

# WordTemplate.SetMailMerge(Object>(), String(), Int32, Boolean)

## 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. This should largely imitate the behavior of a Microsoft Word Mail merge.

### C#

```
public void SetMailMerge(System.Object[][] jaggedArray, System.String[] columnNames,
int maxRows, boolean transpose)
```

### vb.net

```
Public Sub SetMailMerge(ByVal jaggedArray As Object>(), ByVal columnNames As
String(), ByVal maxRows As Integer, ByVal transpose As Boolean)
```

## Parameters

### *jaggedArray*

The jagged array (array-of-arrays) to use as the data source.

### *columnNames*

The one-dimensional array of strings that represent the field names to be replaced by the data. These names must match the column names from your 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](#).

### *transpose*

If transpose is set to `true`, the array is treated as `Object[column][row]`. If transpose is set to `false`, the array is treated as `Object[row][column]`.

## 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#

```
...
//--- A 2-D jagged array of values
//--- This is a "transposed" array with columns in the first
//--- dimension and rows in the second
object[][] data = new object[][]{
    new string[]{"Knoxville", "Boston",
        "Washington", "Seattle",
        "Chicago", "New York",
        "Atlanta", "Los Angeles",
        "Houston"},
    new string[]{"Tennessee", "Massachusetts",
        "DC", "Washington",
        "Illinois", "New York",
        "Georgia", "California",
        "Texas"}
};

//--- Names array, elements correspond to merge field names
string[] names = new string[] { "City", "State" };
WordTemplate wt = new WordTemplate();
wt.Open(Server.MapPath("template/MergeFieldTest.doc"));

//--- Set the mail merge
//--- The data source is the 2-D jagged data array
//--- MaxRows is set to ALL_ROWS, which allows all rows to be imported
//--- Transpose is set to true to handle the transposed array
wt.SetMailMerge(data, names, WordTemplate.ALL_ROWS, true);
wt.Process();
wt.Save(Page.Response, "output.doc", false);
```

```
'--- A 2-D jagged array of values
'--- This is a "transposed" array with columns in the first
'--- dimension and rows in the second
Dim data()() As Object = { _
    New Object() {"Knoxville", "Boston", _
        "Washington", "Seattle", _
        "Chicago", "New York", _
        "Atlanta", "Los Angeles", _
        "Houston"}, _
    New Object() {"Tennessee", "Massachusetts", _
        "DC", "Washington", _
        "Illinois", "New York", _
        "Georgia", "California", _
        "Texas"} _
}

'--- names array, elements correspond to merge field names
Dim names() As String = {"City", "State"}
Dim wt As New WordTemplate()
wt.Open(Server.MapPath("template/MergeFieldTest.doc"))

'--- Set the mail merge
'--- The data source is the jagged data array
'--- MaxRows is set to ALL_ROWS, which allows all rows to be imported
'--- Transpose is set to true to handle the transposed array
wt.SetMailMerge(data, names, WordTemplate.ALL_ROWS, True)
wt.Process()
wt.Save(Page.Response, "output.doc", False)
```