

WordTemplate.SetMailMerge(System.Data.IDataReader, Int32)

Description

Sets the MailMerge's data source to a two-dimensional (rectangular) array of objects. A MailMerge by default treats the page content as a repeat block, repeating the page content for each row unless NEXT fields are used. If the [WordTemplate.EnableNEXTFields](#) property is set to true, the NEXT field can also be used to indicate that the next row of data should be inserted instead of the current row at the next occurrence of the merge fields.

C#

```
public void SetMailMerge(System.Data.IDataReader dr, int maxRows)
```

vb.net

```
Public Sub SetMailMerge(ByVal dr As System.Data.IDataReader, ByVal maxRows As Integer)
```

Parameters

dr

A SqlDataReader, OleDbDataReader, or AdomdDataReader to use as the data source.

maxRows

Specifies the maximum number of rows to import from the data source. To import the maximum rows available, use the constant [WordTemplate.ALL_ROWS](#).

Exceptions

ArgumentNullException

[Save](#) will throw this exception if null (C#) or Nothing (VB.NET) is passed to the method.

ArgumentException

Remarks

You can call [SetMailMerge](#) once for each instance of [WordTemplate](#). If you are using the Word 2003 binary template file type (.doc/.dot), you can call SetMailMerge or SetRepeatBlock, but not both.

Additionally, only the page content is repeated for each row, not the entire page itself. If you wish to have the page itself repeat for each row, you will need to remember to place a page break at the bottom of the page. Alternatively, you can create a hidden page break at the top of the page as follows:

- Put the cursor at the top of the document
- Go to Page Layout and open the paragraph formatting dialog
- On the Line and Page Breaks tab, select "Page Break Before"

Merge Fields for using the SetMailMerge method must not specify a data source – the data source is implied, and using a data source name will cause WordTemplate to throw an error. Valid merge field formats for use with SetMailMerge include field names («**fieldname**») and field ordinals («**#1**»).

Introduced in build 8.2

[SetMailMerge](#) will now work with headers and footers. A section break is required instead of a page break if each header or footer will be different.

Examples

C#

```
//--- Create connection string for Excel file data
string connString =
    "Provider=Microsoft.Jet.OLEDB.4.0;" +
    "Data Source=" + Server.MapPath("data/MailMergeData.xls") + ";" +
    "Extended Properties=Excel 8.0;";

//--- Use OleDb to get all data from Sheet1 of the MailMergeData.xls
//--- spreadsheet
//--- Place all of the data in a DataReader object
OleDbConnection conn = new OleDbConnection(connString);
conn.Open();
OleDbCommand cmd = new OleDbCommand("SELECT * FROM [Sheet1$]", conn);
OleDbDataReader dr = cmd.ExecuteReader();
WordTemplate wt = new WordTemplate();
wt.Open(Server.MapPath("template/MergeFieldTest.doc"));

//--- Set the mail merge
//--- The data source is the DataReader
//--- MaxRows is set to ALL_ROWS, which allows all rows to be imported
wt.SetMailMerge(dr, WordTemplate.ALL_ROWS);
wt.Process();
wt.Save(Page.Response, "output.doc", false);
```

```
'--- Create connection string for Excel file data
Dim connString As String = _
    "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & Server.MapPath("data/MailMergeData.xls") & ";" & _
    "Extended Properties=Excel 8.0;"

'--- Use OleDb to get all data from Sheet1 of the MailMergeData.xls
spreadsheet
'--- Place all of the data in a DataReader object
Dim conn As New OleDbConnection(connString)
conn.Open()
Dim cmd As New OleDbCommand("SELECT * FROM [Sheet1$]", conn)
Dim dr As OleDbDataReader = cmd.ExecuteReader()
Dim wt As New WordTemplate()
wt.Open(Server.MapPath("template\MergeFieldTest.doc"))

'--- Set the mail merge
'--- The data source is the DataReader
'--- MaxRows is set to ALL_ROWS, which allows all rows to be imported
wt.SetMailMerge(dr, WordTemplate.ALL_ROWS)
wt.Process()
wt.Save(Page.Response, "output.doc", False)
```