目录

前言

MSChart 是微软一款免费的图表控件，也可能是微软最好用的一款图表控件，其特点是在数组中构建好数据，赋值给控件，控件生成图表。这对于一些不喜欢在 EXCEL 中处理数据的朋友来说，是很好的选择，特别是有些特殊图表，需要添加很多辅助列，会破坏原有数据的布局。因为 MSChart 接受数组作为数据源，因此，可以方便的使用 EXCEL、数据库，甚至是自己虚拟出来的数据创建图表。本教程主要涉及：

1. 图表对象模型

1.1 控件属性

MSChart 控件包括：Plot、Backdrop、Footnote、Title 和 Legend 对象，Plot 是图表绘图区，是最重要的对象，Backdrop 描述在图表或图表元素后面的阴影、图案或图片，Footnote 是注脚，Title 是标题，Legend 是图例。Backdrop 和 Footnote 用得很少，Title 和 Legend 使用简单，就标题和图例，设置起来没有难度，唯一需要重点关注的是 Plot 对象。

MSChart 的重要属性：ChartType 和 ChartData，ChartType 属性返回或设置用于显示图表的图表类型，支持的图表类型如表 1：

常数	描述
VtChChartType3dBar	3D 条形图
VtChChartType2dBar	2D 条形图
VtChChartType3dLine	3D 折线图
VtChChartType2dLine	2D 折线图
VtChChartType3dArea	3D 面积图
VtChChartType2dArea	2D 面积图
VtChChartType3dStep	3D 阶梯图
VtChChartType2dStep	2D 阶梯图
VtChChartType3dCombination	3D 组合图
VtChChartType2dCombination	2D 组合图
VtChChartType2dPie	2D 饼图
VtChChartType2dXY	2D XY 散点图

经过多次试验，发现组合图不能组合散点图，想组合一些特殊的图表变得不可行。

ChartData 是一个数组，用于保存图表的数据和行、列标签。ChartData 数组的索引基于 0，第 0 行数据用于保存列标签（等效于图例），第 0 列用于保存行标签。ChartData 数组具有一定判断能力，如果你定义的 arr 数组的第一列不是字符串，控件会自动添加 R1、R2……行标签，如果 arr 数组的第一行不是字符串，控件会自动添加 C1、C2……列标签（系列名称，和图例 Legend 是一个意思），如图 1 所示。把单元格区域 A1:B10 赋值给数组 arr，然后把 arr 赋值给控件的属性 ChartData，再输出到单元格 G1:I11，数据多了一行和一列。

	A	B	C	D	E	F	G	H	I
1	5	5						C1	C2
2	9	5					R1	5	5
3	5	3					R2	9	5
4	7	2					R3	5	3
5	6	6					R4	7	2
6	7	9					R5	6	6
7	9	3					R6	7	9
8	2	8					R7	9	3
9	1	7					R8	2	8
10	8	7					R9	1	7
11							R10	8	7
12									
13									

图 1

输出的代码为：

```
Private Sub UserForm_Initialize()
    Dim arr()
    arr = Range("a1").CurrentRegion
    With MSChart1
        .ChartData = arr
        Range("g1").Resize(UBound(.ChartData) - LBound(.ChartData) + 1, _
            UBound(.ChartData, 2) - LBound(.ChartData, 2) + 1) = .ChartData
    End With
End Sub
```

细心的朋友应该会发现，arr 定义为 arr()，一个变体型数组，不能是 VBA 中常用的 Dim arr，把 arr 定义成变体型变量，否则提示类型错误。当然，也不能定义为单精度数组 Dim arr!()或双精度数组 Dim arr#()，因为 VBA 也不接受把单元格区域赋值给非变体型数组。ChartData 是基于 0 的数组，所以要用上标减去下标，并加 1.

用 ChartData 也可以方便的把图表数据输出，和普通数组的用法一样，不再赘叙。

如果不使用数组赋值，而是自己虚构一个，可以使用如下代码：

```
With MSChart1
    .ColumnCount = 8
    .RowCount = 8
    For c = 1 To 8
        For r = 1 To 8
            .Column = c
            .Row = r
            .Data = Rnd()
        Next
    Next
End With
```

上边的代码构建了一个 8 行 8 列的图表，其效果和使用数组是一样的。另外使用 ColumnCount 属性可以动态的添加新系列，如：.ColumnCount= .ColumnCount+1 就给图表新增了一个系列，可以通过赋值，绘制图表。这时只需要循环列 8 就行了（不能使用 ChartData 对单个数据点赋值，否则 EXCEL 应用程序会立即崩溃！）：

```
With MSChart1
    .ColumnCount = 9
     For r = 1 To 8
        .Row = r
        .Data = Rnd()
     Next
End With
```

1.2 Plot 对象

Plot 对象表示图表显示的区域，是 MSChart 最重要的对象。其下包括 Axis、Backdrop、SeriesCollection、Light、LocationRect、View3D、Wall 和 Plotbase 对象。Axis 对象代表图表的 x 、

y 和 z 轴（3D 类型时可见），BackDrop 对象代表坐标轴后面的区域，Light 对象代表图表的环境光和边缘光，LocationRect 对象代表图表的位置，SeriesCollection 对象代表系列的集合，View3D 对象代表立体图像的正视图和旋转图，Wall 对象代表三维图表中 y 坐标轴所在的平面区域，Plotbase 对象代表图表中正下方区域的外观（基座）。重要的对象是：Axis 和 SeriesCollection，后面会单独讲到，其他对象用得不多。

Plot 对象的 UniformAxis 属性指定图表的所有值坐标轴的单位刻度是否一致，一般要设置为 FALSE，即每个坐标轴的单位刻度不一致。

1.3 Axis 对象

Axis 对象表示图表上的所有坐标轴。其下包括 AxisGrid、AxisScale、AxisTitle、CategoryScale、Intersection、Labels、Pen、Tick 和 ValueScale 对象。如图 2 所示：

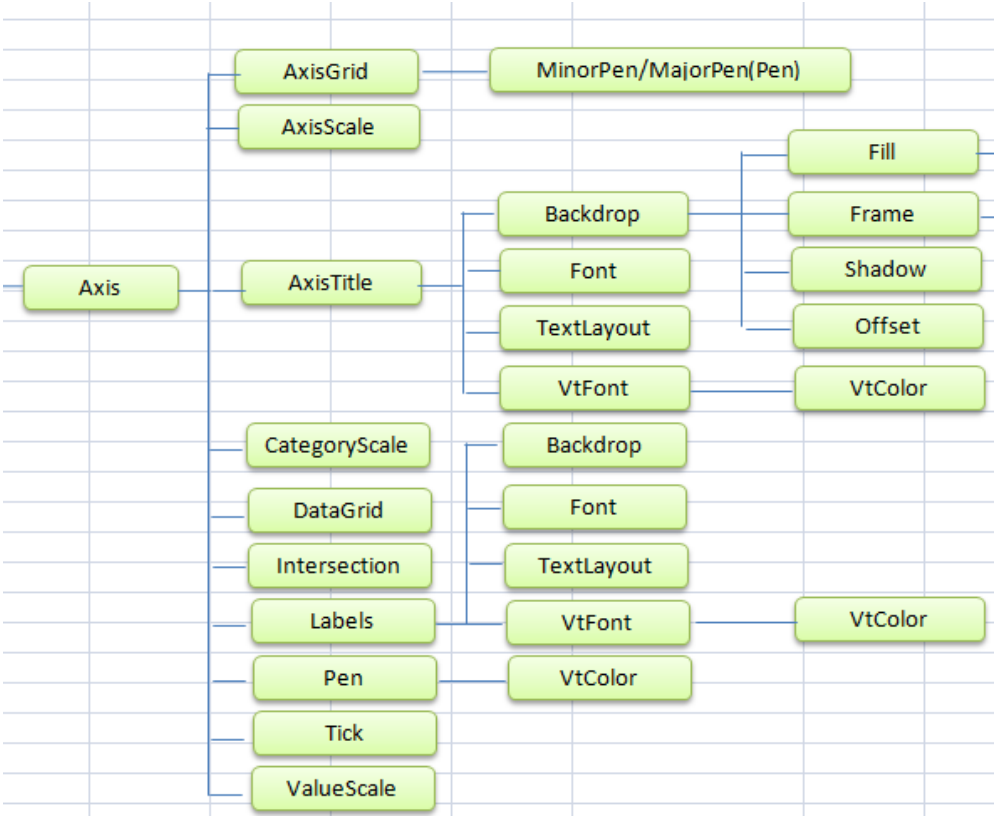


图 2

引用坐标轴的方式如：

MSChart1.Plot.Axis(VtChAxisIdX)

其中 VtChAxisIdX 为 X 坐标轴 ID，坐标轴 ID 共有 4 个常数：

常数	描述
VtChAxisIdX	标识 x 坐标轴。0
VtChAxisIdY	标识 y 坐标轴。1
VtChAxisIdY2	标识次要 y 坐标轴。2
VtChAxisIdZ	标识 z 坐标轴。3

1.3.1 AxisGrid 对象

AxisGrid 对象代表坐标轴的网格线，所有与网格线相关的属性都在这里设置。网格线分主网格线和次网格线。MajorPen 描述主坐标轴网格线的外观，MinorPen 描述次坐标轴网格线的外观，其实这两个属性都是对 Pen 对象的引用。图表上所有的线段或边框的颜色及线型由 Pen 对象（画笔）绘制，而另一个对象 Brush（画刷）决定图表元素的填充类型和颜色。

Pen 对象的风格可为：

常数	描述
VtPenStyleNull	空画笔
VtPenStyleSolid	实线画笔
VtPenStyleDashed	虚线画笔
VtPenStyleDotted	点线画笔
VtPenStyleDashDot	点划线画笔
VtPenStyleDashDotDot	点点划画笔
VtPenStyleDitted	短线（Ditted line）画笔
VtPenStyleDashDit	长短线（Dash-ditted line）画笔
VtPenStyleDashDitDit	短短长线（Dash-dit-dit line）画笔

可以用 MSChart1.Plot.Axis(VtChAxisIdX).AxisGrid.MajorPen.style = VtPenStyleNull 取消显示 X 轴的主网格线，不显示次网格线用 MinorPen，其他坐标轴的设置改变坐标轴 ID 即可。Pen 对象的 VtColor 属性决定画笔的颜色，而其 Width 属性决定画笔的宽度，单位磅。

Brush 对象设置比较复杂，不同风格的画刷，设置不同。画刷风格为：

常数	描述
VtBrushStyleNull	无画笔（背景透出）。
VtBrushStyleSolid	纯色画笔。
VtBrushStylePattern	位图图案画笔。
VtBrushStyleHatched	阴影线画笔。

Solid 风格的画刷（为了区别 Pen 对象（画笔），Brush 对象改叫画刷）只需要设置 Brush 对象的 FillColor 属性，用于指定填充颜色。Null 画刷为透明色，设置什么也看不到，别白费表情了。如果指定画刷为 Pattern 和 Hatched，还需要选择画刷的图案，两种风格画刷的图案是不同的，Pattern 的可选图案为：

常数	描述
VtBrushPattern94percent	94% 图案颜色
VtBrushPattern88percent	88% 图案颜色
VtBrushPattern75percent	75% 图案颜色
VtBrushPattern50percent	50% 图案颜色
VtBrushPattern25percent	25% 图案颜色
VtBrushPatternBoldHorizontal	粗水平线
VtBrushPatternBoldVertical	粗垂直线
VtBrushPatternBoldDownDiagonal	向下的粗对角线
VtBrushPatternBoldUpDiagonal	向上的粗对角线
VtBrushPatternChecks	Checks 图案
VtBrushPatternWeave	Weave 图案

