

Using a Database as a Data Source

Table of Contents

- Code Samples
 - Code Sample: Purchase Order Document
 - Code Sample: Mailing Labels
 - Code Sample: Mail Merge
- Using an ADO.NET DataTable or DataSet as a Data Source
- Using an SqlDataReader, OleDbDataReader or AdomdDataReader as a Data Source

You can use a database table as a data source for a WordWriter template by passing the SetDataSource method a DataTable, a DataSet, an SqlDataReader, OleDbDataReader or an AdomdDataReader. If the specified data source contains more than one row, SetDataSource will use the first row of the first table as the data source. To import multiple rows from data sources [use a repeat block](#).

Import the System.Data namespace for DataTable and DataSet objects. Import System.Data.SqlClient for SQL Server-specific database classes, or System.Data.OleDb for the OLEDB client and connections to other databases such as Microsoft Access .MDB files.

The following samples use SqlConnection classes and connect to the AdventureWorks2000 database. These code samples use ADO.NET objects as data sources:

Code Samples

Code Sample: Purchase Order Document

[C#] | [VB.NET]

Code Sample: Mailing Labels

[C#] | [VB.NET]

Code Sample: Mail Merge

[C#] | [VB.NET]

Using an ADO.NET DataTable or DataSet as a Data Source

The following code example shows how to query the AdventureWorks2000 database and return the data as a DataSet object. The DataSet object is then used as the data source in a WordTemplate document:

C#

```
void GenerateDocument(int employeeID)
{
    ///--- Query the database
    DataTable dt = new DataTable();
    using(SqlConnection conn = new SqlConnection(connString))
    {
        string sql =
            "SELECT FirstName, LastName FROM Employee WHERE EmployeeId = @id";
        SqlCommand cmd =
            new SqlCommand(sql, conn);

        ///--- Use the employee ID as a SQL parameter
        cmd.Parameters.Add("@id", employeeID);
        SqlDataAdapter adpt = new SqlDataAdapter(cmd);
        adpt.Fill(dt);
    }

    ///--- Use a DataTable as the data source
    WordTemplate wt = new WordTemplate();
    wt.Open(templatePath);
    wt.SetDataSource(dt);
    wt.Process();
    wt.Save(Page.Response, "DatabaseOutput.doc", false);
}
```

VB.NET

```
Private Sub GenerateDocument(ByVal employeeID As Integer)
    '--- Query the database
    Dim dt As New DataTable()
    Dim conn As New SqlConnection(connString)
    Dim sql As String = _
        "SELECT FirstName, LastName FROM Employee WHERE EmployeeId = @id"
    Try
        Dim cmd As New SqlCommand(sql, conn)

        '--- Use the employee ID as a SQL parameter
        cmd.Parameters.Add("@id", employeeID)
        Dim adpt As New SqlDataAdapter(cmd)
        adpt.Fill(dt)
    Finally
        If Not conn Is Nothing Then
            conn.Dispose()
        End If
    End Try

    '--- Use a DataTable as the data source
    Dim wt As New WordTemplate()
    wt.Open(templatePath)
    wt.SetDataSource(dt)
    wt.Process()
    wt.Save(Page.Response, "DatabaseOutput.doc", False)
End Sub
```

The first block of code in the `GenerateDocument` method connects to a database using a variable `connString` which represents a SQL Server connection string. Then, a `SqlCommand` is created and a parameter is added to the query. A `SqlDataAdapter` then executes the `SqlCommand` and fills a `DataTable` with the data. The next block of code shows how to use this `DataTable` with `WordTemplate`.

You can also use a `DataSet` as the data source. When using a `DataSet`, `WordTemplate` will use only the first `DataTable` in the `DataSet.Tables` collection.

Using an `SqlDataReader`, `OleDbDataReader` or `AdomdDataReader` as a Data Source

`SqlDataReader`, `OleDbDataReaders` and `AdomdDataReader` are also available for use by `WordTemplate`. Remember that unlike `DataSet` and `DataTable` objects, `DataReader` objects require an open connection to the database while reading data. Remember to close `SqlDataReader`, `OleDbDataReader` or `AdomdDataReader` objects calling `WordTemplate.Process()`.

C#

```
SqlCommand cmd;

//--- CloseConnection will close the associated connection
//--- when the reader is closed
SqlDataReader rdr = cmd.ExecuteReader(CommandBehavior.CloseConnection);

try
{
    WordTemplate wt = new WordTemplate();
    wt.Open(templatePath);
    wt.SetDataSource(rdr);
    wt.Process();
    wt.Save(Page.Response, "DataReader.doc", false);
}
finally
{
    if(rdr!=null)
        rdr.Close();
}
```

VB.NET

```
Dim cmd As SqlCommand

'--- CloseConnection will close the associated connection when
'--- the reader is closed
Dim rdr As SqlDataReader = cmd.ExecuteReader(CommandBehavior.CloseConnection)

Try
    Dim wt As New WordTemplate();
    wt.Open(templatePath)
    wt.SetDataSource(rdr)
    wt.Process()
    wt.Save(Page.Response, "DataReader.doc", False)
Finally
    If Not rdr Is Nothing Then
        rdr.Close()
    End If
End Try
```