Importing Data

Table of Contents

WordApplication's ImportData method allows you to import blocks of data to a worksheet from a database or a rectangular array. ImportData is a method of both Element and Table. Element.importData returns a Table object representing a table in the document that contains the imported data.Table.importData adds an new row to an existing table.

Importing from a Database

You can import values from a database to a table in your document by passing the ImportData method a DataTable, DataView, SqlDataReader, OleDbDataReader, or AdomdDataReader.

The DataTable and DataView classes are in the System.Data namespace. Use an Import directive to import the namespace to the aspx page:

```
<%@ Import namespace="System.Data" %>
```

To import System.Data to a C# code-behind page (.aspx.cs), use:

```
using System.Data;
```

To import database values using OleDb, import the System.Data.OleDb namespace to your page. To import database values using SqlClient, import the System.Data.SqlClient namespace to your page.

To import values from a database to your document:

1. Connect to the database and execute a query to return a DataTable, DataView, SqlDataReader, OleDbDataReader, or AdomdDataReader, for example:

```
private DataTable GetEmployeeDataTable()
{
     string employeeSQL = "SELECT TOP 10 FirstName + ' ' + LastName As Name, " +
          "Title FROM Employees";

     DataTable dt = new DataTable();
     using(SqlConnection conn = new SqlConnection(connString))
          new SqlDataAdapter(employeeSQL, conn).Fill(dt);

     return dt;
}
```

2. Create a document:

```
WordApplication wwApp = new WordApplication();
Document doc = wwApp.Create();
```

3. Pass the DataTable returned from the database to ImportData:

```
Table employeesTable = doc.ImportData(dt);
```

Importing from an Array

1. Create a rectangular array, for example:

```
string[,] arrayData = {{"Nancy", "Davolio", "Sales Manager"},
     {"Michael", "Suyama", "HR Representative"},
     {"Adrian", "King", "IS Support"}};
```

2. Create a document:

```
WordApplication wwApp = new WordApplication();
Document doc = wwApp.Create();
```

3. Pass the array to ImportData:

```
Table employeesTable = doc.ImportData(arrayData);
```

Customizing Your Data Import

The DataImportProperties class contains a set of properties that are used when importing data to a table in a document. The settings of a DataImportProperties object will be applied to a data import if the object is passed to ImportData (with the set of values to import). You can create several DataImportProperties objects and assign a different one to each data import, or re-use one object in multiple ImportData calls.

To customize a data import using a DataImportProperties object:

1. Create a DataImportProperties object:

```
WordApplication wwApp = new WordApplication();
Document doc = wwApp.Create();
DataImportProperties importProps = doc.CreateDataImportProperties();
```

2. Set one or more data import properties:

```
//--- Import column names to the first row of the table.
importProps.UseColumnNames = true;

//--- Automatically resize the table's columns and rows to fit
//--- the imported data.
importProps.AutoFit = true;
```

3. Define a DataTable, DataView, SqlDataReader, OleDbDataReader, AdomdDataReader or rectangular array, for example:

```
OleDbConnection Conn = new OleDbConnection();
DataTable employeeDt = null;
int employeeId;

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);
}
```

4. Pass the data and the DataImportProperties object to ImportData:

```
Table importTable = doc.ImportData(employeeDt , importProps);
```