VtBrushPatternHorizontal	水平线
VtBrushPatternVertical	垂直线
VtBrushPatternDownDiagonal	向下对角线
VtBrushPatternUpDiagonal	向上对角线
VtBrushPatternGrid	Grid 图案
VtBrushPatternTrellis	Trellis 图案
VtBrushPatternInvertedTrellis	倒置的 trellis 图案

Hatched 的可选图案为：

常数	描述
VtBrushHatchHorizontal	水平阴影线
VtBrushHatchVertical	垂直阴影线
VtBrushHatchDownDiagonal	向下的对角阴影线
VtBrushHatchUpDiagonal	向上的对角阴影线
VtBrushHatchCross	十字阴影线
VtBrushHatchDiagonalCross	对角十字阴影线

画刷图案用 Brush 对象的 Index 属性指定，绘制图案的颜色由 PatternColor 属性指定。当然，网格线 AxisGrid 是不能设置 Brush 的，在这里提出，是为了区别 Pen 和 Brush。

1.3.2 AxisScale 对象

AxisScale 对象控制如何在坐标轴上绘制图表数值。有 4 个属性：Hide、LogBase、PercentBasis 和 Type 属性。Hide 决定是否隐藏图表的坐标轴，如隐藏 X 坐标轴：

MSChart1.Plot.Axis(VtChAxisIdX).AxisScale.Hide = TRUE

坐标轴（包括刻度标签、刻度标记、轴标题）会被隐藏起来。

LogBase 和 PercentBasis 由 Type 属性确定。Type 属性决定坐标轴的类型，可选值：

常数	描述
VtChScaleTypeLinear	使用线性刻度绘制图表值，值的范围从最小的图表值到最大的图表值。
VtChScaleTypeLogarithmic	对图表值求以 logBase 为底的对数值，以此结果在对数刻度上进行绘制。
VtChScaleTypePercent	根据图表值占图表值域的百分比，在线性刻度上绘制这些图表值。

Type 的默认类型为 Linear，即使用线性刻度绘制图表值。

如果坐标轴类型为 Logarithmic，LogBase 属性设置在对数坐标轴上绘制图表数值时使用的对数底数，缺省的底数为 10，有效的取值范围从 2 到 100。

如果坐标轴类型为 Percent，PercentBasis 属性设置在百分比坐标轴上用来绘制图表数值的百分比类型：

常数	描述
VtChPercentAxisBasisMaxChart	将图表中的最大值作为百分之百，图表上显示的是各个值对应这个值的百分比。

VtChPercentAxisBasisMaxRow	将每一行中的最大值作为百分之百，图表上显示的是各个值对应其所在行的最大值的百分比。
VtChPercentAxisBasisMaxColumn	将每一系列中的最大值作为百分之百，图表上显示的是各个值对应其所在系列的最大值的百分比。
VtChPercentAxisBasisSumChart	将图表中所有值相加所得的总和作为百分之百。图表上显示的是各个值对应总和的百分比。
VtChPercentAxisBasisSumRow	将每一行中所有值相加所得的总和作为百分之百。显示的是各个值对应所在行的总和的百分比。这种方式是百分制堆放图表的基础。
VtChPercentAxisBasisSumColumn	将每一系列中所有值相加所得的总和作为百分之百。图表上显示的是各个值对应其所属系列的总和的百分比。

1.3.3 AxisTitle 对象

AxisTitle 对象设置图表的坐标轴标题，包括文本、字体、颜色和对齐方式，等等，设置并不复杂，不会深入讲解。附件中的示例文件，对 AxisTitle 对象的所有属性都有设置演示。

显示或隐藏坐标轴标题：

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle.Visible = TRUE

设置坐标轴标题字体：

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle.VtFont.Name = “宋体”

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle .VtFont.Size = 16

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle.VtFont.VtColor.Set 255,0,0

设置坐标轴标题文字布局由 TextLayout 对象完成：

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle.TextLayout.WordWrap=TRUE

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle.TextLayout.HorzAlignment

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle.TextLayout. VertAlignment

MSChart1.Plot.Axis(VtChAxisIdX).AxisTitle.TextLayout. Orientation

VtHorizontalAlignment 常数提供文本对齐方式的选项。

常数	描述
VtHorizontalAlignmentLeft	所有文本行向左边界对齐。
VtHorizontalAlignmentRight	所有文本行向右边界对齐。
VtHorizontalAlignmentCenter	所有文本行水平居中。

VtVerticalAlignment 常数提供了以下垂直对齐文本的方法。

常数	描述
VtVerticalAlignmentTop	所有的文本行均与上边距对齐。0
VtVerticalAlignmentBottom	所有的文本行均与下边距对齐。1
VtVerticalAlignmentCenter	所有的文本行均居中垂直对齐。2

VtOrientation 常数提供了以下放置文本的选项。

常数	描述
VtOrientationHorizontal	水平显示文本。0
VtOrientationVertical	文本中每个字母都从上至下显示在其后续字母的正上方。1
VtOrientationUp	将文本旋转为从下往上阅读。2
VtOrientationDown	将文本旋转为从上往下阅读。3

1.3.4 CategoryScale 对象

CategoryScale 对象决定分类坐标轴的刻度。CategoryScale 对象和 ValueScale 对象容易混淆，ValueScale 对象决定坐标轴显示数值时所用的刻度。一条坐标轴既可以是分类坐标轴，也可以是值坐标轴，除了 XY 散点图，X 轴一般都是作为分类坐标轴，Y 轴作为值坐标轴，分类坐标轴的刻度是等分的，XY 散点图中，X 轴也是值坐标轴，刻度就是实际的数值。CategoryScale 对象和 ValueScale 对象非常重要，虽然它们都只有 4-5 个属性，要重点理解。

CategoryScale 对象有 4 个属性：Auto、DivisionsPerLabel、DivisionsPerTick 和 LabelTick 属性。Auto 属性指示是否自动为坐标轴决定刻度，如果不希望控件自动决定，把该值设置为 FALSE。DivisionsPerLabel 属性设置刻度标签之间跳过的部分数，DivisionsPerTick 属性设置主刻度之间跳过的部分数，两者经常设置为相同的值。LabelTick 属性指示分类坐标轴标签是否位于坐标轴刻度标记的中心。下面的示例设置分类坐标轴的比例属性：

```
With MSChart1.Plot.Axis(VtChAxisIdX)
    .CategoryScale.Auto = False '设置为人工缩放。
    .CategoryScale.DivisionsPerLabel = 2 '每两个单位显示标签。
    .CategoryScale.DivisionsPerTick = 2 '每两个单位显示刻度。
    .CategoryScale.LabelTick = True '在刻度标记顶端显示标签。
End With
```

1.3.5 ValueScale 对象

ValueScale 对象决定坐标轴显示数值时所用的刻度。ValueScale 对象有 5 个属性：Auto、MajorDivision 、Maximum 、Minimum 和 MinorDivision 属性。

Auto 属性决定是否使用自动调整刻度的特性来显示坐标轴的数值。MajorDivision 属性设置在坐标轴上显示的主刻度的数目，MSChart 控件会根据 Maximum 和 Minimum 属性设置的值等分坐标轴主刻度数量为 MajorDivision 指定的数字。MinorDivision 属性设置在坐标轴上每一格主刻度显示的次要刻度线的数目。每格主刻度显示的次刻度数量设置为 0，则是不显示，设置为 1 是和主刻度重合，设置为 2 以上才有效果。Maximum 属性设置图表坐标轴的最大值或终止值。Minimum 属性设置图表坐标轴的最小值或起始值。下面示例设置 XY 散点图各坐标轴的 ValueScale 属性，因为 XY 散点图的 X 轴和 Y 轴都是值坐标轴，数据见附件源代码：

```
Private Sub UserForm_Initialize()
Dim arr()
arr = Range("a1:b200")
With MSChart1
    .TitleText = "变化趋势图"
    .Title.VtFont.VtColor.Blue = 255
```

```
    .Plot.Axis(VtChAxisIdX).AxisTitle.Text = "时间（单位：秒）"
    .Plot.SeriesCollection(1).Pen.Width = 20
    .Plot.SeriesCollection(1).Pen.Style = VtPenStyleSolid
    With .Plot.Axis(VtChAxisIdX).ValueScale
        .Auto = False
        .Maximum = 10
        .Minimum = 0
        .MajorDivision = 6
        .MinorDivision = 0
    End With
    .Plot.Axis(VtChAxisIdX).AxisGrid.MajorPen.Style = VtPenStyleDotted
    With .Plot.Axis(VtChAxisIdY).ValueScale
        .Auto = False
        .Maximum = 400
        .Minimum = -800
        .MajorDivision = 12
        .MinorDivision = 0
    End With
    .Plot.Axis(VtChAxisIdY).AxisGrid.MajorPen.Style = VtPenStyleDotted
    .Plot.UniformAxis = False
    .ChartType = VtChChartType2dXY
    .ChartData = arr
End With
End Sub
```

1.3.6 DataGrid 对象

DataGrid 对象表示一个虚拟矩阵，它包含该 MSChart 控件的标签和数据点，言外之意，所有的图表标签和数据值都可以通过它进行修改，主要用于设置二级标签的文本，一般并不会修改数据，因为图表是用来发现规律的，修改数据的方式并不可取。DataGrid 对象由行和列组成，可以在该矩阵中添加和删除行、列以及标签以改变该图表的外观。下面的示例设置一个三维条形图表的图表参数，用随机数据填充图表并为数据网格列分配标签。

```
Option Base 1
Private Sub Command1_Click()
    Dim rowLabelCount, columnLabelCount, rowCount As Integer
    Dim columnCount, labelIndex, Column, Row As Integer
    MSChart1.ChartType = VtChChartType3dBar
    With MSChart1.DataGrid
    '使用方法设置图表参数。
        rowLabelCount = 2’第二级行标签
        columnLabelCount = 2’第二级列标签
        rowCount = 6’行数
        columnCount = 6’列数
        .SetSize RowLabelCount, columnLabelCount, RowCount, ColumnCount
```

```
        .RandomDataFill'随机填充数据。
        labelIndex = 2
        column = 1'然后为第二级分配标签。
        .ColumnLabel(column, labelIndex) = "Product 1"
        column = 4
        .ColumnLabel(column, labelIndex) = "Product 2"
        row = 1
        .RowLabel(row, labelIndex) = "1994"
        row = 4
        .RowLabel(row, labelIndex) = "1995"
    End With
End Sub
```

ColumnCount 属性:返回或设置与图表关联的当前数据网格中的列数。

ColumnLabel 属性:返回或设置与图表关联的网格中数据列的标签。其语法为：
object.ColumnLabel(column, labelIndex) [= text]，参数说明如下：
column 标识特定的数据列，列编号时从 1 开始，由左向右依次编号，任何包含标签的列不作为数据列计数。
labelIndex 标识特定的标签，如果对该列有多个列标签级别，则必须标识其中之一，列标签从 1 开始，从左向右依次编号。
Text 指示列标签文本。

ColumnLabelCount 属性：返回或设置与图表关联的数据网格中的列的标签级别数。设置此属性可以添加或删除数据网格列上的标签级别。从 1 开始、自下而上编号列标签级别，从顶部添加或减少级别。

RowCount 属性 ：返回或设置与图表关联的数据网格每一列的行数。

RowLabel 属性：返回或设置与图表关联的当前数据网格中的特定行标签。其语法为：
object.RowLabel (row, labelIndex) [= text]，参数说明如下：
row 指定行。行从 1 开始、由顶端向底端依次编号。
labelIndex 指定特定的行标签级别。行标签从 1 开始、由顶端向底端依次编号。
Text 指定行标签的文本。

RowLabelCount 属性：返回或设置与图表关联的数据网格行标签的级别数。通过设置该属性来添加或减少数据网格的行标签级别。行标签的级别从 1 开始，从右往左进行编号。标签的添加或减少均在左边进行。

DeleteColumnLabels 方法：从与图表关联的数据网格中删除数据列的标签级。其语法：
object.DeleteColumnLabels (labelIndex, count) ，参数说明如下：
labelIndex 标识要删除的第一级标签编号。列标签从 1 开始，由底端向顶端依次编号。
count 指定要删除的标签级别数。要删除的列从 labelIndex 标识的列开始向上计算。

DeleteColumns 方法：在与图表关联的数据网格中删除数据列以及与它们关联的标签。其语法：
object.DeleteColumns (column, count) ， 参数说明如下：
column 标识特定的数据列。列从 1 开始、由左向右依次编号。
count 指定要删除的列数。

DeleteRowLabels 方法：从与图表关联的数据网格的数据行中删除标签级。其语法：
object.DeleteRowLabels (labelIndex, count) ， 参数说明如下：
labelIndex 指定要删除的第一级标签编号。行标签从 1 开始，由左向右依次编号。
count 指定要删除的标签级别编号。要删除的行标签从 labelIndex 标识的行开始向左计算。

DeleteRows 方法：从与图表关联的数据网格中删除数据行及与它们关联的标签。其语法：
object.DeleteRows (row, count) ， 参数说明如下：
row 标识特定的数据行。行从 1 开始、由顶端向底端依次编号。
count 指定要删除的行数。

GetData 方法：在与图表关联的数据网格中返回特定数据点中的当前存储数值。其语法：
object.GetData (row, column, dataPoint, nullFlag) ， 参数说明如下：
row 标识包含数据点数值的行。
column 标识包含数据点数值的列。
dataPoint 数据点的数值。
nullFlag 指示数据点数值是否为空值。

InsertColumnLabels 方法：在与图表关联的数据网格的数据列中插入标签级。其语法：
object.InsertColumnLabels (labelIndex, count) ， 参数说明如下：
labelIndex 标识要插入的第一级标签编号。列标签级从 1 开始、由底端向顶端依次编号。
count 指定要插入的标签级数。要插入的列数从由 labelIndex 标识的列开始向上计算。

InsertColumns 方法：在与图表关联的数据网格中添加一个或多个数据列。其语法：
object.InsertColumns (column, count) ， 参数说明如下：
column 标识特定的数据列。列从 1 开始、由左向右依次编号。
Count 指定要插入的列数。

InsertRowLabels 方法：在与图表关联的数据网格的数据行中插入标签级。其语法：
object.InsertRowLabels (labelIndex, count) ， 参数说明如下：
labelIndex 标识要插入的第一级标签编号。行标签从 1 开始、由右向左依次编号。
Count 指定要插入的标签级数。插入的行标签由 labelIndex 标识的行起始，向左计算。

InsertRows 方法：在与图表关联的数据网格中添加一个或多个数据行。其语法：
object.InsertRows (row, count)
row 标识特定的数据行。行从 1 开始、由顶端向底端依次编号。
Count 指定要插入的行数。行中都包含空数据直到用数据行填充它们。

SetData 方法：在与图表关联的数据网格中设置数据值。其语法：

object.SetData (row, column, dataPoint, nullFlag) ， 参数说明如下：
row 标识包含数据点数值的行。
column 标识包含数据点数值的列。
dataPoint 数据点的数值。
nullFlag 指示数据点数值是否为空值。

SetSize 方法：调整数据行、数据列数以及与图表关联的数据网格中的列标签级数和行标签级数。此方法可代替 RowCount、ColumnCount、RowLabelCount 和 ColumnLabelCount。其语法：object.SetSize (rowLabelCount, columnLabelCount, dataRowCount, columnLabelCount) ， 参数说明如下：
rowLabelCount 返回或设置数据网格中的行标签级数。
columnLabelCount 返回或设置数据网格中的列标签级数。
dataRowCount 返回或设置数据网格上的数据行数。
columnLabelCount 返回或设置数据网格上的数据列数。

关于多级标签和数据网格的设置，附件中有详细的演示代码。

1.3.7 Intersection 对象

Intersection 对象用于设置两坐标轴相交点，有 5 个属性：Auto、AxisId、Index、LabelsInsidePlot 和 Point 属性。

Auto 属性：返回或设置一个数值，它决定 Intersection 对象是否用 Point 属性的数值来放置坐标轴。其语法：object.Auto [= boolean]，参数说明如下：
boolean 的设置值为：
True （缺省）将坐标轴放置在标准位置。
False 将相交的坐标轴放置在 Point 指定的数值处。

LabelsInsidePlot 属性：返回或设置一个数值，它决定是将坐标轴标签保留在正常位置，还是将它们与坐标轴一起移动到新的相交点。其语法：
object.LabelsInsidePlot [= boolean] ， 参数说明如下：
boolean 的设置值为：
True （缺省）坐标轴标签保留在正常位置。
False 图象内部的标签移动到新的相交点处。

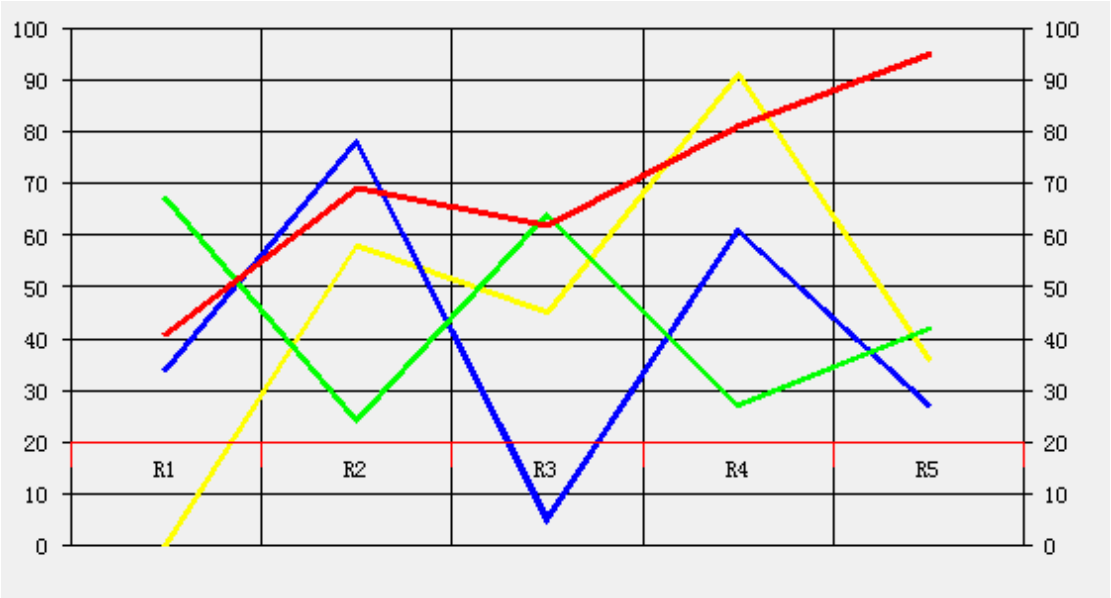
Point 属性：返回或设置当前坐标轴与另一坐标轴的相交点。其语法：
object.Point [= point] ， 参数说明如下：
point 指示当前坐标轴与一个轴的相交点。

举例说明，特别注意的是 LabelsInsidePlot 属性的影响：

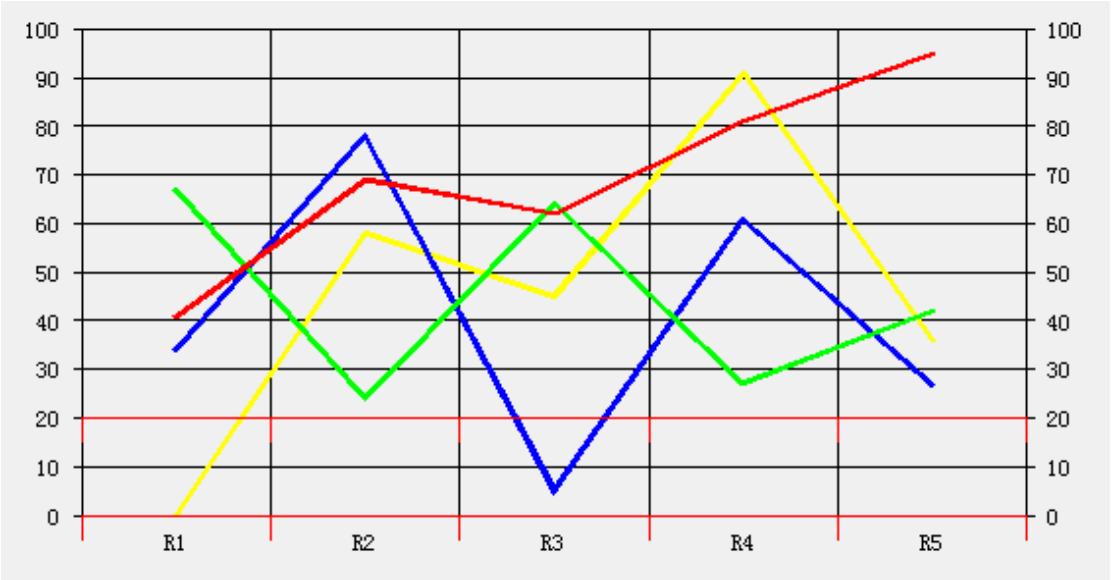
```
Private Sub UserForm_Initialize()
    With MSChart1
        .ChartType = VtChChartType2dLine
```

```
        With .Plot.Axis(VtChAxisIdX).Pen
            .Width = 4
            .VtColor.Red = 255
        End With
        With .Plot.Axis(VtChAxisIdX).Intersection
            '设置相交属性。
            .Auto = False     '设置手工定位。
            .Point = 20       '将与 Y 坐标轴上的值 20 相交。
            .LabelsInsidePlot = True '将标签与坐标轴一起显示
        End With
    End With
End Sub
```

其显示结果为下图，红色的细线为 X 坐标轴，相较于 Y 轴的 20：



现在把 LabelsInsidePlot 设置 FALSE，其结果为下图，可见标签移动了：



1.3.8 Label 对象

Labels 集合中的项目，它描述特定的图表坐标轴标签。坐标轴标签设置文本布局很简单，和其他需要设置文本布局的对象方法一样，没什么好讲的，直接放在附件演示代码中了。不过 Labels 集合只能为 Labels(1)，代表指定轴的所有标签，不可以单独设置一个标签的格式，唯一能设置的就是其文本，可用数据网格对象的 DataGrid.RowLabel 和 DataGrid.ColumnLabel 进行设置。Label 对象唯一要讲的就是 Format 属性，其返回或设置用来定义显示坐标轴标签格式的字符，其格式文本跟 VB 函数 Format 使用的规则一样，如设置 Y 轴标签为两位小数：

```
MSChart1.Plot.Axis(VtChAxisIdY).Labels(1).Format = "0.00"
```

1.3.9 Tick 对象

Tick 对象代表沿图表坐标轴标识图表项的标记，该对象是指坐标轴上的刻度标记，刻度不是网格线，要注意区别。该对象只有两个属性，我也想不出有修改它们的必要性，但为了讲解得全面点，还是列举吧。

Length 返回或设置坐标轴 tick 标记的长度，以点为单位。

Style 用来描述坐标轴刻度标记位置，其可选值为：

常数	描述
VtChAxisTickStyleNone	坐标轴上不显示刻度标记。
VtChAxisTickStyleCenter	刻度标记位于坐标轴的中心。
VtChAxisTickStyleInside	刻度标记显示在坐标轴内侧。
VtChAxisTickStyleOutside	刻度标记显示在坐标轴外侧。

要设置 X 坐标轴刻度的长度和位置，可以用如下代码：

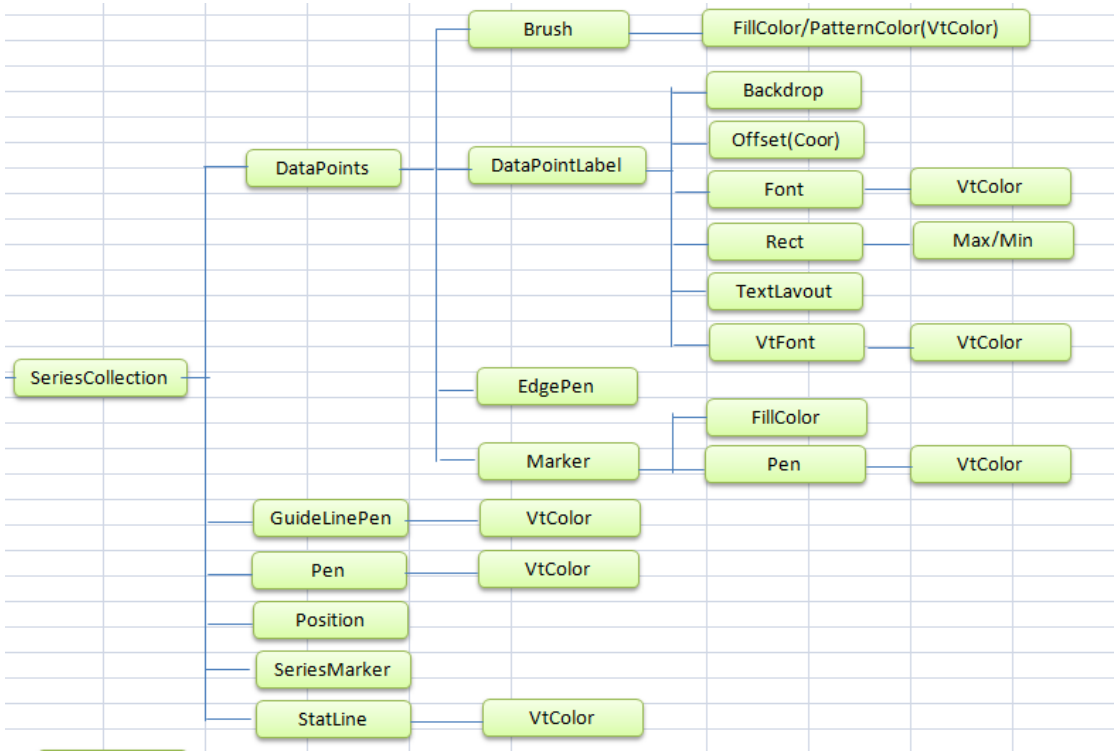
```
With MSChart1.Plot.Axis(VtChAxisIdX).Tick
    .Length = 500
    .Style = VtChAxisTickStyleOutside
End With
```

1.4 SeriesCollection 集合与 Series 对象

Series 对象（系列）是 SeriesCollection 集合中的项，代表图表中的一组数据点。Series 对象和 Axis 对象是图表中最常使用的对象，是图表的基础，要重点掌握的。其对象模型如下图。这里先介绍其属性。

DataPoints 属性：返回对 DataPoint 集合（数据点）的引用，它描述图表系列中的数据点。

GuidelinePen 属性：返回对 Pen 对象的引用，该对象描述用于显示引导线的线型及颜色。设置该属性会自动将 ShowGuideLines 属性设为 True。



下面的示例为二维 xy 图表系列设置画笔属性。其中 GuideLinePen 属性返回对 Pen 对象的引用。

```
Private Sub Command1_Click()
    '为二维的 XY 图表的系列 1 设置引导线。
    MSChart1.ChartType = VtChChartType2dXY
    MSChart1.Plot.SeriesCollection.Item(1) _
    .ShowGuideLine(VtChAxisIdX) = True
    With _
    MSChart1.Plot.SeriesCollection.Item(1).GuideLinePen
        '设置画笔属性。
        .VtColor.Set 255, 0, 0
        .Width = 4
        .Style = VtPenStyleDashDot
        .Join = VtPenJoinRound
        .Cap = VtPenCapRound
    End With
End Sub
```

LegendText 属性：返回或设置在图表的图例中标识系列的文本。

Pen 属性：返回或设置对 Pen 对象的引用，该对象描述系列线的颜色及线型。

Position 属性 ：返回对 SeriesPosition 对象的引用，该对象描述与其它图表系列互相关联的系列的位置。在堆叠图表的时候，需要设置哪个系列在图层的上面，需要使用到这个属性。其下还有 Hide 属性可用于隐藏一个系列。

SecondaryAxis 属性：返回或设置值，这个值决定是否在次坐标轴绘制该系列。非常重要的属性。

SeriesMarker 属性：返回对 SeriesMarker 对象的引用，该对象描述标识图表系列中的所有数据点的标记。SeriesMarker 对象的 Auto 属性返回或设置一个值，该值决定 SeriesMarker 对象是否用下一个可用标记来标记系列中的所有数据点，如果要使用自定义的标记类型，需要把 Auto 属性设置为 FALSE，SeriesMarker 对象的 Show 属性返回或设置一个值，该值决定是否在图表上显示系列的标记。

SeriesType 属性：返回或设置用于显示当前系列的类型。非常重要的属性。当使用组合图时，使用该属性修改每个系列的图表类型。可选的数值为：

常数	描述
VtChSeriesType3dBar	三维直方图
VtChSeriesType2dBar	二维直方图
VtChSeriesType3dLine	三维折线图
VtChSeriesType2dLine	二维折线图
VtChSeriesType3dArea	三维面积图
VtChSeriesType2dArea	二维面积图
VtChSeriesType3dStep	三维阶梯图
VtChSeriesType2dStep	二维阶梯图
VtChSeriesType2dXY	XY 散点图
VtChSeriesType2dPie	二维饼图

StatLine 属性：返回对 StatLine 对象的引用，该对象描述如何在图表中显示统计线。后面会单独讲到它。

1.4.1 DataPoint 对象

DataPoint 对象是 DataPoints 集合中的成员，它描述图表中单个数据点的属性。DataPoints 集合必须使用-1 的索引，如：MSChart1.Plot.SeriesCollection(1).DataPoints(-1)，因为不可以对单独的数据点格式进行设置，-1 代表所有数据点。先说明其属性。

Brush 属性：返回对 Brush 对象的引用，它描述用于显示数据点的填充类型。Brush 对象在之前已经讲过，这里不再重复。假如要修改系列 1 所有数据点的填充颜色和风格，可用：

```
With MSChart1.Plot.SeriesCollection(1).DataPoints(-1)
    .Brush.Style = VtBrushStyleSolid
    .Brush.FillColor.Set 0, 255, 255
    .Marker.Visible = True
End With
```

DataPointLabel 属性：返回对 DataPointLabel 对象的引用，它描述个别图表数据点上的标签。这个在后边单独讲。

EdgePen 属性：返回 Pen 对象，此对象用于绘制图表中数据点的边界。Pen 对象已讲过了。

Marker 属性：返回对 Marker 对象的引用，该对象描述用来标识图表中的数据点的图标。可用 Marker 对象的 FillColor 属性对标记进行填充颜色。Marker 对象的 Pen 属性用于绘制数据点标记的边框颜色及线型，标记大小由 Size 属性指定。数据点标记的 Style（类型）为：

常数	描述
VtMarkerStyleNull	标记为空
VtMarkerStyleDash	标记为短线
VtMarkerStylePlus	标记为加号
VtMarkerStyleX	标记为 X 符号
VtMarkerStyleStar	标记为星号
VtMarkerStyleCircle	标记为圆
VtMarkerStyleSquare	标记为正方形
VtMarkerStyleDiamond	标记为菱形
VtMarkerStyleUpTriangle	标记为上三角形
VtMarkerStyleDownTriangle	标记为下三角形
VtMarkerStyleFilledCircle	标记为填充的圆
VtMarkerStyleFilledSquare	标记为填充的正方形
VtMarkerStyleFilledDiamond	标记为填充的菱形
VtMarkerStyleFilledUpTriangle	标记为填充的上三角形
VtMarkerStyleFilledDownTriangle	标记为填充的下三角形
VtMarkerStyle3dBall	标记为立体球

1.4.2 DataPointLabel 对象

DataPointLabel 对象代表图表中数据点的标签。其重要属性如下：

Component 属性：返回或设置用来标识数据点的标签类型。

常数	描述
VtChLabelComponentValue	数据点数值出现在标签中。散点图表、极坐标图表和气泡图表中的数据点实际包含两个或三个数值。这些图表的缺省标签以标准格式显示所有数值。但是也可以自定义这种格式，以突出显示单个的数据值。
VtChLabelComponentPercent	在标签中将数据点数值显示成该数据点数值占系列总数值的百分比。
VtChLabelComponentSeriesName	用系列名称来标注数据点。该名称源于与数据网格列关联的标签。
VtChLabelComponentPointName	用数据点名称来标注数据点。

LineStyle 属性：
返回或设置用来连接图表中数据点和标签的线段类型。

常数	描述
VtChLabelLineStyleNone	标签和系列之间没有线段连接。
VtChLabelLineStyleStraight	标签和系列之间用直线连接。
VtChLabelLineStyleBent	标签和直线之间用弯曲线连接。

LocationType 属性：返回或设置显示数据点标签的位置。

常数	描述
VtChLabelLocationTypeNone	不显示标签。
VtChLabelLocationTypeAbovePoint	标签显示在数据点上方。
VtChLabelLocationTypeBelowPoint	标签显示在数据点下方。
VtChLabelLocationTypeCenter	标签显示在数据点正中。
VtChLabelLocationTypeBase	标签沿分类坐标轴显示在图表底部，并位于数据点的正下方。
VtChLabelLocationTypeInside	标签显示在饼图的饼块内部。
VtChLabelLocationTypeOutside	标签显示在饼图的饼块外部。
VtChLabelLocationTypeLeft	标签显示在数据点左侧。
VtChLabelLocationTypeRight	标签显示在数据点右侧。

PercentFormat 属性：返回或设置一个字符串，它描述用来显示百分比标签的格式。还有一个 ValueFormat 属性，对值的格式设置。挺繁琐的。

数据点标签的文本布局跟其他对象一样，略……。

1.4.3 StatLine 对象

StatLine 对象描述如何在图表中显示统计线。StatLine 对象只有 4 个属性:
Width 属性指定统计线的宽度

VtColor 属性指定统计线的颜色。

Flag 属性设置显示系列的是哪条统计线。

常数	描述
VtChStatsMinimum	显示该系列的最小值。
VtChStatsMaximum	显示该系列的最大值。
VtChStatsMean	显示该系列的数据值的数学期望。
VtChStatsStddev	显示该系列的数据值的标准方差。
VtChStatsRegression	显示系列中数据值所确定的趋势线。

如果需要显示的统计线不止一条，则使用 OR 操作符来组合这些常量。

Style 属性：返回或设置用于显示统计线的线型。其语法：
object.Style (type)[= style]，参数说明如下：

type 是用于描述统计线类型的常数，即 Flag 属性指定的值。
style 描述统计线样式的 VtPenStyle 常数。

常数	描述
VtPenStyleNull	空画笔
VtPenStyleSolid	实线画笔
VtPenStyleDashed	虚线画笔
VtPenStyleDotted	点线画笔
VtPenStyleDashDot	点划线画笔
VtPenStyleDashDotDot	点点划画笔
VtPenStyleDitted	短线画笔
VtPenStyleDashDit	长短线画笔
VtPenStyleDashDitDit	短短长线画笔

