Using a Database as a Data Source

Code Sample: Importing from a Database

[C#][VB.NET]
The sample above generates a spreadsheet from the template using ADO.NET objects as the data source.

The template contains the following data markers:

Data Markers

```
 %%=Employee.Name
%%=Employee.Title
%%=Orders.OrderID
%%=Orders.Customer
%%=Orders.OrderDate
%%=Orders.OrderTotal
```

These data markers will be populated in the ASP.NET code by two database data sources. The data source for the %%=Employee.* data markers is an ADO.NET DataTable, and the data source for the %%=Orders.* data markers is an ADO.NET DataSet.

To set a template data source to a DataSet or DataTable using OleDB, import the System.Data and System.Data.OleDB namespaces to the ASP.NET page. For example:

```
using System.Data;
using System.Data.OleDb;
```

```
Imports System.Data
Imports System.Data.OleDb
```

Setting a Data Source to a DataSet with OleDB

To create a DataSet to use as a data source:

1. Create an OleDbConnection object to open a database connection.
2. Create a SQL query to get data from the database.
3. Create an OleDbDataAdapter that will execute the SQL command at the data source and fill the DataSet.
4. Create a DataSet and use the OleDbDataAdapter to fill it with the data retrieved from the database.

For example, in Databasebind.aspx, the following code creates and fills OrdersDs, the DataSet which will be used as the data source for the set of %%=Orders.* data markers.

```
OleDbConnection Conn = new OleDbConnection();
DataSet OrdersDs = null;
try
{
     Conn.ConnectionString = Application["connstring"].ToString();

     //--- SQL Query for orders
     string OrdersSQL = "SELECT Orders.OrderID, " +
          "Customers.CompanyName As Customer, " +
          "Orders.OrderDate, " +
          "([Order Details].UnitPrice * " +
          "[Order Details].Quantity) As [OrderTotal] " +
          "FROM Orders, [Order Details], Customers " +
          "WHERE Orders.OrderID=[Order Details].OrderID AND " +
          "Orders.CustomerID=Customers.CustomerID AND Orders.EmployeeID=?";
     OleDbCommand CmdOrders = new OleDbCommand(OrdersSQL, Conn);
     CmdOrders.Parameters.Add("@EmployeeID", EmployeeId);
     OleDbDataAdapter AdptSales = new OleDbDataAdapter(CmdOrders);
     OrdersDs = new DataSet();
     AdptSales.Fill(OrdersDs, "Orders");
}
```

```
 Dim Conn As New OleDbConnection()
Dim OrdersDs As DataSet = Nothing
Try
     Conn.ConnectionString = Application("connstring").ToString()

     '--- SQL Query for orders
     Dim OrdersSQL As String = "SELECT Orders.OrderID, " & _
          "Customers.CompanyName As Customer, " & _
          "Orders.OrderDate, " & _
          "([Order Details].UnitPrice * " & _
          "[Order Details].Quantity) As [OrderTotal] " & _
          "FROM Orders, [Order Details], Customers " & _
          "WHERE Orders.OrderID=[Order Details].OrderID AND " & _
          "Orders.CustomerID=Customers.CustomerID AND Orders.EmployeeID=?"
     Dim CmdOrders As New OleDbCommand(OrdersSQL, Conn)
     CmdOrders.Parameters.Add("@EmployeeID", EmployeeId)
     Dim AdptSales As New OleDbDataAdapter(CmdOrders)
     OrdersDs = New DataSet()
     AdptSales.Fill(OrdersDs, "Orders")
End Try
```

To bind a DataSet to a template data marker, call [BindData|ExcelTemplate.BindData] using the following signature:

```
ExcelTemplate.BindData(DataSet aData,
String aDataMarkerName,
DataBindingProperties aDataBindingProperties)
```

This signature takes a DataSet, the name of the data marker to which the data source should bind, and a collection of binding property values. In Databasebind.aspx, this BindData call binds the DataSet OrdersDs to the data markers ***%%=Orders.OrderID***, ***%%=Orders.Customer***, ***%%=Orders.OrderDate***, and ***%%=Orders.OrderTotal*** in the template:

```
xlt.BindData(OrdersDs, "Orders", xlt.CreateDataBindingProperties());
```

```
 xlt.BindData(OrdersDs, "Orders", xlt.CreateDataBindingProperties())
```

Setting a Data Source to a DataTable with OleDB

To create a DataTable to use as a data source:

1. Create an OleDbConnection object to open a database connection.
2. Create a SQL query to get data from the database.
3. Create an OleDbDataAdapter that will execute the SQL command at the data source and fill the DataTable.
4. Create a DataTable and use the OleDbDataAdapter to fill it with the data retrieved from the database.

For example, in Databasebind.aspx, the following code creates and fills EmployeeDt, the DataTable which will be used as the data source for the set of %%=Employee.* data markers.

```
OleDbConnection Conn = new OleDbConnection();
DataTable EmployeeDt = null;
try
{
     Conn.ConnectionString = Application["connstring"].ToString();

     //--- SQL Query for employee information
     string EmployeeSQL = "SELECT FirstName + ' ' + LastName As Name, Title " +
          "FROM Employees WHERE EmployeeID=?";
     OleDbCommand CmdEmployee = new OleDbCommand(EmployeeSQL, Conn);
     CmdEmployee.Parameters.Add("@EmployeeID", EmployeeId);
     OleDbDataAdapter AdptEmployee = new OleDbDataAdapter(CmdEmployee);
     EmployeeDt = new DataTable();
     AdptEmployee.Fill(EmployeeDt);
}
```

```
 Dim Conn As New OleDbConnection()
Dim EmployeeDt As DataTable = Nothing
Try
     Conn.ConnectionString = Application("connstring").ToString()

     '--- SQL Query for employee information
     Dim EmployeeSQL As String = "SELECT FirstName & ' ' & LastName As Name, Title " &
_
          "FROM Employees WHERE EmployeeID=?"
     Dim CmdEmployee As New OleDbCommand(EmployeeSQL, Conn)
     CmdEmployee.Parameters.Add("@EmployeeID", EmployeeId)
     Dim AdptEmployee As New OleDbDataAdapter(CmdEmployee)
     EmployeeDt = New DataTable()
     AdptEmployee.Fill(EmployeeDt)
End Try
```

To bind a DataTable to a template data marker, call [BindData|ExcelTemplate.BindData] using the following signature:

```
BindData(DataTable aData, String aDataMarkerName, DataBindingProperties,
aDataBindingProperties)
```

This signature takes a DataTable, the name of the data marker to which the data source should bind, and a collection of binding property values. In Databasebind.aspx, this BindData call binds the DataTable EmployeeDt to the data markers ***%%=Employee.Name*** and ***%%=Employee.Title*** in the template:

```
xlt.BindData(EmployeeDt,"Employee",xlt.CreateDataBindingProperties());
```

```
 xlt.BindData(EmployeeDt, "Employee", xlt.CreateDataBindingProperties())
```