



**Building a Simple ADO.NET Application using
Pervasive.SQL V8 and OLE DB**

*Version 1.0
March 2003*

Pervasive Software Inc.

TABLE OF CONTENTS

BUILDING A SIMPLE ADO.NET APPLICATION USING PERVASIVE.SQL V8 AND OLE DB.....	1
Introduction	3
BUILDING A WEB FORM USING ASP.NET AND PERVASIVE.SQL.....	3
Create the Project	3
Create the DataAdapter	3
Create the Dataset	5
Add the DataGrid to the workarea	5
Permissions and Security	7
BUILDING A WINDOWS FORM USING ASP.NET AND PERVASIVE.SQL.....	10
Create the project	10
Build the Form	10
Add the Code	11
Permissions and Security	12
Run the application	12
Where to Get More Information	13

Introduction

With Pervasive.SQL V8, you can quickly and easily integrate data with Microsoft Visual Studio.NET development tools and .NET platform services. This paper demonstrates step-by-step how to integrate Pervasive.SQL V8 data in a simple data browser application using the Pervasive.SQL V8 OLE DB Provider, and how to run that application as a web form and Windows form under ASP.NET.

This document assumes that the ASP.Net application will be created using the following components:

- Windows XP operating system
- Microsoft Visual Studio.NET
- Microsoft Internet Information Service (IIS)

In order to use a different configuration, you may need to adjust the procedures accordingly. Follow the procedures as they appear in this document to complete the project.

Building a Web Form Using ASP.NET and Pervasive.SQL

Create the Project

1. Start Visual Studio.NET
2. Start a new project
3. Select Visual Basic Project, ASP.NET Web Application
4. Name the project “TestASPGrid” or another meaningful name.
5. Click **OK**.

Visual Studio.NET then builds the application in IIS and creates a project.

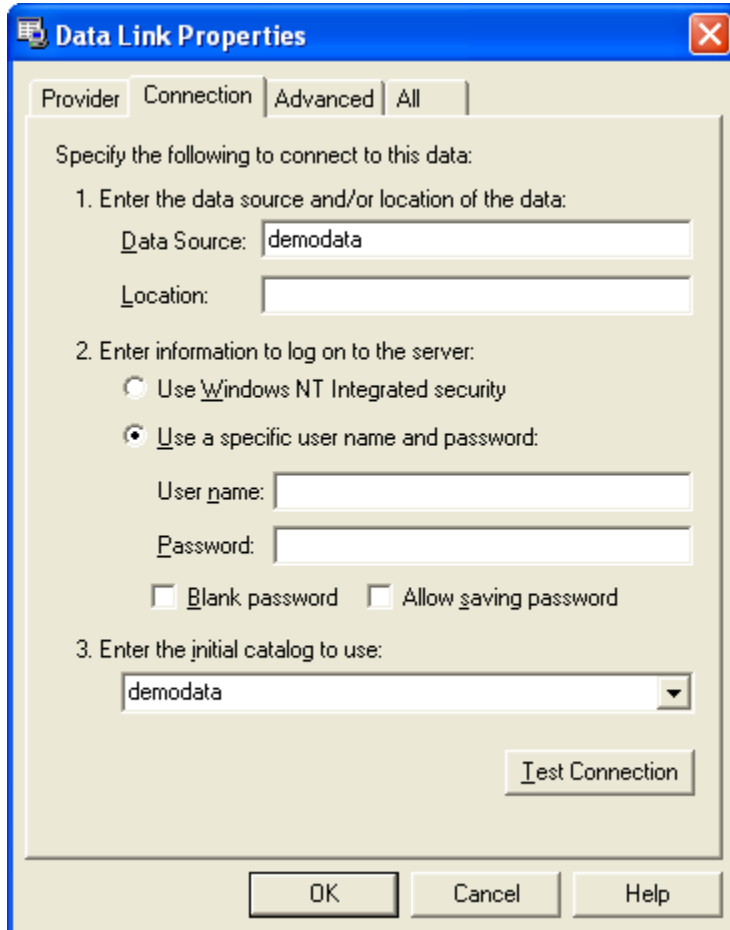
Create the DataAdapter

1. From the Toolbox, Select the “Data” menu item

You may have to Select “View” from the toolbar at the top, then select “Toolbar” from the dropdown menu
2. Select **OleDbDataAdapter**
3. Now click the WebForm work area. The “Data Adapter Configuration Wizard” displays.

4. Click **Next**
5. Select **New Connection...**
6. Select “Pervasive.SQL V8 OLE DB Provider” from the Provider tab
7. Click the **Connection** tab
8. Type “demodata” in both the “Data Source” and the “Initial Catalog” fields.

Note: Pervasive has no use for the initial catalog field, but the wizard erroneously connects if it is not specified.

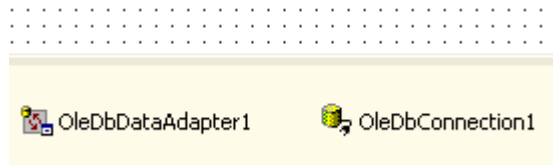


9. Click **OK**.
10. Click **Next**
11. Enter “Select * from Class” in the frame.
12. Click **Next**

You may notice that the wizard generates the Select and Insert commands for you but not the Update or Delete commands. Microsoft only officially supports their own providers within these GUI wizards. Therefore, Microsoft will not correct any deficiencies in these wizards that occur with other providers such as Pervasive. For most issues, you can use a programmatic solution to the issues in the Microsoft wizards.

13. Click **Finish**

Your work area should now resemble the following figure:

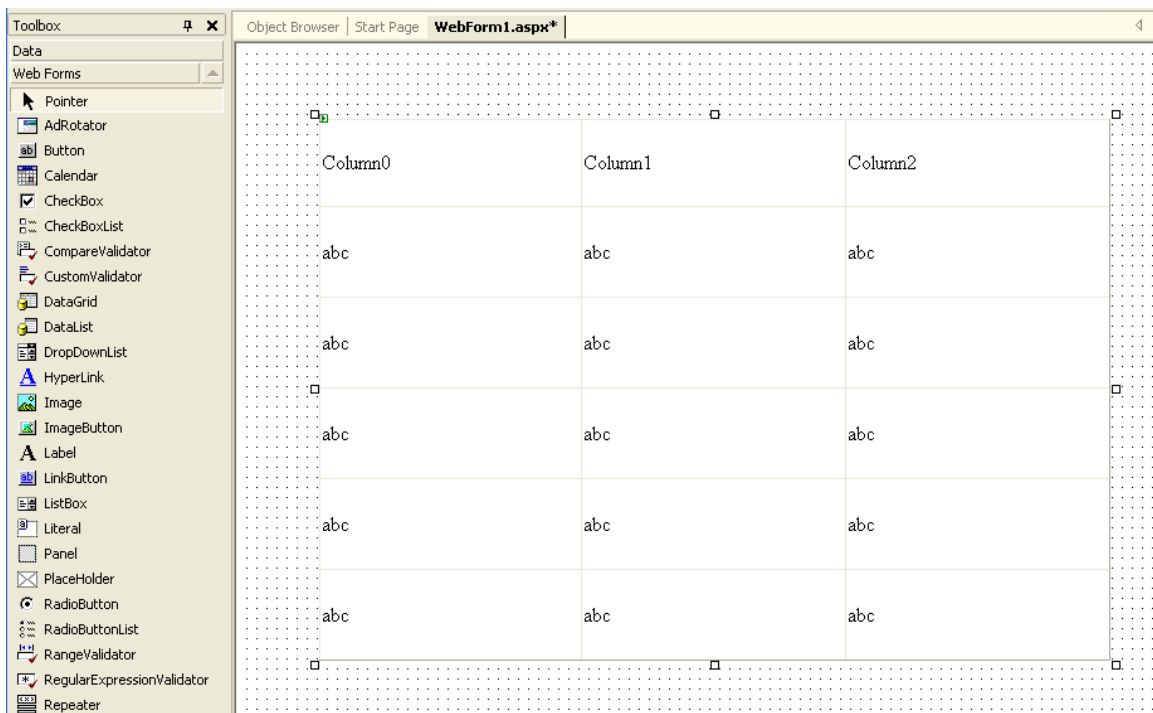


Create the Dataset

1. Right-click the entry for “OleDbDataAdapter1”
2. Select **Generate DataSet**
3. Click **OK**

Add the DataGrid to the workarea

1. In the Toolbox, click the **Web Forms** menu item. This will display all the available web forms
2. Double-click **DataGrid**
3. Resize the grid as necessary as shown in the following figure:

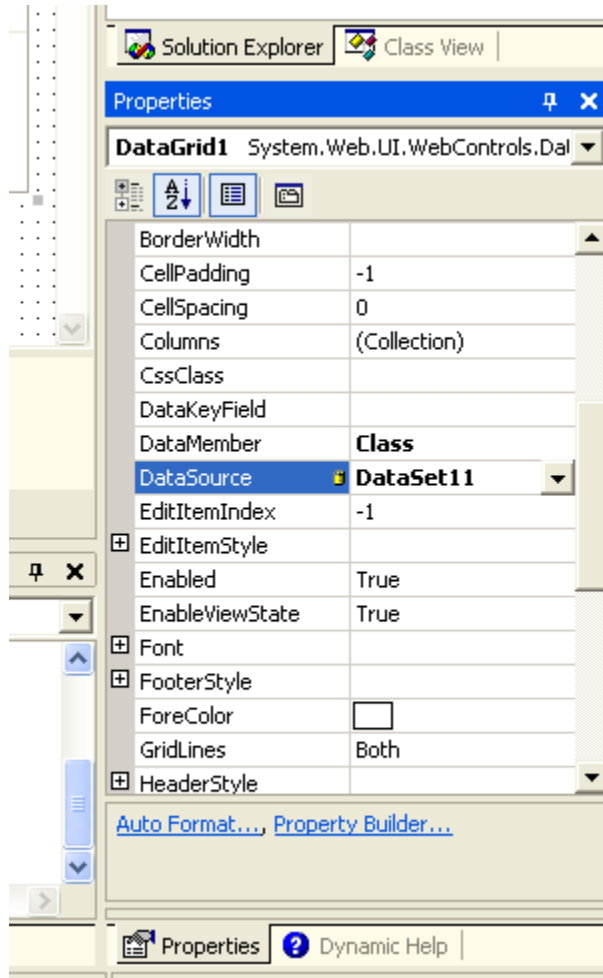


Set the DataGrid properties

1. Select the DataGrid by clicking anywhere within its boundaries in the workarea.
2. In the **Properties** window, the DataGrid properties are now displayed

If the **Properties** window is not visible, select **View ► Properties Window** from the toolbar.

3. Set the **DataSource** property to “DataSet11”.
4. Set the **DataMember** property to “Class”.



Add a button

1. In the Toolbox, click **Web Forms**. This will display all the available web forms
2. Double-click **Button**.
3. Resize the button as necessary.

Connect the Button to the Datagrid

1. Double-click the new button you just created (*Button1*) in the workspace. This moves you to the code portion of the workspace. Your cursor should be positioned in the `Button1_Click` “Sub” method.
2. Add the following code to the method:

```
Me.OleDbDataAdapter1.Fill(Me.DataSet11)  
  
Me.DataGrid1.DataBind()
```
3. You are now ready to build the ASP.NET project. See the next section on Permissions to ensure proper access.

Permissions and Security

This section details any permissions requirements for this project.

Granting permissions for data access

The application needs permission to access PervasiveOLEDB and the necessary data files.

Give Read/Execute Permissions to the Provider

1. Open File Explorer
2. Browse to the folder in which Pervasive.SQL is installed. The default location is `c:\pvsw`.
3. Right-click the `bin` directory
4. Click **Properties**
5. Click the **Security** tab
6. Click **Add**
7. Click **Locations**
8. Select the icon with your computer’s name.
9. Click **OK**
10. In the textbox, enter `ASPNET; IUSR_machineName`

The `IUSR_machineName` field needs to contain the Internet Guest Account for your machine. For example, if the machine name is `TESTSERVER`, the Internet Guest Account for that machine would be `IUSR_TESTSERVER`
11. Click **OK**
12. In the “Group or user names” textbox, select the Internet Guest Account.
13. Grant Read & Execute permissions to the Internet Guest Account

14. From the “Group or user names” textbox, select the user `aspnet_wp`.
15. Grant the `aspnet_wp` account Read & Execute permissions.
16. Click **OK**

Give Modify Permissions to the data files

1. Open File Explorer
2. Browse to the folder in which Pervasive.SQL is installed. The default location is `c:\pvsw`.
3. Right-click the `DEMODATA` directory
4. Click **Properties**
5. Click the **Security** tab
6. Click **Add**
7. Click **Locations**
8. Select the icon with your computer’s name.
9. Click **OK**
10. In the textbox, add `ASPNET; IUSR_machineName`
11. The `IUSR_machineName` needs to be renamed to the Internet Guest Account for your machine. For example, if the machine name is `TESTSERVER`, the name will be `IUSR_TESTSERVER`
12. Click **OK**
13. From the “Group or user names” textbox, select the Internet Guest Account.
14. Grant Modify permission to the Internet Guest Account
15. From the “Group or user names” textbox, select the user `aspnet_wp`.
16. Grant Modify permission to the `aspnet_wp` account.
17. Click **OK**

Give Modify Permissions to the data files

1. Open File Explorer
2. Navigate to the Windows folder. This can be `WINNT` for some operating systems
3. Find the **DBNAMES.CFG** file in the Windows folder
4. Click **Properties**

