

# GemBox Support Center

Portal > Knowledgebase > GemBox.Document > How to create a chart in a Word file

---

## How to create a chart in a Word file

Mario - GemBox - 2018-03-28 - 0 Comments - in GemBox.Document

Currently GemBox.Document lacks API support for chart elements; they are supported only through preservation in a DOCX file.

However, our other component, [GemBox.Spreadsheet](#), does have API support for them and is able to [create a chart element in an Excel file](#).

So, by using both GemBox.Spreadsheet and GemBox.Document, we can accomplish this task. With GemBox.Spreadsheet, we can create a chart element and save it to an image stream while with GemBox.Document, we can create a picture element from that image stream and add it to our document.

The following example demonstrates how to create a chart, export it as image stream and add it into a document.

### C# code

```
class Program
{
    static void Main(string[] args)
    {
        ComponentInfo.SetLicense("FREE-LIMITED-KEY");
        SpreadsheetInfo.SetLicense("FREE-LIMITED-KEY");

        // Create chart as an image stream.
        MemoryStream chartStream = CreateChart(
            new string[] { "John Doe", "Fred Nurk", "Hans Meier" },
            new double[] { 3300, 2200, 1100 });

        // Create a document.
        var document = new DocumentModel();

        // Add an image.
        document.Sections.Add(
            new Section(document,
                new Paragraph(document,
                    new Picture(document, chartStream))));

        // Save a document.
```

```

        document.Save("Sample.docx");
    }

    static MemoryStream CreateChart(string[] names, double[] values)
    {
        // Create a spreadsheet.
        var workbook = new ExcelFile();
        var worksheet = workbook.Worksheets.Add("Sheet1");

        // Add data that will be used by chart.
        int count = names.Length;
        for (int i = 0; i < count; i++)
        {
            worksheet.Cells[i, 0].Value = names[i];
            worksheet.Cells[i, 1].Value = values[i];
        }

        // Create chart and select data for it.
        var chart = (BarChart)worksheet.Charts.Add(ChartType.Bar, "D2", "J15");
        chart.SelectData(worksheet.Cells.GetSubrangeRelative(0, 0, 2, count), true, false, true);
        chart.Series[0].IsLegendEntryVisible = false;

        // Save chart to an image stream.
        var imageStream = new MemoryStream();
        var imageOptions = GemBox.Spreadsheet.SaveOptions.ImageDefault;

        chart.Format().Save(imageStream, imageOptions);

        return imageStream;
    }
}

```

## VB.NET code

```
Module Program
```

```
    Sub Main()
```

```
        ComponentInfo.SetLicense("FREE-LIMITED-KEY")
```

```
        SpreadsheetInfo.SetLicense("FREE-LIMITED-KEY")
```

```

' Create chart as an image stream.
Dim chartStream As MemoryStream = CreateChart(
    New String() {"John Doe", "Fred Nurk", "Hans Meier"},
    New Double() {3300, 2200, 1100})

' Create a document.
Dim document = New DocumentModel()

' Add an image.
document.Sections.Add(
    New Section(document,
        New Paragraph(document,
            New Picture(document, chartStream))))

' Save a document.
document.Save("Sample.docx")
End Sub

Private Function CreateChart(names As String(), values As Double(
)) As MemoryStream
    ' Create a spreadsheet.
    Dim workbook = New ExcelFile()
    Dim worksheet = workbook.Worksheets.Add("Sheet1")

    ' Add data that will be used by chart.
    Dim count = names.Length
    For i = 0 To count - 1
        worksheet.Cells(i, 0).Value = names(i)
        worksheet.Cells(i, 1).Value = values(i)
    Next

    ' Create chart and select data for it.
    Dim chart = DirectCast(worksheet.Charts.Add(ChartType.Bar, "D
2", "J15"), BarChart)
    chart.SelectData(worksheet.Cells.GetSubrangeRelative(0, 0, 2,
count), True, False, True)
    chart.Series(0).IsLegendEntryVisible = False

    ' Save chart to an image stream.
    Dim imageStream = New MemoryStream()
    Dim imageOptions = GemBox.Spreadsheet.SaveOptions.ImageDefaul

```

t

```
chart.Format().Save(imageStream, imageOptions)
```

```
Return imageStream
```

```
End Function
```

```
End Module
```

This is the resulting **Sample.docx** file:

