



SQL Anywhere 12 Integration with Visual Studio 2008

A WHITEPAPER FROM SYBASE IANYWHERE

Contents:

Introduction.....	1
Requirements	1
SQL Anywhere .NET Integration Tools	2
SQL Anywhere Data Providers.....	3
Server Explorer Plug-in.....	4
Visual Studio Data Sources.....	7
Entity Data Models.....	13
Summary and Additional Resources	15

Revision History

Version 1.1 - July 2010

Introduction

SQL Anywhere 12 contains a number of integration features with Microsoft Visual Studio 2008 and 2010. These features are designed to make it easier to work with a SQL Anywhere database while developing an application using Visual Studio. This whitepaper outlines the integration features that are present for Visual Studio 2008, and contains tutorials demonstrating how the integration features can be used to ease application development while working with a database.

Requirements

- SQL Anywhere 12 or higher – obtain a free Developer Edition of SQL Anywhere from this location: <http://www.sybase.com/detail?id=1016644>
- Visual Studio 2008
 - Note that Visual Studio 2008 Service Pack 1 and Microsoft .NET Framework 3.5 with Service Pack 1 or later is required for the 'Entity Data Models' section

SQL Anywhere .NET Integration Tools

The SQL Anywhere setup program automatically installs the .NET integration components to your Visual Studio 2008 installation. However, if you install Visual Studio 2008 after installing SQL Anywhere, you must install the SQL Anywhere integration tools:

- Ensure Visual Studio is not running.
- Open a Command Prompt and navigate to the following directory:

```
C:\Program Files\SQL Anywhere 12\Assembly\v2
```

- Run the following command:

```
SetupVSPackage.exe /i
```

If you want to un-install the integration tools:

- Run the following command at the same directory:

```
SetupVSPackage.exe /u
```

SQL Anywhere Data Providers

SQL Anywhere supports the Microsoft .NET Framework through the following namespaces.

iAnywhere.Data.SQLAnywhere The ADO.NET object model is an all-purpose data access model. ADO.NET components were designed to factor data access from data manipulation. There are two central components of ADO.NET that do this: the DataSet, and the .NET Framework data provider, which is a set of components including the Connection, Command, DataReader, and DataAdapter objects. SQL Anywhere includes a .NET Framework Data Provider that communicates directly with a SQL Anywhere database server without adding the overhead of OLE DB or ODBC. The SQL Anywhere .NET Data Provider is represented in the .NET namespace as iAnywhere.Data.SQLAnywhere.

The Microsoft .NET Compact Framework is the smart device development framework for Microsoft .NET. The SQL Anywhere .NET Compact Framework Data Provider supports devices running Windows Mobile.

System.Data.OleDb This namespace supports OLE DB data sources. This namespace is an intrinsic part of the Microsoft .NET Framework. You can use System.Data.OleDb together with the SQL Anywhere OLE DB provider, SAOLEDB, to access SQL Anywhere databases.

System.Data.Odbc This namespace supports ODBC data sources. This namespace is an intrinsic part of the Microsoft .NET Framework. You can use System.Data.Odbc together with the SQL Anywhere ODBC driver to access SQL Anywhere databases.

On Windows Mobile, only the SQL Anywhere .NET Data Provider is supported.

There are some key benefits to using the SQL Anywhere .NET Data Provider:

In the .NET environment, the SQL Anywhere .NET Data Provider provides native access to a SQL Anywhere database. Unlike the other supported providers, it communicates directly with a SQL Anywhere server and does not require bridge technology.

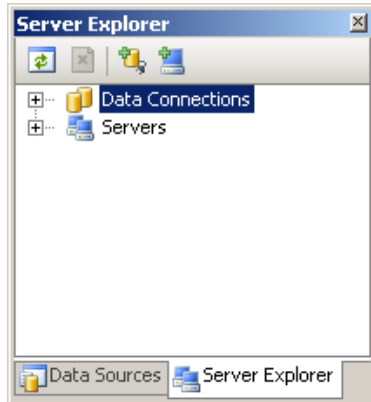
As a result, the SQL Anywhere .NET Data Provider is faster than the OLE DB and ODBC Data Providers. It is the recommended data provider for accessing SQL Anywhere databases.

Server Explorer Plug-in

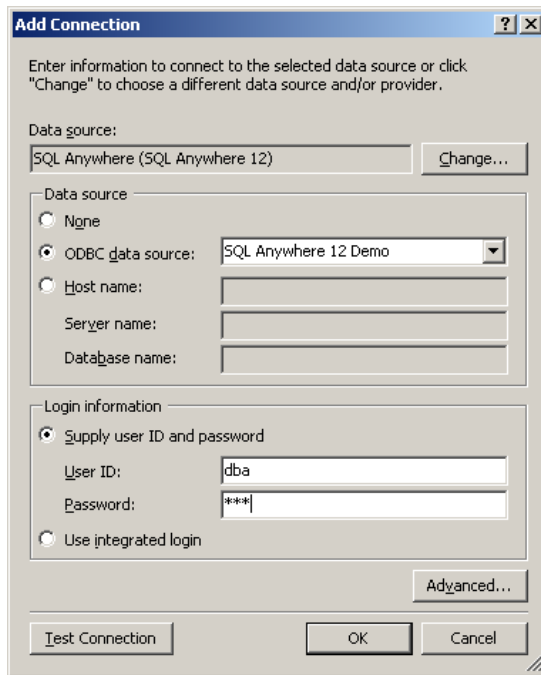
The Visual Studio Server Explorer can be used to display information about databases, such as their schema and the data they contain.

To Open the Server Explorer

1. In Visual Studio, choose **View > Server Explorer**. The Server Explorer appears.



2. Right-click **Data Connections** and choose **Add Connection**. The **Add Connection** dialog appears.
3. If the **Data Source** is not set to **SQL Anywhere (SQL Anywhere 12)**, click **Change** and select **SQL Anywhere** from the list.
4. Select **SQL Anywhere 12 Demo** in the **ODBC Data Source Name** field.
5. In the **User ID** field, type **DBA**, and in the **Password** field, type **sql**.



6. Click **Test Connection** to test the supplied parameters.

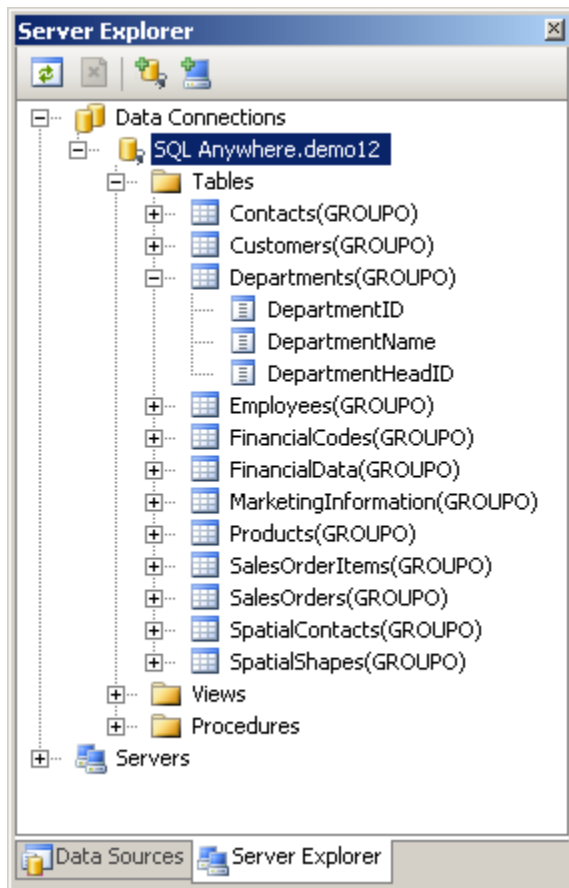
A window appears and indicates whether the connection is successful or if there are problems.

7. Click **OK** to add the connection.

The Server Explorer now displays the new connection—**SQL Anywhere.demo12**.

