

GemBox Support Center

Portal > Knowledgebase > GemBox.Document > Mail Merge remove empty table

Mail Merge remove empty table

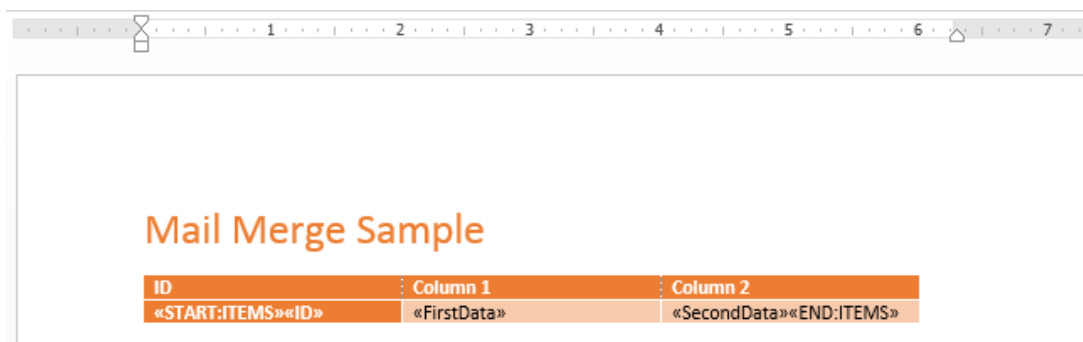
Mario - GemBox - 2014-06-09 - 0 Comments - in GemBox.Document

When working with mail merge, one of the common uses is to execute a mail merge operation over the Table element in order to get a dynamic number of TableRows (depending on the data source records).

Demonstration sample of this feature can be found on the following link:

<https://www.gemboxsoftware.com/document/examples/mail-merge-ranges/903>

Also, here is another sample which uses the following "[Mail Merge Sample.docx](#)" file as an input:



The following code will perform a mail merge on "ITEMS" range and import the data from the data source.

Note that an input file contains range name prefixes different from the ones that GemBox.Document uses by default.

C# code

```
DocumentModel document = DocumentModel.Load("Mail Merge Sample.docx");
```

```
// Change the prefix that indicates range start, default is "RangeStart:".
```

```
document.MailMerge.RangeStartPrefix = "START:";
```

```
// Change the prefix that indicates range end, default is "RangeEnd:".
```

```
document.MailMerge.RangeEndPrefix = "END:";
```

```
// Create a data source.
```

```
var dataSource = new object[]
```

```
{
```

```
    new { ID = 1, FirstData = "Sample 1", SecondData = "Sample 1" },
```

```
    new { ID = 2, FirstData = "Sample 2", SecondData = "Sample 2" },
```

```
    new { ID = 3, FirstData = "Sample 3", SecondData = "Sample 3" },
```

```
    new { ID = 4, FirstData = "Sample 4", SecondData = "Sample 4" },
```

```
    new { ID = 5, FirstData = "Sample 5", SecondData = "Sample 5" },
```

```
};
```

```
document.MailMerge.ClearOptions = MailMergeClearOptions.RemoveEmptyRanges;
```

```
document.MailMerge.Execute(dataSource, "ITEMS");
```

```
document.Save("Sample Out Empty.docx");
```

VB.NET code

```
Dim document As DocumentModel = DocumentModel.Load("Mail Merge Sample.docx")
```

```
' Change the prefix that indicates range start, default is "RangeStart:".
```

```
document.MailMerge.RangeStartPrefix = "START:"
```

```
' Change the prefix that indicates range end, default is "RangeEnd:".
```

```
document.MailMerge.RangeEndPrefix = "END:"
```

```
' Create a data source.
```

```
Dim dataSource = New Object() _
```

```
{
```

```
    New With {.ID = 1, .FirstData = "Sample 1", .SecondData = "Sample 1"},
```

```
    New With {.ID = 1, .FirstData = "Sample 2", .SecondData = "Sample 2"},
```

```
    New With {.ID = 1, .FirstData = "Sample 3", .SecondData = "Sample 3"},
```

```
    New With {.ID = 1, .FirstData = "Sample 4", .SecondData = "Sample 4"},
```

```
    New With {.ID = 1, .FirstData = "Sample 5", .SecondData = "Sample 5"}
```

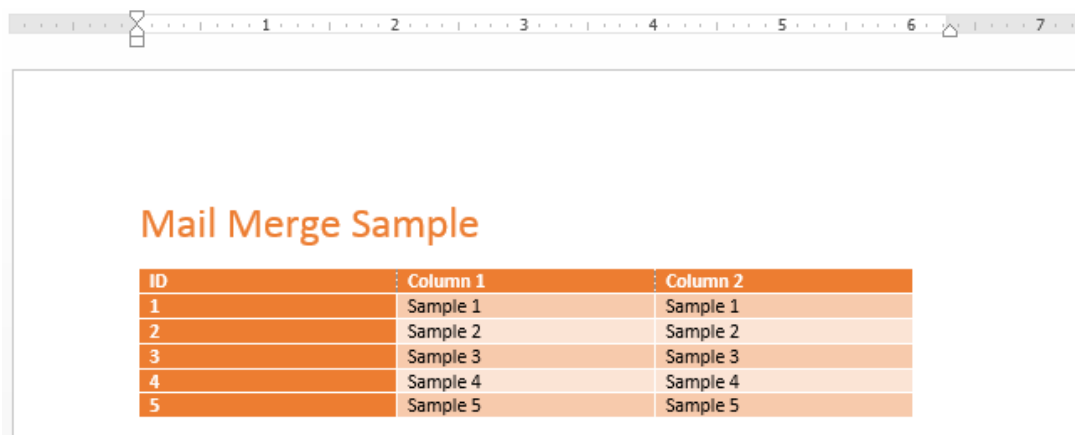
```
}
```

```
document.MailMerge.ClearOptions = MailMergeClearOptions.RemoveEmptyRanges
```

```
document.MailMerge.Execute(dataSource, "ITEMS")
```

```
document.Save("Sample Out Empty.docx")
```

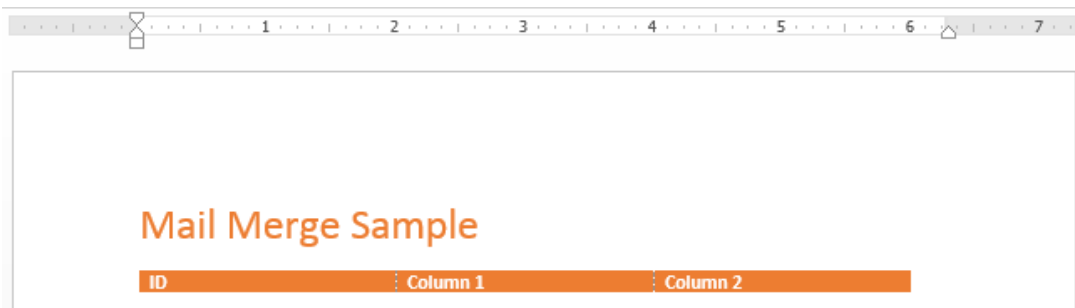
The following is the resulting "Sample Out.docx" file:



The screenshot shows a Word document titled "Mail Merge Sample". It contains a table with 5 rows and 3 columns. The first row is the header, and the following four rows contain data. The table is styled with orange headers and light orange data cells.

ID	Column 1	Column 2
1	Sample 1	Sample 1
2	Sample 2	Sample 2
3	Sample 3	Sample 3
4	Sample 4	Sample 4
5	Sample 5	Sample 5

Data source values are successfully imported into a Table element, but if we were to use an empty data source or a data source with all the values equal to null or empty, the resulting file would be the following:



The screenshot shows a Word document titled "Mail Merge Sample". It contains a table with 1 row and 3 columns. The first row is the header, and the following rows are empty. The table is styled with orange headers and light orange data cells.

