.NET Framework 4

**DataAdapter DataTable and DataColumn Mappings (ADO.NET)**

A **DataAdapter** contains a collection of zero or more [DataTableMapping](http://msdn.microsoft.com/en-us/library/system.data.common.datatablemapping.aspx) objects in its **TableMappings** property. A **DataTableMapping** provides a master mapping between the data returned from a query against a data source, and a [DataTable](http://msdn.microsoft.com/en-us/library/system.data.datatable.aspx). The **DataTableMapping** name can be passed in place of the **DataTable** name to the **Fill** method of the **DataAdapter**. The following example creates a **DataTableMapping** named **AuthorsMapping** for the **Authors** table.

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl03_code');" \o "Copy Code)

workAdapter.TableMappings.Add("AuthorsMapping", "Authors")

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl04_code');" \o "Copy Code)

workAdapter.TableMappings.Add("AuthorsMapping", "Authors");

A **DataTableMapping** enables you to use column names in a **DataTable** that are different from those in the database. The **DataAdapter** uses the mapping to match the columns when the table is updated.

If you do not specify a **TableName** or a **DataTableMapping** name when calling the **Fill** or **Update** method of the **DataAdapter**, the **DataAdapter** looks for a **DataTableMapping** named "Table". If that **DataTableMapping** does not exist, the **TableName** of the **DataTable** is "Table". You can specify a default **DataTableMapping** by creating a **DataTableMapping** with the name of "Table".

The following code example creates a **DataTableMapping** (from the [System.Data.Common](http://msdn.microsoft.com/en-us/library/system.data.common.aspx) namespace) and makes it the default mapping for the specified **DataAdapter** by naming it "Table". The example then maps the columns from the first table in the query result (the **Customers** table of the **Northwind** database) to a set of more user-friendly names in the **Northwind Customers** table in the [DataSet](http://msdn.microsoft.com/en-us/library/system.data.dataset.aspx). For columns that are not mapped, the name of the column from the data source is used.

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl07_code');" \o "Copy Code)

Dim mapping As DataTableMapping = \_

adapter.TableMappings.Add("Table", "NorthwindCustomers")

mapping.ColumnMappings.Add("CompanyName", "Company")

mapping.ColumnMappings.Add("ContactName", "Contact")

mapping.ColumnMappings.Add("PostalCode", "ZIPCode")

adapter.Fill(custDS)

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl08_code');" \o "Copy Code)

DataTableMapping mapping =

adapter.TableMappings.Add("Table", "NorthwindCustomers");

mapping.ColumnMappings.Add("CompanyName", "Company");

mapping.ColumnMappings.Add("ContactName", "Contact");

mapping.ColumnMappings.Add("PostalCode", "ZIPCode");

adapter.Fill(custDS);

In more advanced situations, you may decide that you want the same **DataAdapter** to support loading different tables with different mappings. To do this, simply add additional **DataTableMapping** objects.

When the **Fill** method is passed an instance of a **DataSet** and a **DataTableMapping** name, if a mapping with that name exists it is used; otherwise, a **DataTable** with that name is used.

The following examples create a **DataTableMapping** with a name of **Customers** and a **DataTable** name of **BizTalkSchema**. The example then maps the rows returned by the SELECT statement to the **BizTalkSchema** **DataTable**.

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl09_code');" \o "Copy Code)

Dim mapping As ITableMapping = \_

adapter.TableMappings.Add("Customers", "BizTalkSchema")

mapping.ColumnMappings.Add("CustomerID", "ClientID")

mapping.ColumnMappings.Add("CompanyName", "ClientName")

mapping.ColumnMappings.Add("ContactName", "Contact")

mapping.ColumnMappings.Add("PostalCode", "ZIP")

adapter.Fill(custDS, "Customers")

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl10_code');" \o "Copy Code)

ITableMapping mapping =

adapter.TableMappings.Add("Customers", "BizTalkSchema");

mapping.ColumnMappings.Add("CustomerID", "ClientID");

mapping.ColumnMappings.Add("CompanyName", "ClientName");

mapping.ColumnMappings.Add("ContactName", "Contact");

mapping.ColumnMappings.Add("PostalCode", "ZIP");

adapter.Fill(custDS, "Customers");

|  |
| --- |
| **Description: NoteNote** |
| If a source column name is not supplied for a column mapping or a source table name is not supplied for a table mapping, default names will be automatically generated. If no source column is supplied for a column mapping, the column mapping is given an incremental default name of **SourceColumn***N,* starting with **SourceColumn1**. If no source table name is supplied for a table mapping, the table mapping is given an incremental default name of **SourceTable***N*, starting with **SourceTable1**. |
| **Description: NoteNote** |
| We recommend that you avoid the naming convention of **SourceColumn***N* for a column mapping, or **SourceTable***N* for a table mapping, because the name you supply may conflict with an existing default column mapping name in the **ColumnMappingCollection** or table mapping name in the **DataTableMappingCollection**. If the supplied name already exists, an exception will be thrown. |

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifHandling Multiple Result Sets

If your **SelectCommand** returns multiple tables, **Fill** automatically generates table names with incremental values for the tables in the **DataSet**, starting with the specified table name and continuing on in the form **TableName***N*, starting with **TableName1**. You can use table mappings to map the automatically generated table name to a name you want specified for the table in the **DataSet**. For example, for a **SelectCommand** that returns two tables, **Customers** and **Orders**, issue the following call to **Fill**.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl30_ctl00_ctl00_code');" \o "Copy Code)

adapter.Fill(customersDataSet, "Customers")

Two tables are created in the **DataSet**: **Customers** and **Customers1**. You can use table mappings to ensure that the second table is named **Orders** instead of **Customers1**. To do this, map the source table of **Customers1** to the **DataSet** table **Orders**, as shown in the following example.

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl30_ctl00_ctl01_code');" \o "Copy Code)

adapter.TableMappings.Add("Customers1", "Orders")

adapter.Fill(customersDataSet, "Customers")