下面的示例为图表的统计线设置颜色及画笔的参数：

```
'显示系列 2 的所有统计线。
MSChart1.ChartType = VtChChartType2dLine
With MSChart1.plot.SeriesCollection(2).StatLine
    .VtColor.Set 255, 0, 0
    .Flag = vtChStatsMinimum Or VtChStatsMaximum _
    Or VtChStatsMean Or VtChStatsStddev Or _
    VtChStatsRegression
    .Style(vtChStatsMinimum) = VtPenStyleDotted
    .Width = 4
End With
```

1.4.4 Backdrop 对象

Backdrop 对象表示图表元素后面的阴影或图案。本来不打算讲这个对象的，因为我不觉得它有什么用，设置它会把图表搞得眼花缭乱，我追求的是图表简洁、简单，但是很多对象都包含这个对象，我就稍微说一下吧。Backdrop 对象有 3 个属性：Fill、Frame 和 Shadow 属性。

Fill 属性：返回对 Fill 对象的引用，它描述图表对象背景的类型和外观。Fill 对象有 2 个属性，Brush 和 Style 属性，Brush 就是前面说过的画刷对象， Style 属性提供指定用于绘制背景的填充类型的选项，可用选项为：

常数	描述
VtFillStyleNull	无填充（背景透出）。
VtFillStyleBrush	纯色或图案填充。

Frame 属性：返回对 Frame 对象的引用，它描述图表元素周围框架的外观。Frame 对象有 4 个属性：FrameColor、SpaceColor 、Style 和 Width 属性。其中，FrameColor 属性返回对 VtColor 对象的引用，该对象指定环绕图表元素的单框所用的颜色。SpaceColor 属性返回对 VtColor 对象的引用，该对象指定填充环绕图表元素的双框间的空白所用的颜色。Width 属性返回或设置以

磅为单位的框架线宽。Style 属性描述框架的类型，可选值为：

常数	描述
VtFrameStyleNull	无框架。
VtFrameStyleSingleLine	背景周围围绕单线。
VtFrameStyleDoubleLine	背景周围围绕两条等宽线。
VtFrameStyleThickInner	背景周围围绕一条粗的内线和一条细的外线。
VtFrameStyleThickOuter	背景周围围绕一条细的内线和一条粗的外线。

下面的示例在图表背景上设置蓝色的双线框架。

```
With MSChart1.backdrop.Frame
    .Style = VtFrameStyleDoubleLine
    .Width = 2
    .FrameColor.Set 0, 0, 255     '蓝色框架。
    .SpaceColor.Set 255, 0, 0     '红色间隔。
End With
```

Style 属性 (MSChart)

Shadow 属性：返回对 Shadow 对象的引用，该对象描述图表元素的阴影样式。Shadow 对象有 3 个属性：Brush、Offset 和 Style 属性。Brush 属性返回对 Brush 对象的引用，它描述用于显示图表元素的填充类型。Offset 属性返回或设置，图表元素与它的缺省位置偏移的距离。Style 属性描述阴影类型，可选值为：

常数	描述
VtShadowStyleNull	无阴影。
VtShadowStyleDrop	点阴影。

下面的示例给图表的背景设置阴影。

```
With MSChart1.backdrop.Shadow
    .Style = VtShadowStyleDrop
    .Offset.x = 10
    .Offset.y = 10
End With
```

再写一个设置 backdrop 对象 3 个属性的代码：

```
Private Sub UserForm_Initialize()
    With MSChart1
        .ChartType = VtChChartType2dBar
        .RandomFill = True
        With .Backdrop.Fill
            .style = VtFillStyleBrush
            .Brush.FillColor.Red = 255
            .Brush.style = VtBrushStylePattern
            .Brush.Index = VtBrushPattern50Percent
            .Brush.PatternColor.Blue = 255
        End With
        With .Backdrop.Frame
```

```
            .style = VtFrameStyleDoubleLine
            .FrameColor.Red = 255
            .SpaceColor.Blue = 255
            .Width = 2
        End With
        With .Backdrop.Shadow
            .style = VtShadowStyleDrop
            .Brush.style = VtBrushStyleSolid
        End With
    End With
End Sub
```

1.5 总结

MSChart 控件的对象非常多，对图表元素的操控也会很细致，如果本教程对所有图表元素的操作都一一举例，教程会变得非常庞大。因此，我把对图表元素的操作全部写在代码里，大家可以在窗体上点击组合框的选项就可以看到操作结果，既可以对图表元素有个直观的认识，也可在需要的时候直接复制相关代码。为了实现在窗体上直观演示，我多写了 10 倍的代码量，这个方法看来不是很明智，至少对我而言是这样。这里提一下，获取系统字体的方法。论坛上的大神用如下简单的代码就获取了系统的字体：

```
With Application.CommandBars.FindControl(ID:=1728)
  For i = 1 To .ListCount
    Cells(i, 1) = .List(i)
  Next
End With
```

不过这个方法不通用，不同版本的 EXCEL 的 ID 可能不同，有些版本会什么也获取不到。于是我查阅了微软的资料，用 API 实现字体的获取，这样通用性应该好些，顺便把安装字体和卸载字体的代码一并提供，这样谁想设计一个小软件的时候，可以直接复制代码，也应该不错的。代码可供参考：

```
'获取系统字体清单
Private Declare Function EnumFontFamiliesEx Lib "gdi32" Alias "EnumFontFamiliesExA" (ByVal hDC As Long, lpLogFont As LOGFONT, ByVal lpEnumFontProc As Long, ByVal lParam As Long, ByVal dw As Long) As Long
Private Declare Function GetDC Lib "user32" (ByVal hWnd As Long) As Long
Private Declare Function ReleaseDC Lib "user32" (ByVal hWnd As Long, ByVal hDC As Long) As Long
Private Const LF_FACESIZE = 32
Private Const LF_FULLFACESIZE = 64
Private Const ANSI_CHARSET = 0

Private Type LOGFONT
        lfHeight As Long
        lfWidth As Long
```

```
        lfEscapement As Long
        lfOrientation As Long
        lfWeight As Long
        lfItalic As Byte
        lfUnderline As Byte
        lfStrikeOut As Byte
        lfCharSet As Byte
        lfOutPrecision As Byte
        lfClipPrecision As Byte
        lfQuality As Byte
        lfPitchAndFamily As Byte
        lfFaceName(1 To LF_FACESIZE) As Byte
End Type

Private Type ENUMLOGFONTEX
    elfLogFont As LOGFONT
    elfFullName(1 To LF_FULLFACESIZE) As Byte
    elfStyle(1 To LF_FACESIZE) As Byte
    elfScript(1 To LF_FACESIZE) As Byte
End Type

Public frmname As String
Public cmbname As String
Public frm As Object
'
'参考资料：
'https://docs.microsoft.com/zh-cn/windows/desktop/api/wingdi/nf-wingdi-enumfontfamiliesexa
'ivccav 2019/1/15
Private Function EnumFontFamExProc(ByRef lpelfe As ENUMLOGFONTEX, ByVal lpntme As
Long, ByVal FontType As Long, ByVal lParam As Long) As Long
    Dim fn$, pos&
    fn = StrConv(lpelfe.elfLogFont.lfFaceName, vbUnicode)
    pos = InStr(fn, Chr$(0))
    If pos Then fn = Left$(fn, pos - 1)
    With frm.Controls(cmbname)
        .AddItem fn
    End With
    EnumFontFamExProc = 1
End Function

Public Sub GetAllFonts()
    Dim lf As LOGFONT
    Dim hDC&, f As Object
    Dim bln As Boolean
```

```
    For Each f In UserForms
        If f.Name = frmname Then bln = True: Exit For
    Next
    If bln Then Set frm = f Else Exit Sub
    lf.lfCharSet = ANSI_CHARSET
    hDC = GetDC(0&)
    EnumFontFamiliesEx hDC, lf, AddressOf EnumFontFamExProc, 0&, 0&
    ReleaseDC 0&, hDC
End Sub

Private Declare Function AddFontResource Lib "gdi32" Alias "AddFontResourceA" (ByVal lpFileName As String) As Long
Private Declare Function RemoveFontResource Lib "gdi32" Alias "RemoveFontResourceA" (ByVal lpFileName As String) As Long
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hWnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any) As Long
Private Const HWND_BROADCAST = &HFFFF&
Private Const WM_FONTCHANGE = &H1D

'参考资料：
'https://docs.microsoft.com/zh-cn/windows/desktop/api/wingdi/nf-wingdi-addfontresourcea
"ivccav 2019/1/15
Public Sub AddNewFont() '安装字体
    Dim lRet As Long, FontPath As String
    With Application.FileDialog(msoFileDialogFilePicker)
        .AllowMultiSelect = False
        .InitialFileName = "C:\Windows\Fonts\"
        .Filters.Clear
        .Filters.Add "字体文件", "*.ttf;*.ttc;*.otf"
        .Title = "请选择要安装的字体文件"
        If .Show = True Then
            FontPath = .SelectedItems.Item(1)
            lRet = AddFontResource(FontPath)
            If lRet = 0 Then
                MsgBox "添加字体失败!", vbCritical
                Exit Sub
            Else
                SendMessage HWND_BROADCAST, WM_FONTCHANGE, 0, 0
                MsgBox "添加字体成功", vbInformation
            End If
        End If
    End With
End Sub
```

```
Public Sub RemoveFont() '卸载字体
    Dim lRet As Long, FontPath As String
    With Application.FileDialog(msoFileDialogFilePicker)
        .AllowMultiSelect = False
        .InitialFileName = "C:\Windows\Fonts\"
        .Filters.Clear
        .Filters.Add "字体文件", "*.ttf;*.ttc;*.otf"
        .Title = "请选择要卸载的字体文件"
        If .Show = True Then
            FontPath = .SelectedItems.Item(1)
            lRet = RemoveFontResource(FontPath)
            If lRet = 0 Then
                MsgBox "卸载字体失败!", vbCritical
                Exit Sub
            Else
                SendMessage HWND_BROADCAST, WM_FONTCHANGE, 0, 0
                MsgBox "卸载字体成功", vbInformation
            End If
        End If
    End With
End Sub
```

2. 图表实践

2.1 堆叠图表

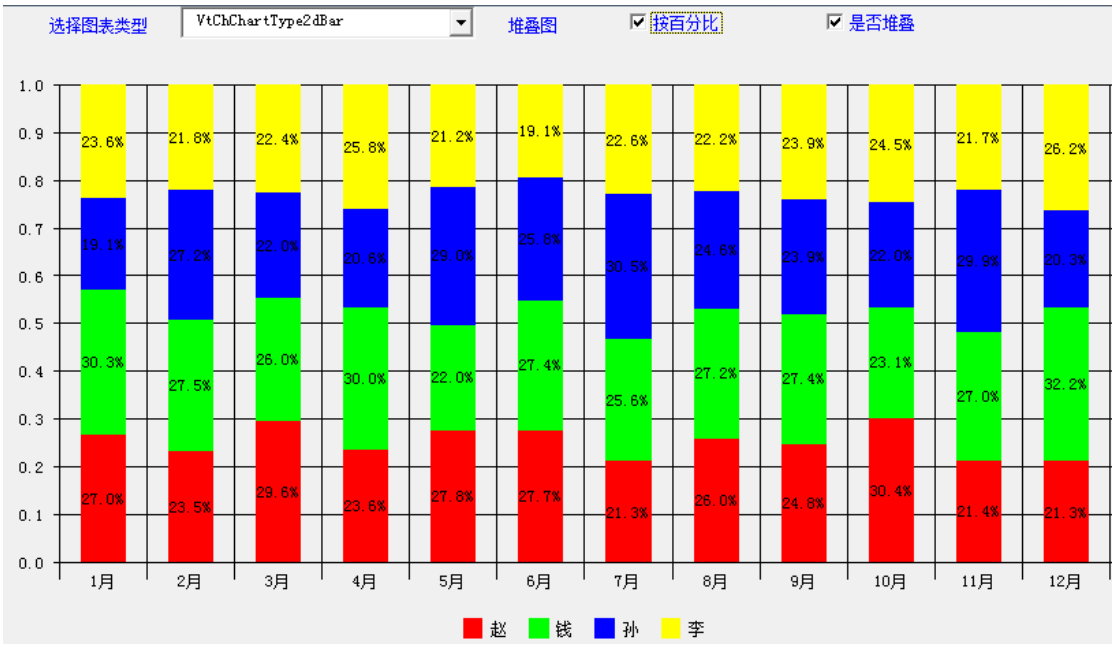
图表的堆叠通过 Stacking 属性设置。堆叠图表可用于比较同一个点不同系列的比较。当然也可以使用饼图实现。本节不仅讲了堆叠柱状图的使用，还使用了百分比坐标轴类型。使用如下数据实现 4 个人相同月份的数据比较：

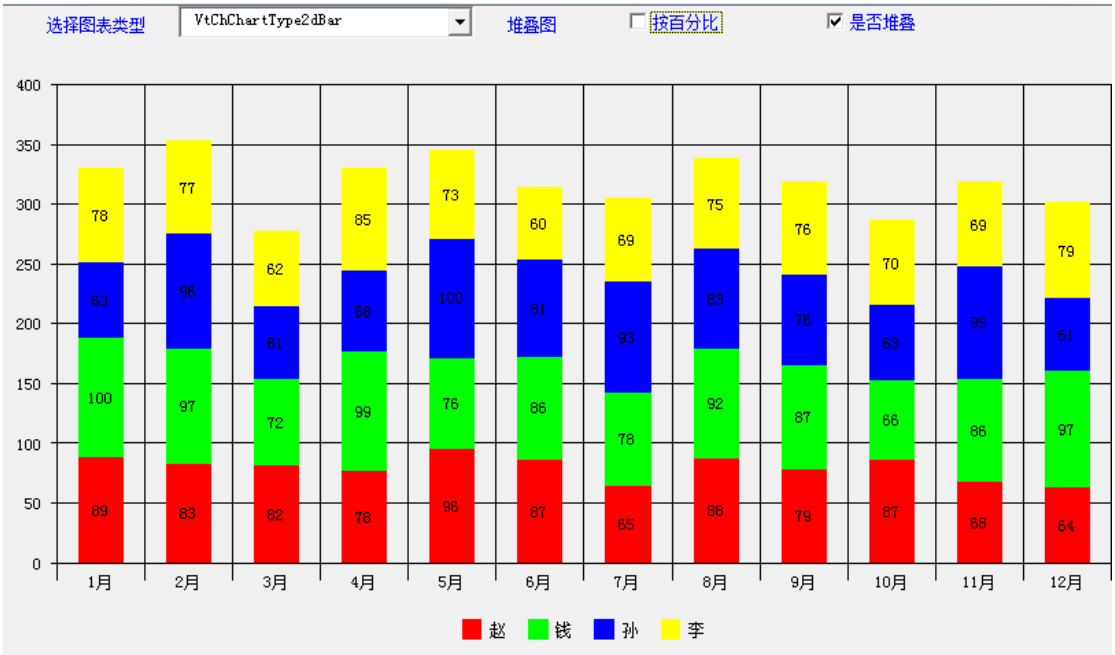
	A	B	C	D	E
1		赵	钱	孙	李
2	1月	89	100	63	78
3	2月	83	97	96	77
4	3月	82	72	61	62
5	4月	78	99	68	85
6	5月	96	76	100	73
7	6月	87	86	81	60
8	7月	65	78	93	69
9	8月	88	92	83	75
10	9月	79	87	76	76
11	10月	87	66	63	70
12	11月	68	86	95	69
13	12月	64	97	61	79

全部代码如下（附件中窗体 4）注意看用百分比和不用百分比的区别：

```
Private Sub CheckBox1_Click()
With MSChart1
    If CheckBox1 Then
        .Plot.Axis(VtChAxisIdY).AxisScale.Type = VtChScaleTypePercent
        .Plot.Axis(VtChAxisIdY).AxisScale.PercentBasis = VtChPercentAxisBasisSumRow
        For i = 1 To .Plot.SeriesCollection.Count
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.LocationType =
VtChLabelLocationTypeCenter
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.Component =
VtChLabelComponentPercent
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.PercentFormat = "0.0%"
        Next
    Else
        .Plot.Axis(VtChAxisIdY).AxisScale.Type = VtChScaleTypeLinear
        For i = 1 To .Plot.SeriesCollection.Count
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.LocationType =
VtChLabelLocationTypeCenter
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.Component =
VtChLabelComponentValue
        Next
    End If
End With
End Sub

Private Sub CheckBox2_Click()
    MSChart1.Stacking = CheckBox2
End Sub

Private Sub UserForm_Initialize()
Dim arr(), i%
arr = Range("a1:e13")
With MSChart1
    .ChartData = arr
    .ChartType = VtChChartType3dArea
    .Stacking = True
    .Legend.Location.Visible = True
    .Legend.Location.LocationType = VtChLocationTypeBottom
    For i = 1 To .Plot.SeriesCollection.Count
        .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.LocationType =
VtChLabelLocationTypeCenter
    Next
End With
```

```
With ComboBox1
    .AddItem "VtChChartType3dBar" '0 3D 条形图
    .AddItem "VtChChartType2dBar" '1 2D 条形图
    .AddItem "VtChChartType3dLine" '2 3D 折线图
    .AddItem "VtChChartType2dLine" '3 2D 折线图
    .AddItem "VtChChartType3dArea" '4 3D 面积图
    .AddItem "VtChChartType2dArea" '5 2D 面积图
    .AddItem "VtChChartType3dStep" '6 3D 阶梯图
    .AddItem "VtChChartType2dStep" '7 2D 阶梯图
    .AddItem "VtChChartType3dCombination" '8 3D 组合图
    .AddItem "VtChChartType2dCombination" '9 2D 组合图
    .AddItem "VtChChartType2dPie" '14 2D 饼图
    .AddItem "VtChChartType2dXY" '16 2D XY 散点图
    .ListIndex = 1
    .style = fmStyleDropDownList
End With
    CheckBox2 = True
    Me.Caption = "认识堆叠图"
End Sub

Private Sub ComboBox1_Change()
    Dim idx%
    idx = ComboBox1.ListIndex
    If idx > 9 Then idx = 14 + IIf(idx = 10, 0, 2)
    MSChart1.ChartType = idx
End Sub
```





2.2 切片图表

其实是没有切片图表类型的，这个说法是 EXCEL 里面的，本质就是按不同的条件进行查询，把数据归类，然后用图表展示。本节举 2 个例子，第一个例子使用饼图的图表类型，第二个例子使用了统计线（使用的统计线类型是趋势线）。例 1 把如下数据按年份切片，统计每个人不同项目的绩效：

	A	B	C	D	E	F
1	年数	姓名	项目A	项目B	项目C	项目D
2	2018年	李	100	102	200	144
3	2010年	李	101	126	185	223
4	2011年	孙	102	173	160	107
5	2013年	孙	102	184	100	167
6	2018年	孙	104	118	160	174
7	2013年	李	105	194	110	110
8	2014年	孙	105	164	195	160
9	2010年	孙	107	132	195	200
10	2013年	李	107	114	177	191
11	2015年	钱	107	115	202	191
12	2016年	李	110	157	164	112
13	2016年	孙	110	215	248	237
14	2016年	孙	112	154	188	118
15	2010年	钱	112	124	185	194
16	2011年	钱	113	107	180	107
17	2012年	赵	114	128	187	174
18	2014年	孙	114	138	173	132
19	2014年	钱	116	117	116	131
20	2011年	李	116	168	200	108
21	2010年	钱	118	186	174	136

代码如下（附件中窗体 11）：

```
Dim cnn As Object

Private Sub CheckBox1_Click()
With MSChart1
    If CheckBox1 Then
        .Plot.Axis(VtChAxisIdY).AxisScale.PercentBasis = VtChPercentAxisBasisSumRow
        .Plot.Axis(VtChAxisIdY).AxisScale.Type = VtChScaleTypePercent
        .Plot.Weighting.Basis = VtChPieWeightBasisNone '所有饼图一样大小
        For i = 1 To .Plot.SeriesCollection.Count '无法设置单个数据点属性，索引必须等于-1，代表所有数据点
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.LocationType = VtChLabelLocationTypeCenter
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.Component = VtChLabelComponentPercent
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.PercentFormat = "0.0%"
        Next
    Else
        .Plot.Axis(VtChAxisIdY).AxisScale.Type = VtChScaleTypeLinear
        .Plot.Weighting.Basis = VtChPieWeightBasisTotal
        For i = 1 To .Plot.SeriesCollection.Count
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.LocationType = VtChLabelLocationTypeCenter
            .Plot.SeriesCollection(i).DataPoints(-1).DataPointLabel.Component = VtChLabelComponentValue
        Next
    End If
End With
End Sub

Private Sub ComboBox1_Change()
    Set rst = CreateObject("adodb.recordset")
    Sql = "select 姓名,sum(项目 A) as 项目 A,sum(项目 B) as 项目 B,sum(项目 C) as 项目 C,sum(项目 D) as 项目 D " & _
            "from [切片图表$] where 年数='" & ComboBox1 & "' group by 姓名"
    rst.Open Sql, cnn, 1, 3
    If rst.RecordCount = 0 Then Exit Sub
    With MSChart1
        .ChartData = Application.Transpose(rst.getrows)
        .ChartType = VtChChartType2dPie
        For i = 1 To .columnCount '修改系列名称，不能用 ChartData 修改
            .Column = i
            .ColumnLabel = rst.fields(i).Name
```

```
            Next
            .AllowSelections = False '禁止选择图表元素
            .ShowLegend = True
            .Legend.Location.LocationType = VtChLocationTypeBottom
        End With
        rst.Close
        Set rst = Nothing
End Sub

Private Sub UserForm_Initialize()
        Set cnn = CreateObject("adodb.connection")
        Set rst = CreateObject("adodb.recordset") '连接数据库
        cnn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Extended Properties='Excel
12.0;HDR=YES';Data Source=" & ThisWorkbook.FullName
        Sql = "select distinct 年数 from [切片图表$] order by 年数 desc"
        rst.Open Sql, cnn, 1, 3
        If rst.RecordCount = 0 Then Exit Sub
        With ComboBox1
            .Column = rst.getrows
            .ListIndex = 0
            .style = fmStyleDropDownList
        End With
        CheckBox1 = True
        Me.Caption = "切片图"
        rst.Close
        Set rst = Nothing
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
        cnn.Close
        Set cnn = Nothing
End Sub
```

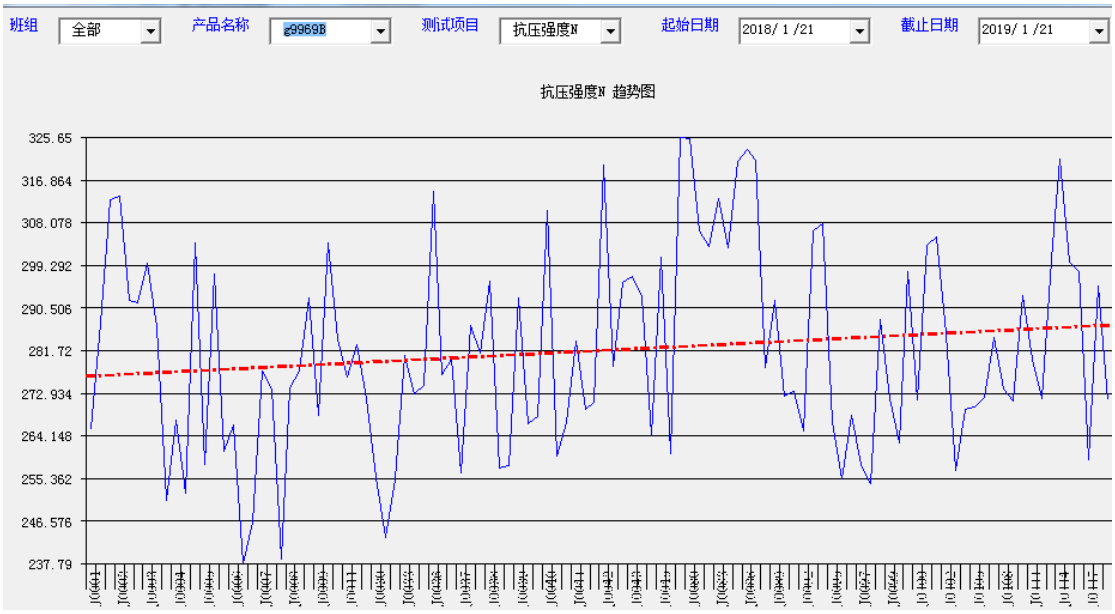
例 2 的数据来源于网友的提问，需要切片的数据很多，还好可用 ADO，不然代码都得写晕。工作表函数 Application.Index 用于获取数组中的一整列，而工作表函数 Application.Max 和 Application.Min 可获取一维数组中的最大值和最小值，善用工作表函数可省略不少代码。最重要的当然是 ValueScale 对象的使用，要理解了，才能作好图表。示例为附件窗体 17.

```
Dim cnn As Object

Private Sub ComboBox1_Change() '班组名称
        Call Query
End Sub
```

```
Private Sub ComboBox2_Change() '产品名称
    Call Query
End Sub

Private Sub ComboBox3_Change() '测试项目
    Call Query
End Sub

Private Sub DTPicker1_Change() '日期范围
    Call Query
End Sub

Private Sub DTPicker2_Change() '日期范围
    Call Query
End Sub

Private Sub UserForm_Initialize()
    Dim sql$, a()
    Set cnn = CreateObject("adodb.connection")
    Set rst = CreateObject("adodb.recordset") '连接数据库
    cnn.Open "Provider=Microsoft.ACE.OLEDB.12.0;Extended Properties='Excel
12.0;HDR=YES';Data Source=" & ThisWorkbook.FullName

    DTPicker1 = DateAdd("yyyy", -1, Date)
    DTPicker2 = Date

    sql = "select distinct 产品名称 from [切片图表 2$] order by 产品名称"
    rst.Open sql, cnn, 1, 3
    If rst.RecordCount = 0 Then Exit Sub
    With ComboBox2
        .Column = rst.getrows
        .ListIndex = 0
        .style = fmStyleDropDownList
    End With
    rst.Close

    With ComboBox3
        .List = Array("5 片厚度 mm", "5 片干重 g", "抗压强度 N", "卷重 kg")
        .ListIndex = 0
        .style = fmStyleDropDownList
    End With

    sql = "select distinct 班组 from [切片图表 2$] order by 班组"
    rst.Open sql, cnn, 1, 3
```

```
    If rst.RecordCount = 0 Then Exit Sub
    With ComboBox1
        .Column = rst.getrows
        .AddItem "全部", 0
        .ListIndex = 0
        .style = fmStyleDropDownList
    End With
    rst.Close: Set rst = Nothing
    Call Query
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    cnn.Close: Set cnn = Nothing
End Sub

Private Function GetSQL() As String
    If ComboBox1 = "全部" Then
        GetSQL = "select format(卷号,'J0000'),[" & ComboBox3 & "] from [切片图表 2$] where 产品名称='" & ComboBox2 & "' and 日期 between #" & DTPicker1 & "# and #" & DTPicker2 & "# order by format(卷号,'0000')"
    Else
        GetSQL = "select format(卷号,'J0000'),[" & ComboBox3 & "] from [切片图表 2$] where 产品名称='" & ComboBox2 & "' and 日期 between #" & DTPicker1 & "# and #" & DTPicker2 & "# and 班组='" & ComboBox1 & "' order by format(卷号,'0000')"
    End If
End Function

Private Sub Query()
    Dim sql$, arr(), YARR(), YMAX!, YMIN!
    If Len(ComboBox1) * Len(ComboBox2) * Len(ComboBox3) = 0 Then Exit Sub
    Set rst = CreateObject("adodb.recordset")
    sql = GetSQL: If Len(sql) = 0 Then Exit Sub
    rst.Open sql, cnn, 1, 3
    If rst.RecordCount = 0 Then Exit Sub
    arr = Application.Transpose(rst.getrows)
    rst.Close: Set rst = Nothing
    YARR = Application.Index(arr, 0, 2)
    YMAX = Application.Max(YARR)
    YMIN = Application.Min(YARR)
    With MSChart1
        .ChartData = arr
        .ChartType = VtChChartType2dLine
        .Title = ComboBox3 & " 趋势图"
        .Plot.UniformAxis = False
```

```
        .Plot.Axis(VtChAxisIdX).AxisGrid.MajorPen.style = VtPenStyleNull
        With .Plot.Axis(VtChAxisIdY).ValueScale
            .Auto = False
            .Maximum = YMAX
            .Minimum = YMIN
            .MajorDivision = 10
            .MinorDivision = 0
        End With
        With .Plot.SeriesCollection(1).Pen
            .Width = 4
            .VtColor.Set 0, 0, 255
        End With
        With .Plot.SeriesCollection(1).StatLine
            .VtColor.Set 255, 0, 0
            .Flag = VtChStatsRegression
            .style(VtChStatsRegression) = VtPenStyleDashDit
            .Width = 40
        End With
    End With
End Sub
```



2.3 组合图表

组合图表类型为 VtChChartType2dCombination，所谓组合图表，就是不同的系列可以使用不同的图表类型（XY 散点图不能组合，挺遗憾的），然后组合成复杂的分析图表。本例使用柱状图和折线图来显示完成数量和完成率，并使用了堆叠属性来显示未完成的数量，要主要的是堆叠顺序的设置，避免想要显示的系列（本例是折线图）被其他系列覆盖。完成率使用次坐标轴，要自行设计其刻度与数值刻度的对应关系，否则图表和数据就不一致了。

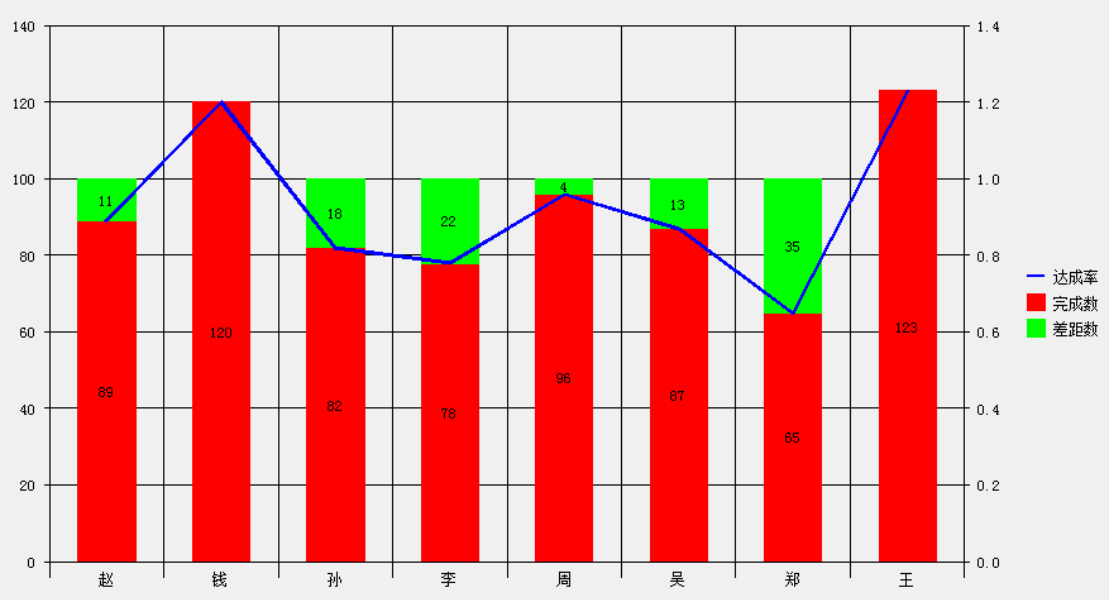
本例在附件窗体 5，使用的数据如下表：

	A	B	C	D
1		完成数	差距数	达成率
2	赵	89	11	0.89
3	钱	120		1.2
4	孙	82	18	0.82
5	李	78	22	0.78
6	周	96	4	0.96
7	吴	87	13	0.87
8	郑	65	35	0.65
9	王	123		1.23

完整代码如下：

```
Private Sub UserForm_Initialize()
Dim arr()
arr = Range("a1:d9")
With MSChart1
    .ChartData = arr
    .ChartType = VtChChartType2dCombination
    .Legend.Location.Visible = True
    .Stacking = True
    .Plot.SeriesCollection(1).DataPoints(-1).DataPointLabel.LocationType =
VtChLabelLocationTypeCenter
    .Plot.SeriesCollection(2).DataPoints(-1).DataPointLabel.LocationType =
VtChLabelLocationTypeCenter
    With .Plot.SeriesCollection(3)
        .SeriesType = VtChSeriesType2dLine '修改系列 3 的图表类型
        .SecondaryAxis = True '应用 Y 轴次坐标轴，要设计百分比和数据一致
    End With
    .Plot.SeriesCollection(3).Position.Order = 1
    .Plot.SeriesCollection(2).Position.Order = 2
    .Plot.SeriesCollection(1).Position.Order = 2 '1 和 2Order 需要一致，否则就不再堆叠了。
End With
Me.Caption = "组合图"
End Sub
```

最后的效果如下图：



2.4 动态散点图

XY 散点图与其他图表类型最大的区别就是需要 2 列数据才能显示一个系列，第一列表示 X 值，第二列表示 Y 值，其他图表类型是不需要 X 值的。如果要显示 N 个系列的 XY 散点图，需要 N*2 列的数据数组。散点图的关键在于设置好 ValueScale。下面这个例子的数据来源于网友的求助，数据很多，截图只是局部：

	A	B	C
1	0	51	
2	0.05	8	
3	0.1	-568	
4	0.15	379	
5	0.2	-800	
6	0.25	-428	
7	0.3	-603	
8	0.35	260	
9	0.4	358	
10	0.45	-679	
11	0.5	-179	

完整代码如下（附件窗体 18）：

```
Private Sub UserForm_Initialize()
Dim arr()
arr = Range("a1:b200")
With MSChart1
    .TitleText = "变化趋势图"
    .Title.VtFont.VtColor.Blue = 255
    .Plot.Axis(VtChAxisIdX).AxisTitle.Text = "时间（单位：秒）"
    .Plot.SeriesCollection(1).Pen.Width = 20
    .Plot.SeriesCollection(1).Pen.style = VtPenStyleSolid
```

```
    With .Plot.Axis(VtChAxisIdX).ValueScale
        .Auto = False
        .Maximum = 10
        .Minimum = 0
        .MajorDivision = 6
        .MinorDivision = 0
    End With
    .Plot.Axis(VtChAxisIdX).AxisGrid.MajorPen.style = VtPenStyleDotted
    With .Plot.Axis(VtChAxisIdY).ValueScale
        .Auto = False
        .Maximum = 400
        .Minimum = -800
        .MajorDivision = 12
        .MinorDivision = 0
    End With
    .Plot.Axis(VtChAxisIdY).AxisGrid.MajorPen.style = VtPenStyleDotted
    .Plot.UniformAxis = False
    .ChartType = VtChChartType2dXY
    .ChartData = arr
End With
End Sub
```