5. Click the **Security** tab
6. Click **Add**
7. Click **Locations**
8. Select the icon with your computer's name.
9. Click **OK**
10. In the textbox, add `ASPNET; IUSR_machineName`

IUSR_machineName needs to be renamed to the Internet Guest Account for your machine. For example, if the machine name is TESTSERVER, the name will be IUSR_TESTSERVER
11. Click **OK**
12. In the **Group or user names** textbox, select the Internet Guest Account.
13. Grant the Modify permission to the Internet Guest Account.
14. In the "Group or user names" textbox, select the user `aspnet_wp`.
15. Grant the Modify permission to the `aspnet_wp` user.
16. Click **OK**

Restart if Necessary After Permissions Update

Changing permission settings sometimes requires that you restart your machine. If you are prompted to restart your machine, please do so.

Run the application

1. Return to Visual Studio.NET
2. From the toolbar, select **Debug**
3. Click **Start** from the drop down box.

Building a Windows Form Using ASP.NET and Pervasive.SQL

This project is similar to the previous one in that both use a DataAdapter to populate a DataSet and use the DataSet to populate a DataGrid.

However, this project is a Windows application. Also, the data grid will be populated without a wizard, and the data grid is updateable.

Create the project

1. Start Visual Studio.NET
2. Start a new project
3. Select Visual Basic Project, Windows Application
4. Name the project `TestVBGrid` or another meaningful name.
5. Click **OK**.

Visual Studio.NET then builds the application.

Build the Form

1. In the Toolbox, select the **Windows Forms** menu item. The available Windows forms components are displayed.
2. Double-click **DataGrid**
3. Double-click **Button**
4. Resize or reposition the button as necessary.
5. Position your cursor in the newly inserted data grid.
6. In the **Properties** window, change the **Name** property from `datagrid1` to `dgClass`

If the **Properties** window is not visible, select **View ▶ Properties Window** from the toolbar.
7. Position your cursor in the newly inserted button.
8. In the **Properties** window, change the **Name** property from `button1` to `btnUpdate`
9. Change the **Text** property of the button to "Update"

Add the Code

1. Double-click the `btnUpdate` button on the workspace. The **Code** view displays.
2. Modify the code to the following:

```
Public Class Form1

    Inherits System.Windows.Forms.Form

+Windows Form Designer generated code

    Dim m_cn As System.Data.OleDb.OleDbConnection

    Dim m_da As System.Data.OleDb.OleDbDataAdapter

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    m_cn = New System.Data.OleDb.OleDbConnection("Provider=PervasiveOLEDB;Data
Source=demodata")

    m_cn.Open()

    m_da = New System.Data.OleDb.OleDbDataAdapter("Select * from Class", m_cn)

    Dim ds As New System.Data.DataSet()

    Dim cb As New System.Data.OleDb.OleDbCommandBuilder(m_da)

    m_da.Fill(ds, "Class")

    Me.dgClass.DataSource = ds

    Me.dgClass.DataMember = "Class"

End Sub

Private Sub Form1_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing

    m_cn.Close()

End Sub

Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdate.Click

    Dim ds As System.Data.DataSet

    ds = Me.dgClass.DataSource

    m_da.Update(ds, "Class")

End Sub

End Class
```

Permissions and Security

This application runs in the same context as the user who starts the application. Because of this, the application will not have the same security restrictions as would an ASP.Net application.

However, the user who runs the application must have permissions to the Database Name just like any other OLE DB application would require.

Run the application

1. From the toolbar, select **Debug**
2. Click **Start** from the drop down box.
3. You can now insert, update, and delete rows.

You now have a working Pervasive.SQL .NET application!

Where to Get More Information

For more information about Pervasive.SQL and .NET, see the following topics.

- Developer Center <http://www.pervasive.com/developerzone/platforms/net.asp>
- Programmer's Guide http://www.pervasive.com/library/index.asp?_shownode=GDE
- Visual Programmer's Reference
http://www.pervasive.com/library/index.asp?_shownode=VIS

For more information about Pervasive and our solutions, please visit our Web site at www.pervasive.com, or locate one of our sales offices worldwide at www.pervasive.com/company/contact/index.asp.

The logo for Pervasive Software, featuring the word "PERSVASIVE" in a blue, sans-serif font. The letter "P" is stylized with a vertical line extending downwards from its base. A registered trademark symbol (®) is located at the top right of the word.

©2003 Pervasive Software Inc. Pervasive, Pervasive.SQL, Btrieve, and the Pervasive logo are registered trademarks of Pervasive Software Inc. All other product names may be trademarks of their respective companies. All rights reserved worldwide.

Disclaimer: The performance demonstrated in this document is for reference purposes only and does not constitute a performance guarantee even if under similar configurations.