

GemBox Support Center

Portal > Knowledgebase > GemBox.Document > How does MailMergeClearOptions.RemoveEmptyRanges work

How does MailMergeClearOptions.RemoveEmptyRanges work

Mario - GemBox - 2017-02-10 - 0 Comments - in GemBox.Document

The common issue when using the [RemoveEmptyRanges](#) is that if we use this clean option and do not specify a mail merge range name, we will unintentionally clean the whole DocumentModel's content.

To explain the reason of this behaviour, we will first reproduce this issue. Let's start by creating a small DocumentModel object.

C# code

```
var document = new DocumentModel();

// Sample content that we do not want to remove.
var para1 = new Paragraph(document, "RemoveEmptyRanges Sample!");

// Sample range that we want to remove.
var para2 = new Paragraph(document,
    new Field(document, FieldType.MergeField, "RangeStart:RangeSample"),
    new Field(document, FieldType.MergeField, "FieldSample"),
    new Field(document, FieldType.MergeField, "RangeEnd:RangeSample"));

document.Sections.Add(new Section(document, para1, para2));
```

VB.NET code

```
Dim document As New DocumentModel()

' Sample content that we do not want to remove.
Dim para1 = New Paragraph(document, "RemoveEmptyRanges Sample!")

' Sample range that we want to remove.
Dim para2 = New Paragraph(document, _
    New Field(document, FieldType.MergeField, "RangeStart:RangeSample"), _
    New Field(document, FieldType.MergeField, "FieldSample"), _
    New Field(document, FieldType.MergeField, "RangeEnd:RangeSample"))

document.Sections.Add(New Section(document, para1, para2))
```

Now we will call the [Execute](#) method, which will execute the mail merge operation with the empty range name, a simple data source and a RemoveEmptyRanges clear option.

C# code

```
document.MailMerge.ClearOptions = MailMergeClearOptions.RemoveEmptyRanges;
var dataSource = new { None = "Sample data source" };
document.MailMerge.Execute(dataSource, string.Empty);
document.Save("Clean sample.docx");
```

VB.NET code

```
document.MailMerge.ClearOptions = MailMergeClearOptions.RemoveEmptyRanges
Dim dataSource = New With {.None = "Sample data source"}
document.MailMerge.Execute(dataSource, String.Empty)
document.Save("Clean sample.docx")
```

The resulting "Clean sample.docx" file is completely empty, having no content at all.

The reason for this is the following:

When starting the mail merge process, the GemBox.Document will first locate a mail merge range in which it will operate. You can read more details about how the GemBox.Document identifies a mail merge range [here](#).

The important thing is that if the *range name* is of value *null* or *empty*, the mail merge range is the whole document content.

After the mail merge execution, the GemBox.Document will remove the appropriated parts of the DocumentModel, depending on the ClearOptions which we used. In our case, we specified that we want to remove the whole mail merge range if the execution hasn't imported any data to that range.

And because the mail merge range was resolved to be the whole document content, we have indicated to the GemBox.Document to remove everything.

To avoid this issue, we need to remember to specify a range name if we are using the RemoveEmptyRanges option.

C# code

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

VB.NET code

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

In this case, we get the resulting "Clean sample.docx" file:

