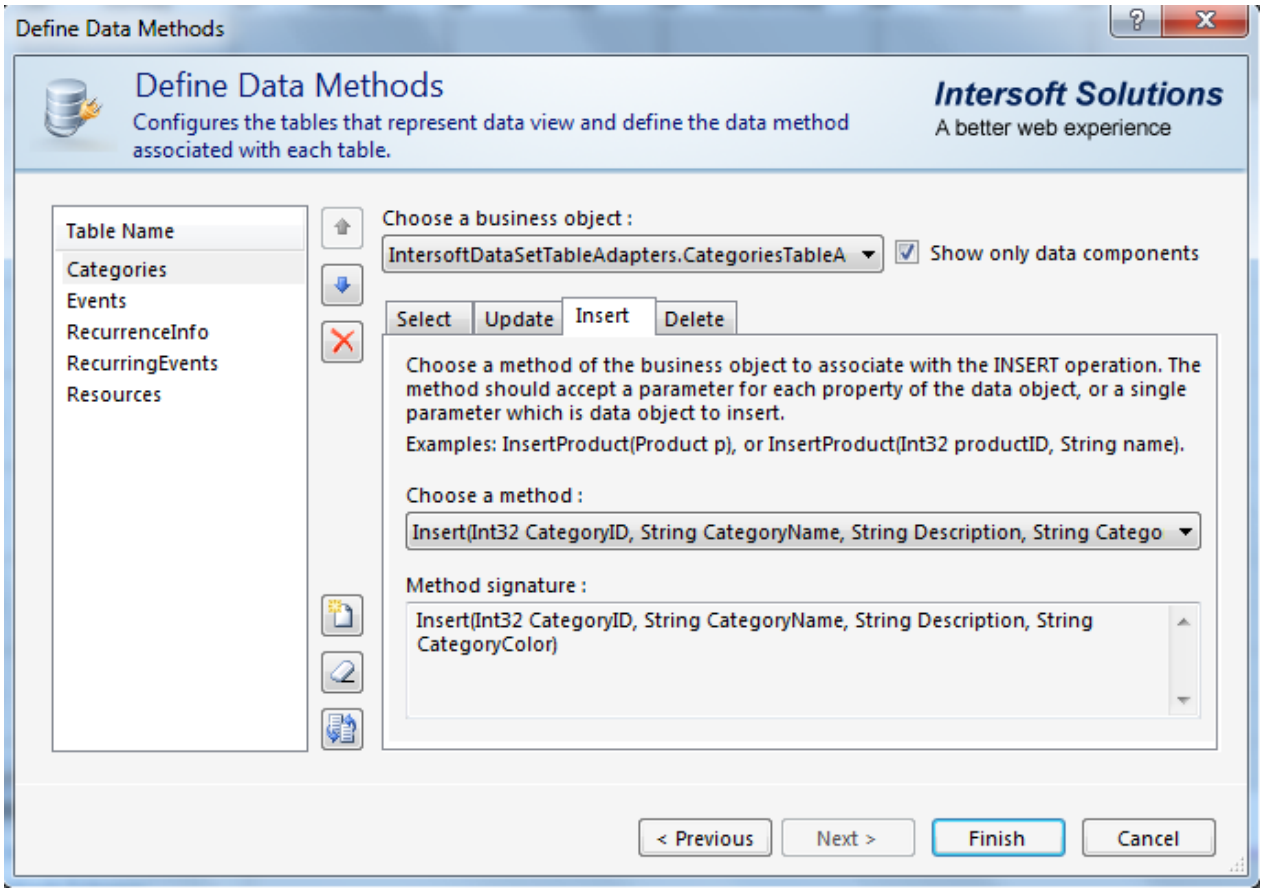
```
    command.Connection = Me.Connection
    Return Integer.Parse(command.ExecuteScalar().ToString())
   Finally
    Me.Connection.Close()
   End Try
  End Function
 End Class
 Public Partial Class ResourcesTableAdapter
  Inherits Global.System.ComponentModel.Component
  <System.Diagnostics.DebuggerNonUserCodeAttribute> _
  <System.ComponentModel.Design.HelpKeywordAttribute("vs.data.TableAdapter")>
_

<System.ComponentModel.DataObjectMethodAttribute(Global.System.ComponentModel.
DataObjectMethodType.Insert, True)> _
  Public Overridable Function Insert(ResourceID As Integer, ResourceName As
String, Description As String, Location As String, ResourceColor As String) As
Integer
   Me.Connection.Open()
   Try
    Me.Insert(ResourceName, Description, Location, ResourceColor)
    Dim command As SqlCommand = New OleDbCommand()
    command.CommandText = "SELECT @@IDENTITY"
    command.Connection = Me.Connection
    Return Integer.Parse(command.ExecuteScalar().ToString())
   Finally
    Me.Connection.Close()
   End Try
  End Function
 End Class
End Namespace
'=========================================================
'Service provided by Telerik (www.telerik.com)
'Conversion powered by NRefactory.
```

```
'Twitter: @telerik
'Facebook: facebook.com/telerik
'=======================================================
```

18. Build the solution.
19. Click the **SmartTag** and select **Configure Data Source** on the upper right of the **ISDataSource** control.
20. Select SchemaType to **DataSet** and Schema Name to **IntersoftDataSet** then click **Next**.



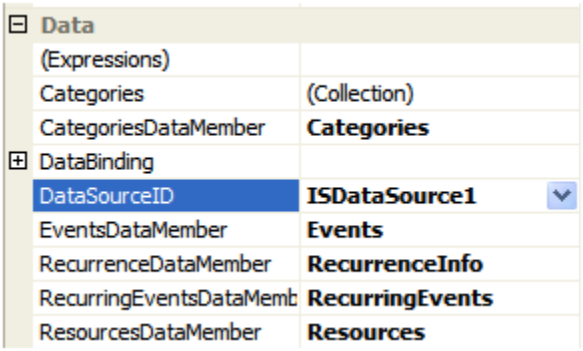
21. Make sure the Select, Insert, Update and Delete tab for each table is not empty then click **Finish**.



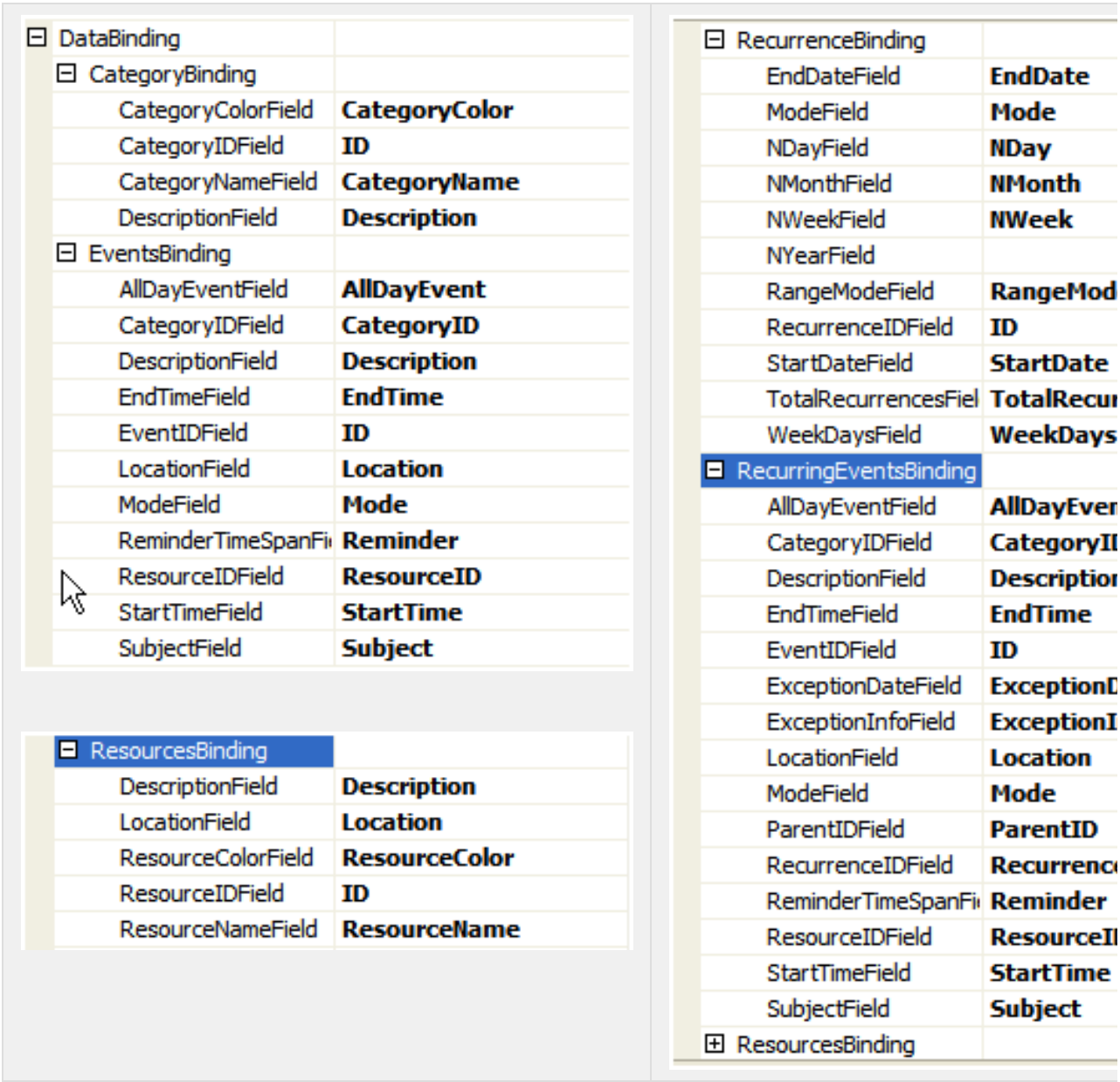
22. Select WebScheduler instance and press F4 to show the WebScheduler's properties.
23. Set the following properties:

Property	Value
DataSourceID	ISDataSource1
CategoriesDataMember	Categories
RecurrenceDataMember	RecurrenceInfo

EventsDataMember	Events
RecurringEventsDataMember	RecurringEvents
ResourcesDataMember	Resources



24. Go to DataBinding and specify the field of CategoryBinding, EventsBinding, RecurrenceBinding, RecurrenceEventsBinding, and ResourcesBinding based on the database field.



25. WebScheduler binding is done. Now you are able to run the project.
    The database is empty so the current WebScheduler will not show any event or resources.

Related Topics

- Use SingleLoadOnDemand
- Use Disabled Time in Month View
- Use Disabled Time
- Specify Holiday(s) in WebScheduler
- Show Week Number in WebScheduler

92 related results