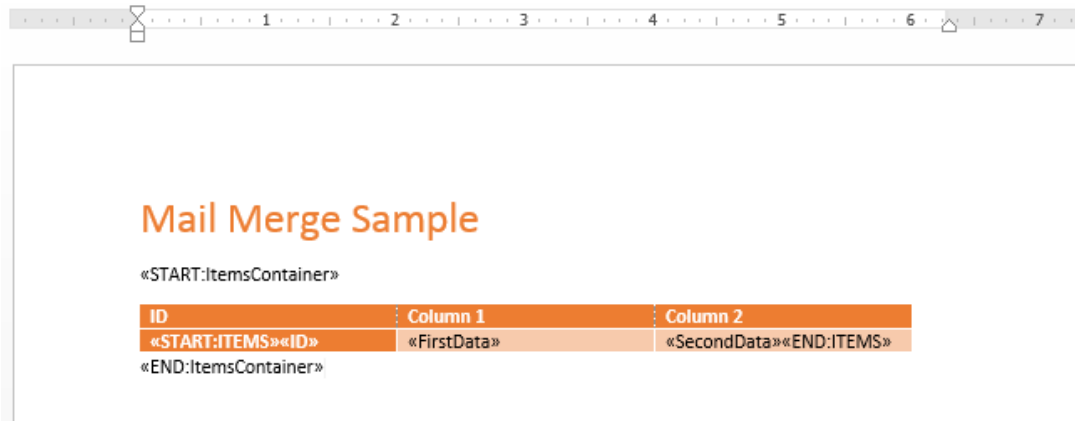
ID	Column 1	Column 2

Due to the use of `MailMergeClearOptions.RemoveEmptyRanges`, the resulting output file will not contain the "ITEMS" range, but the Table element's header row still remains because it is not part of the range.

In order to remove the whole Table element for empty data sources, we will define another

merge range "ItemsContainer" which will contain an existing "ITEMS" range.

The following is a "[Mail Merge Sample 2.docx](#)" file that will be used as an input:



The screenshot shows a Word document with a ruler at the top. The main content is a table with three columns: 'ID', 'Column 1', and 'Column 2'. The table is enclosed in a range named 'ItemsContainer'. The first row of the table contains the headers. The second row contains the data: '«START:ITEMS»«ID»', «FirstData», and «SecondData»«END:ITEMS».

ID	Column 1	Column 2
«START:ITEMS»«ID»	«FirstData»	«SecondData»«END:ITEMS»

In order to use this new input file, we will have to adjust the data source used for a mail merge execution, as the following:

C# code

// Create a data source.

```
var dataSource = new
{
    ITEMS = new object[]
    {
        new { ID = 1, FirstData = "Sample 1", SecondData = "Sample 1" },
        new { ID = 2, FirstData = "Sample 2", SecondData = "Sample 2" },
        new { ID = 3, FirstData = "Sample 3", SecondData = "Sample 3" },
        new { ID = 4, FirstData = "Sample 4", SecondData = "Sample 4" },
        new { ID = 5, FirstData = "Sample 5", SecondData = "Sample 5" },
    }
};
```

```
document.MailMerge.ClearOptions = MailMergeClearOptions.RemoveEmptyRanges;
document.MailMerge.Execute(dataSource, "ItemsContainer");
document.Save("Sample Out Empty.docx");
```

VB.NET code

' Create a data source.

```
Dim dataSource = New With
{
    .ITEMS = New Object() _
    {
        New With {.ID = 1, .FirstData = "Sample 1", .SecondData = "Sample 1"},
        New With {.ID = 1, .FirstData = "Sample 2", .SecondData = "Sample 2"},
        New With {.ID = 1, .FirstData = "Sample 3", .SecondData = "Sample 3"},
        New With {.ID = 1, .FirstData = "Sample 4", .SecondData = "Sample 4"},
        New With {.ID = 1, .FirstData = "Sample 5", .SecondData = "Sample 5"}
    }
}
```

```
document.MailMerge.ClearOptions = MailMergeClearOptions.RemoveEmptyRanges
document.MailMerge.Execute(dataSource, "ItemsContainer")
document.Save("Sample Out Empty.docx")
```

But now if we were to use an empty data source, we would achieve the required behaviour

of removing the entire Table element and the resulting output file would be as shown on the following image:

