

12 Excel-Diagramme (Excel 2003 und früher)

Nicht nur Excel-Tabellen können in ein Word-Dokument eingebettet werden. Auch Diagramme werden als OLE-Objekte unterstützt und verleihen dem Textinhalt eine zusätzliche Ausdruckskraft.

Profitipp Beginn

Wegen der mit ihnen gespeicherten Datei-Strukturen lassen OLE-Objekte die Dateigröße stark anwachsen. Sofern Sie kein dynamisches Diagramm brauchen, raten wir daher, das Diagramm als Grafik zu kopieren und einzufügen: Zunächst markieren Sie es in Excel und öffnen bei gedrückter (**Umschalt**)-Taste das Menü *Bearbeiten*. Darin wählen Sie nun den Befehl *Bild kopieren*. Im Word-Dokument schließlich wird aus dem Dialogfeld zum Menübefehl *Bearbeiten/Inhalte einfügen* der gewünschte Grafik-Typ festgelegt.



2007

In Excel 2007 befindet sich der Befehl *Als Bild* im Dropdownmenü zur Schaltfläche *Einfügen* auf der Registerkarte *Start*.

Profitipp Ende

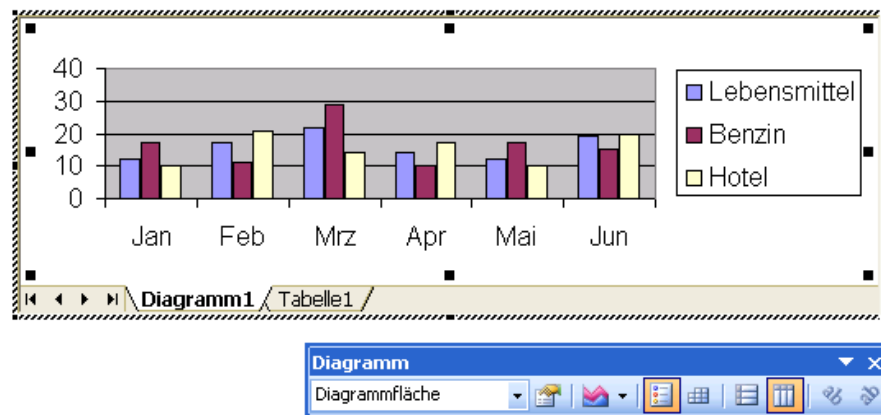
Der Makrorekorder verrät folgende Syntax, um ein Excel-Diagramm einzufügen, wobei der Klassentyp »Excel.Chart.8« verwendet wird:

```
Selection.InlineShapes.AddOLEObject ClassType:="Excel.Chart.8",  
    FileName:= "", LinkToFile:=False, DisplayAsIcon:=False
```

Während das Einfügen einer Excel-Tabelle nur ein einziges Arbeitsblatt im OLE-Objekt generiert, erhält das neue Diagramm-Objekt zuvorderst ein Diagrammblatt und dahinter ein Tabellenblatt (siehe Abbildung 12.1), welches die Daten für das Diagramm enthält. Der Automatisierungscode hat demzufolge zwei Aufgaben zu erledigen:

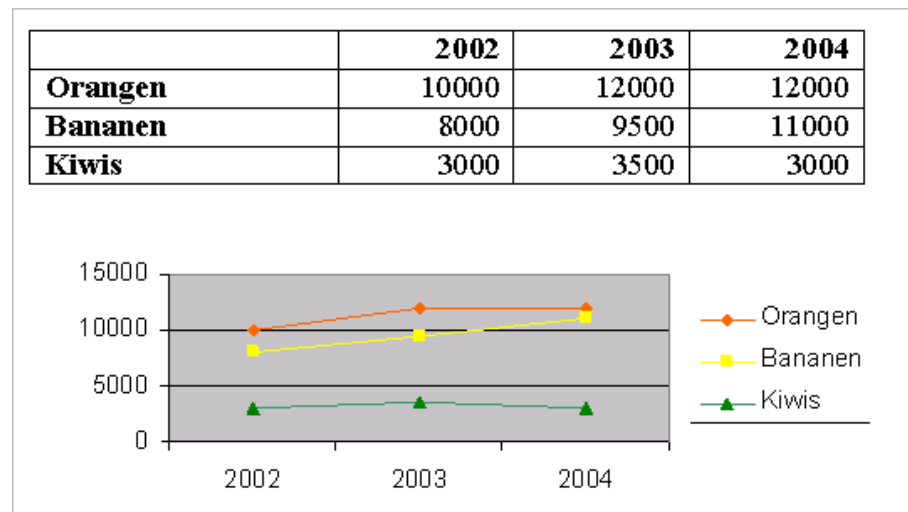
- die im Tabellenblatt hinterlegten Beispieldaten durch die benötigten zu ersetzen und
- das Diagramm nach Wunsch zu formatieren.

Abbildung 12.1: Standardmäßig umfasst ein neues Diagramm-Objekt, ein Diagramm- sowie ein Tabellenblatt



Unser folgendes Beispiel zeigt, wie die Daten aus einer im Dokument befindlichen Word-Tabelle in die Excel-Tabelle übernommen werden. Anschließend wird die Datenquelle des Diagramms angepasst sowie der Typ und die Formatierung geändert. Das Resultat sehen Sie in Abbildung 12.2, den Beispielcode in Listing 12.1.

Abbildung 12.2: Diagramm erstellt auf Basis einer Word-Tabelle. Die Tabellendaten werden in das Excel-Tabellenblatt hinter dem Diagramm eingetragen.



Den Eingangspunkt bildet die Prozedur *DiagrammEinfuegen*. Als erster Schritt wird die erste Tabelle im Dokument einer Objektvariablen zugewiesen. Der Zielbereich für das Diagramm wird in einem darauf folgenden Absatz festgelegt.

Danach wird ein ADO-Recordset angelegt und, zusammen mit der Word-Tabelle, der Prozedur *DatenLaden* übergeben. Hier sehen Sie, wie ein ADO-Recordset von Grund auf erstellt und mit Daten gefüllt wird, ohne Verbindung zu einer Datenbank als Datenquelle. Die erste Tabellenzeile liefert die Feldnamen. Dann wird durch die übrigen Zeilen geschleift, um Datensätze zu bilden.

Nun, da die Daten bereit liegen, wird das Diagramm-Objekt eingefügt und aktiviert. Arbeitsmappe, Tabellenblatt sowie Diagrammblatt werden ihren Objekt-Variablen

zugewiesen. Zunächst werden die Beispieldaten aus dem Tabellenblatt gelöscht, dann die Spaltenüberschriften, die als Diagramm-Legende erscheinen, anhand der Feldnamen des ADO-Recordsets angelegt. Die Daten werden, dank Excels CopyFromRecordset-Methode, in einem Schritt eingefügt.

Schließlich verpasst die Prozedur *DiagrammFormatieren* dem Diagramm den letzten Schliff. Der Diagrammtyp wird festgelegt. Anhand der Adresseninformation des benutzten Tabellenblattbereichs werden Quellbereiche für Diagramm- sowie X-Achsen-Werte ausgerechnet und zugewiesen. Auch die farbliche und förmliche Gestaltung der Datenserien werden angepasst.

Listing 12.1: Ein Excel-Diagramm anhand einer Tabelle im Word-Dokument erstellen

```
Sub DiagrammEinfuegen()
    Dim xlDiagramm As Excel.Chart
    Dim xlMappe As Excel.Workbook
    Dim xlBlatt As Excel.Worksheet
    Dim doc As Word.Document
    Dim rng As Word.Range
    Dim ils As Word.InlineShape
    Dim oleF As Word.OLEFormat
    Dim tbl As Word.Table
    Dim rs As ADODB.Recordset
    Dim fld As ADODB.Field
    Dim lZaehler As Long

    Set doc = ActiveDocument
    Set tbl = doc.Tables(1)
    Set rng = tbl.Range
    'Diagramm unter der Tabelle einfügen
    rng.Collapse Direction:=wdCollapseEnd
    rng.InsertAfter vbCr
    rng.Collapse Direction:=wdCollapseEnd

    'Tabellendaten in ein ADODB-Recordset lesen
    Set rs = New ADODB.Recordset
    DatenLaden rs, tbl
    'Nach Füllen des Recordsets ist der letzte Datensatz markiert.
    'Deshalb muss der erste Datensatz angewählt werden.
    rs.MoveFirst

    'Das Diagramm im Dokument erstellen ...
    Set oleF = doc.InlineShapes.AddOLEObject(ClassType:="Excel.Chart.8", _
        Range:=rng).OLEFormat
    '... und aktivieren.
    oleF.DoVerb

    Der Arbeitsmappe, dem Datenblatt sowie Diagrammblatt Objektvariablen zuweisen
    Set xlMappe = oleF.Object
    Set xlBlatt = xlMappe.Sheets(2)
    Set xlDiagramm = xlMappe.Charts(1)
    'Die vorhandenen Daten löschen.
    xlBlatt.UsedRange.Clear
    'Die Spaltenüberschriften auf Basis der Feldnamen des Recordsets erstellen
    lZaehler = 0
```

```
For Each fld In rs.Fields
    lZaehler = lZaehler + 1
    xlBlatt.Cells(1, lZaehler).Value = fld.Name
Next fld
'Die Daten, ab der 2. Tabellenblattzeile, einfügen.
xlBlatt.Range("A2").CopyFromRecordset rs

'Das Diagramm formatieren
DiagrammFormatieren xlDiagramm, xlBlatt

'Aufräumungsarbeiten
rs.Close
Set rs = Nothing
Set xlDiagramm = Nothing
Set xlBlatt = Nothing
Set xlMappe = Nothing
End Sub

Sub DatenLaden(ByRef rs As ADODB.Recordset, tbl As Word.Table)
    Dim row As Word.Row
    Dim cel As Word.Cell
    Dim rng As Word.Range
    Dim fld As ADODB.Field
    Dim lAnzFelder As Long
    Dim lZaehler As Long
    Dim lFeldTyp As Long
    Dim strZellenInhalt As String

    rs.CursorLocation = adUseClient
    rs.CursorType = adOpenDynamic
    rs.LockType = adLockOptimistic
    lAnzFelder = tbl.Columns.Count
    For Each row In tbl.Rows
        'Die erste Zeile (Spaltenüberschrift) enthält den Feldnamen.
        If row.Index = 1 Then
            'Die Felder des Recordsets definieren
            For Each cel In row.Cells
                Set rng = cel.Range
                'Sicherstellen, dass auch verborgener Text gelesen wird (Die erste Zelle enthält
                'eine Spaltenbezeichnung, weil ein Laufzeitfehler bei leerem Zelleninhalt
                'vorkommt)
                rng.TextRetrievalMode.IncludeHiddenText = True
                strZellenInhalt = TrimZelle(rng.Text) & ""
                'Der Feldtyp ist entweder eine Zeichenkette oder eine Zahl
                lFeldTyp = FeldTypErmitteln(strZellenInhalt)
                rs.Fields.Append Name:=strZellenInhalt, Type:=lFeldTyp, _
                DefinedSize:=30, Attrib:=adFldMayBeNull
                Set rng = Nothing
            Next
            rs.Open
        Else
            'Das Recordset mit Datensätzen füllen...
            rs.AddNew
```

```

        '... ein Datensatz pro Zeile
        For lZaehler = 1 To lAnzFelder
            rs.Fields(lZaehler - 1).Value = _
                TrimZelle(row.Cells(lZaehler).Range.Text)
        Next lZaehler
    End If
Next
End Sub

Sub DiagrammFormatieren(xlDiagramm As Excel.Chart, xlBlatt As Excel.Worksheet)
    Dim xlRange As Excel.Range
    Dim strBereichAdresse As String
    Dim lLetzteSpalte As Long
    Dim lLetzteZeile As Long

    With xlDiagramm
        .ChartType = xlLine
        strBereichAdresse = xlBlatt.UsedRange.AddressLocal(ReferenceStyle:=xlR1C1)
        lLetzteSpalte = Right(strBereichAdresse, 1)
        lLetzteZeile = Mid(strBereichAdresse, 7, 1)
        Set xlRange = xlBlatt.Range( _
            xlBlatt.Cells(1, 2), xlBlatt.Cells(1, lLetzteSpalte))
        .SetSourceData Source:=xlBlatt.Range( _
            xlBlatt.Cells(2, 1), xlBlatt.Cells(lLetzteZeile, lLetzteSpalte)), PlotBy:=xlRows

        'Orangen sind orange.
        DatenserieFormatieren .SeriesCollection(1), xlRange, xlMarkerStyleCircle, _
            5, RGB(255, 153, 0)
        'Bananen sind gelb.
        DatenserieFormatieren .SeriesCollection(2), xlRange, xlMarkerStyleDiamond, _
            5, RGB(255, 255, 0)
        'Kiwis sind grün.
        DatenserieFormatieren .SeriesCollection(3), xlRange, xlMarkerStyleSquare, _
            5, RGB(0, 128, 0)
    End With
End Sub

Sub DatenserieFormatieren(DatenSerie As Excel.Series, xlRange As Excel.Range, _
    xlDatenPunktStil As Long, xlDatenPunktGrösse As Long, _
    colorDatenPunktFarbe)

    With DatenSerie
        .XValues = xlRange
        .MarkerStyle = xlDatenPunktStil
        .MarkerSize = xlDatenPunktGrösse
        .Border.Color = colorDatenPunktFarbe
        .MarkerBackgroundColor = colorDatenPunktFarbe
        .MarkerForegroundColor = colorDatenPunktFarbe
        'Nur ab Excel 2007
        '.Format.Glow.Color = RGB(240, 0, 0)
        '.Format.Glow.Radius = 3 End With
    End Sub

```

```

Function TrimZelle(str)
    str = Mid(str, 1, Len(str) - 2)
    TrimZelle = str
End Function

Function FeldTypErmitteln(str) As Long
    If IsNumeric(str) Then
        FeldTypErmitteln = 3
    Else
        FeldTypErmitteln = 202
    End If
End Function

```



2007

Die Diagramm-Funktionalität wurde für Office 2007 überarbeitet. Die Aufzeichnung mit dem Makrorekorder liefert eine leere Prozedur, da für Office 2007-Diagramme kein OLE-Server vorhanden ist. Es kann jedoch der Code für ein »Excel.Chart.8« eingesetzt werden. Das Ergebnis ist ein Excel 2007-Diagramm, das alle Objekte des neuen Objektmodells unterstützt. Weiterhin besteht die Möglichkeit, zuerst eine Excel 2007-Kalkulationstabelle zu erstellen und anschließend ein Diagrammblatt einzufügen, wie das Listing 12.2 veranschaulicht.

Listing 12.2: Der Weg zu einem Excel 2007-Diagramm führt über das Erstellen eines Excel-Tabellen-Objekts

```

Sub Excel2007DiagrammErstellen()
    Dim rng As Word.Range
    Dim doc As Word.Document
    Dim wb As Excel.Workbook
    Dim oSheet As Excel.Worksheet
    Dim xlRng As Excel.Range
    Dim o As Object 'Das Diagramm

    Set doc = ActiveDocument
    Set rng = doc.Content
    'Das Tabellen-Objekt wird am Dokumentende eingefügt.
    rng.Collapse wdCollapseEnd
    Set wb = ActiveDocument.InlineShapes.AddOLEObject( _
        "Excel.Sheet.12", Range:=rng).OLEFormat.Object
    Set oSheet = wb.Sheets(1)
    oSheet.Name = "Testdaten"
    'Die Daten in die Tabelle schreiben.
    oSheet.Range("B1").Value = "2004"
    oSheet.Range("C1").Value = "2005"
    oSheet.Range("D1").Value = "2006"
    oSheet.Range("E1").Value = "2007"
    oSheet.UsedRange.Font.Bold = True
    oSheet.Range("A2").Value = "Orangen"
    oSheet.Range("B2").Value = 10000
    oSheet.Range("C2").Value = 12000
    oSheet.Range("D2").Value = 12000
    oSheet.Range("E2").Value = 15000
    oSheet.Range("A3").Value = "Bananen"

```

```
oSheet.Range("B3").Value = 8000
oSheet.Range("C3").Value = 9500
oSheet.Range("D3").Value = 11000
oSheet.Range("E3").Value = 10000
oSheet.Range("A4").Value = "Kiwis"
oSheet.Range("B4").Value = 3000
oSheet.Range("C4").Value = 3500
oSheet.Range("D4").Value = 3000
oSheet.Range("E4").Value = 4000
Set xlRng = oSheet.UsedRange
'Ein Diagrammblatt einfügen...
Set o = wb.Charts.Add
'...und mit dem Datenbereich verbinden.
o.SetSourceData xlRng
'Als Linien-Diagramm festlegen
o.ChartType = xlLine
DiagrammFormatieren o, oSheet 'Ruft die Prozedur in Listing 12.1
End Sub
```

CD-ROM Beginn

Die Beispieldateien *Bsp12_02.doc* und *Bsp12_02_2007.docm* finden Sie auf der CD-ROM zum Buch im Ordner *\Beilagen\Kap12_XL8Diagramme\Bsp*.

CD-ROM Ende