8. Expand the connection and the **Tables** entry below it.

The Server Explorer shows you all the tables that are in your database. To view the schema for one of the tables, click + beside its name. For example, expand the Departments table to look at its schema:

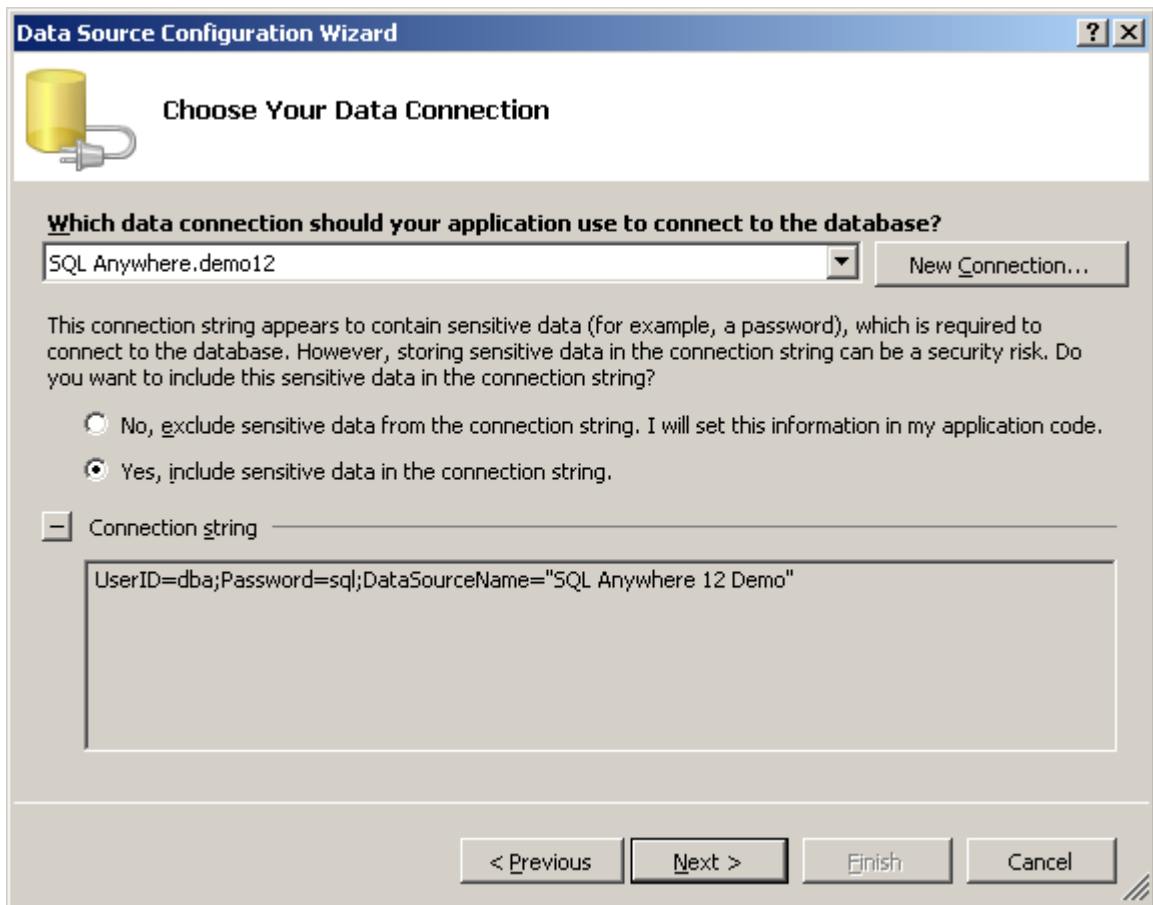


The Departments table contains three columns: DepartmentID, DepartmentName, and DepartmentHeadID.

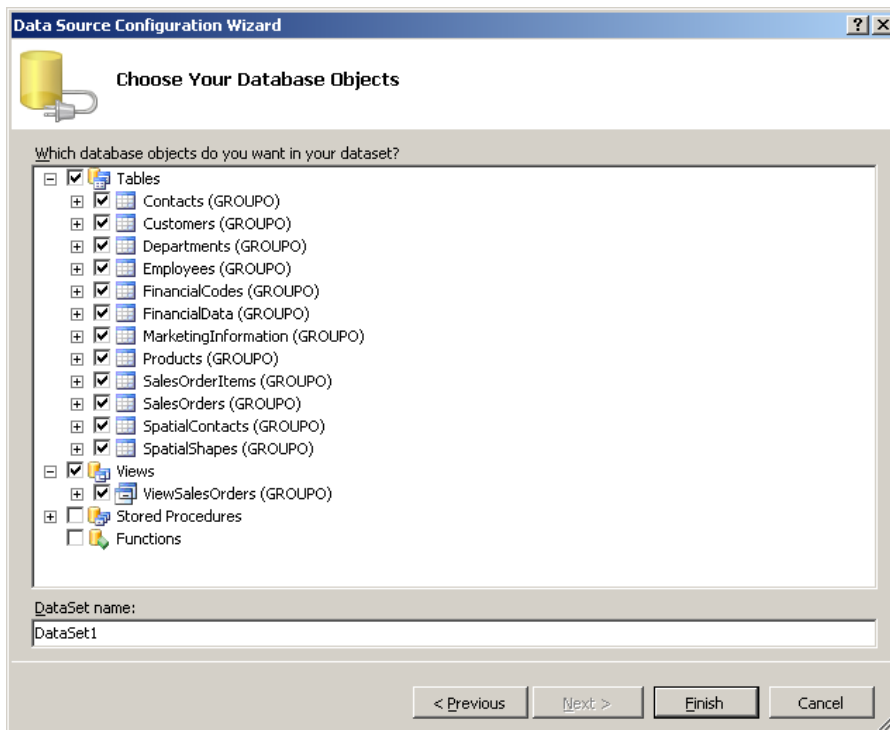
Visual Studio Data Sources

Visual Studio can maintain a list of data sources for your application.

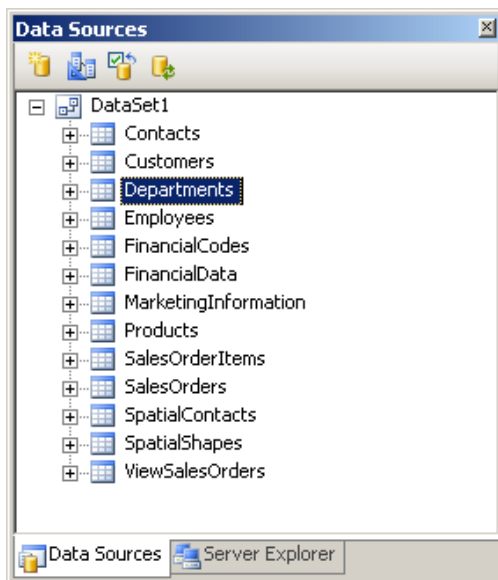
1. Create a new project:
 - a. Choose **File > New Project**.
 - b. Under **Visual C#** or **Visual Basic**, click **Windows Forms Application**.
 - c. Enter the name of your application and click OK.
2. To open the list of data sources or connect to a new data source, choose **Data > Show Data Sources**.
3. Click **Add New Data Source**.
The **Data Source Configuration Wizard** appears.
4. Select the **Database** option, and then click **Next**.
5. Select the **Dataset** option, and then click **Next**.
6. The SQL Anywhere data connection you created in the Server Explorer is available. The wizard informs you that the connection string contains sensitive data. Click + beside **Connection String**. The connection string includes a user ID and password. Since these are the default settings for the database, you can store them in the connection string. Select **Yes, Include Sensitive Data In The Connection String**. Click **Next**.



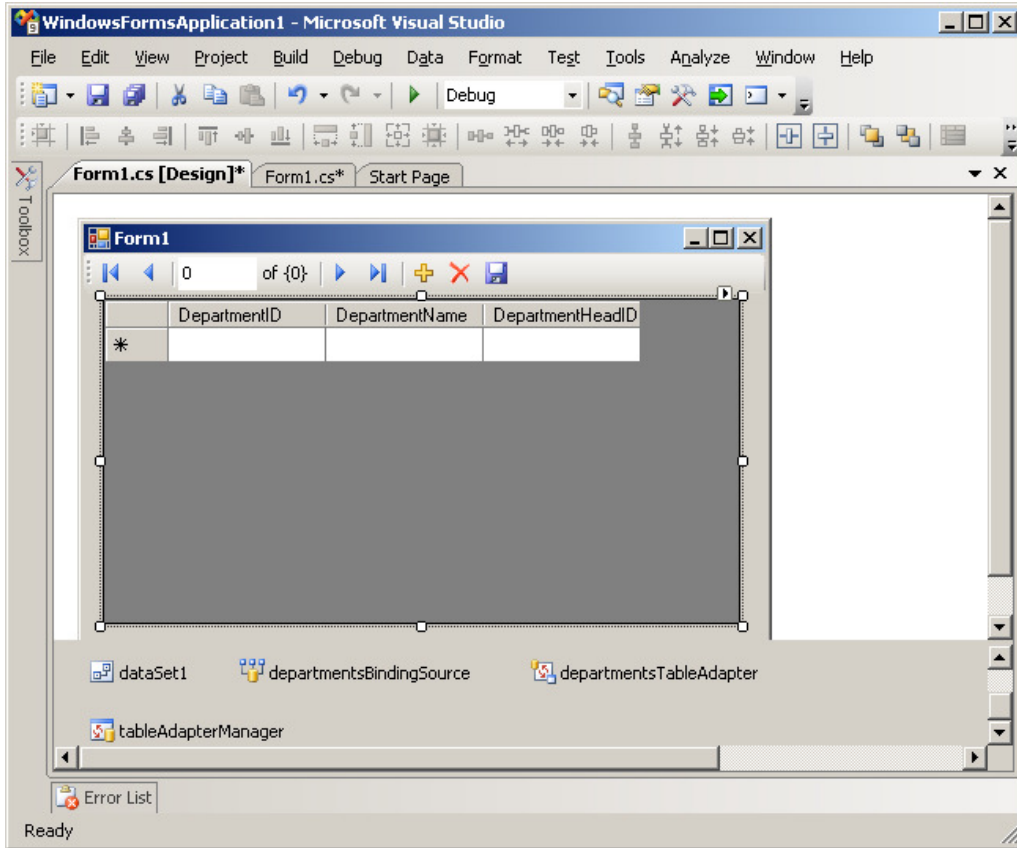
7. Use the default name **ConnectionString**. Click **Next**.
8. Include all of the tables and views in the dataset.



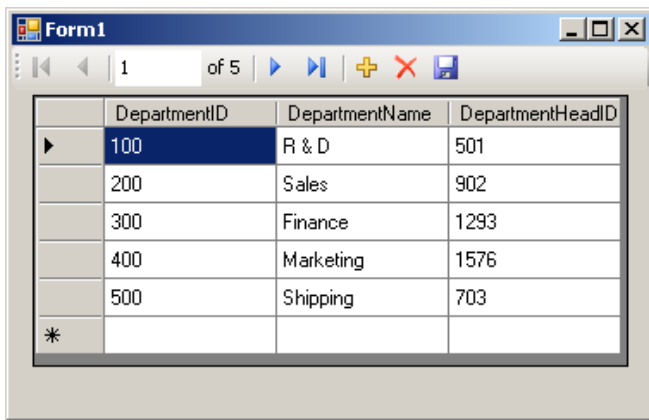
9. Use the default name **DataSet1**. Click **Finish** to close the wizard and create the new data source. **DataSet1** appears in the list of **Data Sources**:



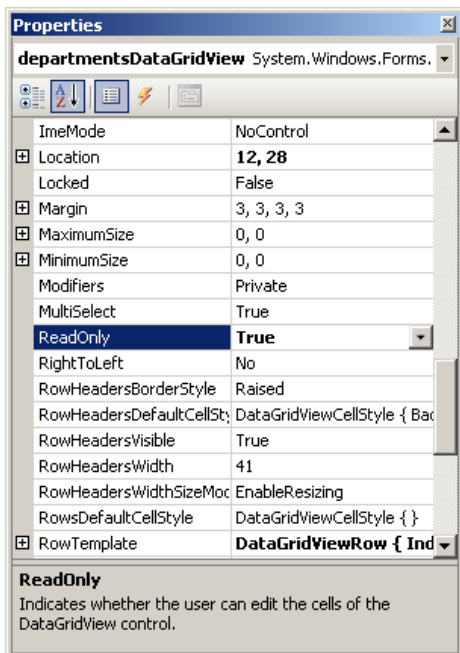
Elements from the DataSet can be dragged and dropped onto your form. For example, drag the Departments table onto your form. Visual Studio automatically creates the necessary bindings and table adapters, and supplies you with a graphical control that you can use to interact with the Departments table.



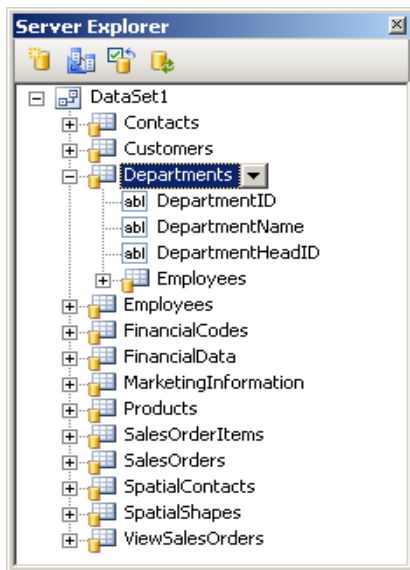
10. Compile and run the application by choosing **Debug > Start Debugging**.
The table is filled with data from the database.



11. Add a new row by typing in the row with the asterisk beside it, or by clicking the + icon in the toolbar. By default, all of the data can be edited and saved to the database.
12. To change the behavior of this control, stop the application, and view the properties in Visual Studio for the data grid. To disallow a user from editing the content, set the **ReadOnly** property to **True**.



13. The **Data Sources** tab also allows you to view the contents of your DataSet. For example, expand the Departments table.



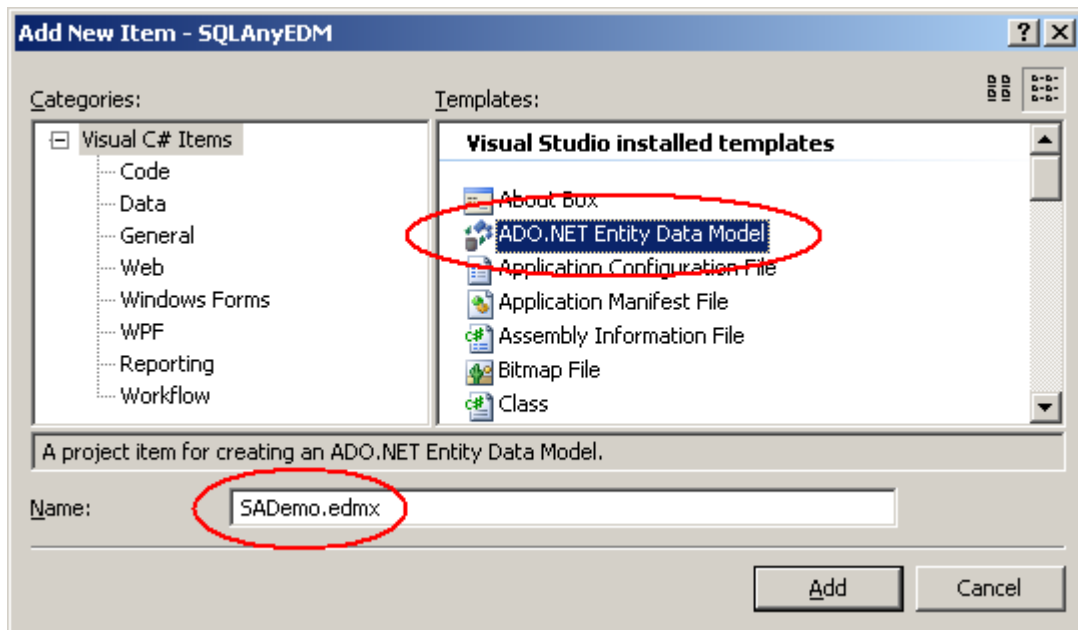
This view shows you the columns that are in that table. It provides another quick way that you can view the table schema from within Visual Studio.

Entity Data Models

A SQL Anywhere database can also be used to create a new entity data model defined in Visual Studio 2008. Follow the steps below to add the SQL Anywhere 12 Demo database as a EDM to your project.

1. Right-click your project, click **Add New Item > ADO.NET Entity Data Model**.
2. In the **Name** field, type **SADemo.edmx**. Click **Add**.

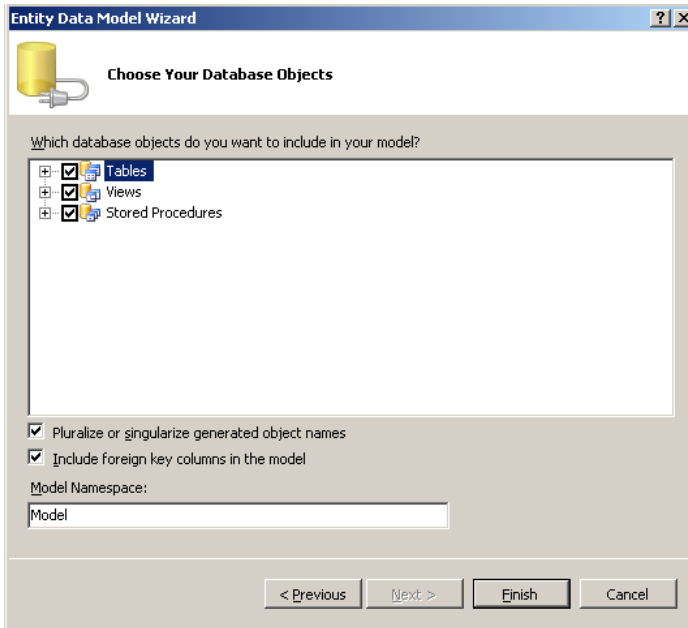
➤ If the ADO.NET Entity Data Model does not show up, verify Visual Studio Service Pack 1 and Microsoft .NET Framework 3.5 with Service Pack 1 are installed properly.



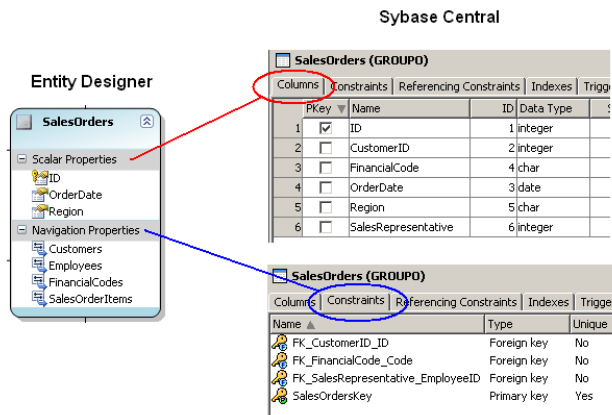
3. Select **Generate from database** and click **Next**.
4. Skip step 4 – 6 if the default connection is SQL Anywhere.demo12. Otherwise, click **New Connection**.
5. In the **Data source** list, click **SQL Anywhere**. Click **Continue**.

➤ If SQL Anywhere does not appear in the Data source list, verify the SQL Anywhere integration components are installed properly.

6. Click **ODBC Data Source name** and select **SQL Anywhere 12 Demo**. Click **OK**.
7. Click **Next**.
8. Include all database objects in the model and Click **Finish**.



- Open **SADemo.edmx** file , a visual representation of the model appears in the Entity Designer. In the diagram below, the generated properties and associations match the database schema.



Summary and Additional Resources

This paper provided an overview of the Server Explorer plug-in and controls available in Visual Studio 2008. It also illustrated how to use the a Dataset object to display information retrieved from a SQL Anywhere database and how to create a Entity Data Model using the SQL Anywhere demo database.

For additional resources such as whitepapers, tutorials, and sample code, please visit the SQL Anywhere .NET Development Center available at this location:

<http://www.sybase.com/developer/library/sql-anywhere-techcorner/microsoft-net>

SYBASE, INC.
WORLDWIDE HEADQUARTERS
ONE SYBASE DRIVE
DUBLIN, CA 94568-7902 USA
Tel: 1 800 8 SYBASE

www.sybase.com/ianywhere