下面再举一个 XY 散点图的例子，图表显示 2 个系列的 XY 散点曲线，数据为随机数，每秒钟填充一次数据，动态显示。VBA 中没有计时器，使用 Application.OnTime 又闪烁得厉害，没法看，因此使用了 API 函数 SetTimer。关闭窗体时记得 KillTimer。控件属性 DrawMode 设置为 VtChDrawModeBlit，应该会减少闪烁。DrawMode 属性决定什么时候以及如何重绘图表，有 2 个可选值，VtChDrawModeDraw 是直接绘制到显示设备上，而 VtChDrawModeBlit 使用 Blit 方式把内存中的图形绘制到显示设备上。

窗体中的代码（附件窗体 19）为：

```
Private Sub UserForm_Initialize()
Dim i&, j&
For i = 1 To 20
    arr(i, 1) = DateAdd("s", -(20 - i), Time)
    arr(i, 2) = Rnd() * 100
    arr(i, 3) = arr(i, 1)
    arr(i, 4) = Rnd() * 80
Next
With Me.MSChart1
    With .Plot.Axis(VtChAxisIdX).ValueScale
        .Auto = False
        .MajorDivision = 10
        .MinorDivision = 0
        .Maximum = Time + TimeValue("0:0:00")
```

```
            .Minimum = Time - TimeValue("0:0:19")
        End With
        .Plot.Axis(VtChAxisIdX).Labels(1).Format = "hh:mm:ss"
        .Plot.UniformAxis = False
        .ChartType = VtChChartType2dXY
        .DrawMode = VtChDrawModeBlit
        .ChartData = arr
End With
Start_Timer
End Sub

Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
Stop_Timer
End Sub
```

因为需要回调函数，而 AddressOf 函数必须放在模块中，模块中的代码如下：

```
Private Declare Function SetTimer Lib "user32" (ByVal hwnd As Long, ByVal nIDEvent As Long, ByVal uElapse As Long, ByVal lpTimerFunc As Long) As Long
Private Declare Function KillTimer Lib "user32" (ByVal hwnd As Long, ByVal nIDEvent As Long) As Long
Private Const WM_TIMER = &H113 '计时器消息，API 常数
Private Const lDuration = 1000& '时间间隔，单位毫秒，可自行设置
Private lTimerID As Long '保存计时器 ID 号
Public arr#(1 To 20, 1 To 4)

Public Sub Start_Timer()
    If lTimerID <> 0 Then Call Stop_Timer
    lTimerID = SetTimer(0&, 0&, lDuration, AddressOf TimerProc)
End Sub

Public Sub Stop_Timer()
    KillTimer 0&, lTimerID
    lTimerID = 0
End Sub

Sub TimerProc(ByVal hwnd As Long, ByVal uMsg As Long, ByVal idEvent As Long, ByVal dwTime As Long)
    Dim i&, j&
    If uMsg = WM_TIMER And idEvent = lTimerID Then
        For i = 1 To 19
            arr(i, 1) = arr(i + 1, 1)
            arr(i, 2) = arr(i + 1, 2)
            arr(i, 3) = arr(i + 1, 3)
```

```
            arr(i, 4) = arr(i + 1, 4)
        Next
        Randomize
        With UserForm19
            arr(20, 1) = Time
            arr(20, 3) = arr(20, 1)
            arr(20, 2) = Rnd() * 100
            arr(20, 4) = Rnd() * 80
            With .MSChart1
                With .Plot.Axis(VtChAxisIdX).ValueScale
                    .Auto = False
                    .MajorDivision = 10
                    .MinorDivision = 0
                    .Maximum = Time + TimeValue("0:0:00")
                    .Minimum = Time - TimeValue("0:0:19")
                End With
                .Plot.Axis(VtChAxisIdX).Labels(1).Format = "hh:mm:ss"
                .Plot.UniformAxis = False
                .ChartType = VtChChartType2dXY
                .DrawMode = VtChDrawModeBlit
                .ChartData = arr
            End With
        End With
    End If
End Sub
```

2.5 图表缩放

图表缩放使用了 2 种技术，第一个示例使用了控件的 MSChart1.Plot.LocationRect 属性直接修改绘图区的矩形大小，缩放之后，按住鼠标可拖动图表查看。设置了 3 个功能：放大、缩小和还原，因此设置了几个变量保存原始设置。注意点是要保持绘图区的高宽比，避免在缩放过程中图形变形，还有就是 LocationRect 的坐标系是以图表控件的左上角为原点的。第二个示例是根据用户选中的数据点，获取其相邻的其他数据点数据，用另一个窗体显示出来，实现细节的展示。

第一个缩放示例（附件窗体 3）的完整代码如下，数据是散点图中的数据：

```
Private XVSMinDiv%, XVSMajDiv%
Private YVSMinDiv%, YVSMajDiv%
Private XCSDivPerTk%, XCSDivPerLb%

Private Sub CommandButton1_Click()
Dim AspectRatio!, PlotH!, PlotW!
Dim XIncrement!, YIncrement!, R!
With MSChart1
```

```
    .Plot.AutoLayout = False
    R = 0.2'一次缩放原大小的 20%
    With .Plot.LocationRect
        PlotH = .Max.Y - .Min.Y
        PlotW = .Max.X - .Min.X
        AspectRatio = PlotH / PlotW '绘图区高宽比
        XIncrement = PlotW * R 'X 轴增量
        YIncrement = (PlotW + XIncrement) * AspectRatio - PlotH 'Y 轴增量
        .Min.Set .Min.X - XIncrement / 2, .Min.Y - YIncrement / 2
        .Max.Set .Max.X + XIncrement / 2, .Max.Y + YIncrement / 2
    End With
    If .ChartType = VtChChartType2dXY Then
        With .Plot.Axis(VtChAxisIdX).ValueScale
            .Auto = False
            .MajorDivision = Application.RoundUp(.MajorDivision * (1 + R), 0)
            .MinorDivision = 0
        End With
        .Plot.Axis(VtChAxisIdX).Labels(1).Format = "0.0"
    Else
        With .Plot.Axis(VtChAxisIdX).CategoryScale
            .Auto = False
            .DivisionsPerTick = Application.Max(Int(.DivisionsPerTick / (1 + R)), 1)
            .DivisionsPerLabel = .DivisionsPerTick
        End With
    End If
    With .Plot.Axis(VtChAxisIdY).ValueScale
        .Auto = False
        .MajorDivision = Application.RoundUp(.MajorDivision * (1 + R), 0)
        .MinorDivision = 0
    End With
    .Plot.Axis(VtChAxisIdY).Labels(1).Format = "0.0"
End With
End Sub

Private Sub CommandButton2_Click()
    With MSChart1
        .Plot.AutoLayout = True
        With .Plot.Axis(VtChAxisIdX).ValueScale
            .Auto = False
            .MajorDivision = XVSMajDiv
            .MinorDivision = XVSMinDiv
        End With
        With .Plot.Axis(VtChAxisIdX).CategoryScale
            .Auto = False
```

```
            .DivisionsPerLabel = XCSDivPerLb
            .DivisionsPerTick = XCSDivPerTk
        End With
        With .Plot.Axis(VtChAxisIdY).ValueScale
            .Auto = False
            .MajorDivision = YVSMajDiv
            .MinorDivision = YVSMinDiv
        End With
    End With
End Sub

Private Sub CommandButton3_Click()
Dim AspectRatio!, PlotH!, PlotW!
Dim XIncrement!, YIncrement!, R!
With MSChart1
    .Plot.AutoLayout = False
    R = 0.2
    With .Plot.LocationRect
        PlotH = .Max.Y - .Min.Y
        PlotW = .Max.X - .Min.X
        AspectRatio = PlotH / PlotW
        XIncrement = PlotW * R
        YIncrement = PlotH - (PlotW - XIncrement) * AspectRatio
        .Min.Set .Min.X + XIncrement / 2, .Min.Y + YIncrement / 2
        .Max.Set .Max.X - XIncrement / 2, .Max.Y - YIncrement / 2
    End With
    If .ChartType = VtChChartType2dXY Then
        With .Plot.Axis(VtChAxisIdX).ValueScale
            .Auto = False
            .MajorDivision = Application.Max(Int(.MajorDivision / (1 + R)), 1)
            .MinorDivision = 0
        End With
        .Plot.Axis(VtChAxisIdX).Labels(1).Format = "0.0"
    Else
        With .Plot.Axis(VtChAxisIdX).CategoryScale
            .Auto = False
            .DivisionsPerTick = Application.RoundUp(.DivisionsPerTick * (1 + R), 0)
            .DivisionsPerLabel = .DivisionsPerTick
        End With
    End If
    With .Plot.Axis(VtChAxisIdY).ValueScale
        .Auto = False
        .MajorDivision = Application.Max(Int(.MajorDivision / (1 + R)), 1)
        .MinorDivision = 0
```

```
        End With
        .Plot.Axis(VtChAxisIdY).Labels(1).Format = "0.0"
End With
End Sub

Private Sub UserForm_Initialize()
Dim arr()
arr = Range("a1:b200")
With MSChart1
        .TitleText = "变化趋势图"
        .Title.VtFont.VtColor.Blue = 255
        .Plot.Axis(VtChAxisIdX).AxisTitle.Text = "时间（单位：秒）"
        .Plot.SeriesCollection(1).Pen.Width = 20
        .Plot.SeriesCollection(1).Pen.style = VtPenStyleSolid
        With .Plot.Axis(VtChAxisIdX).ValueScale
                .Auto = False
                .Maximum = 10
                .Minimum = 0
                .MajorDivision = 6
                .MinorDivision = 0
                XVSMinDiv = .MinorDivision
                XVSMajDiv = .MajorDivision
        End With
        With .Plot.Axis(VtChAxisIdX).CategoryScale
                XCSDivPerTk = .DivisionsPerTick
                XCSDivPerLb = .DivisionsPerLabel
        End With
        With .Plot.Axis(VtChAxisIdY).ValueScale
                .Auto = False
                .Maximum = 400
                .Minimum = -800
                .MajorDivision = 12
                .MinorDivision = 0
                YVSMinDiv = .MinorDivision
                YVSMajDiv = .MajorDivision
        End With
        .Plot.Axis(VtChAxisIdX).AxisGrid.MajorPen.style = VtPenStyleDotted
        .Plot.Axis(VtChAxisIdY).AxisGrid.MajorPen.style = VtPenStyleDotted
        .Plot.UniformAxis = False
        .ChartType = VtChChartType2dXY
        .ChartData = arr
End With
End Sub
```

第二个缩放示例（附件窗体 2）完整代码如下，核心就是取得选中的数据点周围的数据：

```
Private Sub MSChart1_PointSelected(Series As Integer, DataPoint As Integer, MouseFlags As Integer, Cancel As Integer)

    Dim PartData(), XARR(), YARR()
    Dim XMAX!, XMIN!, YMAX!, YMIN!
    Dim pt%, i%, j%, k%

    With MSChart1
        If .rowCount < 7 Then Exit Sub
        ReDim PartData(5, .columnCount)
        Select Case DataPoint
            Case 1, 2
                pt = 1
            Case .rowCount - 1, .rowCount
                pt = .rowCount - 4
            Case Else
                pt = DataPoint - 2
        End Select
        For i = 0 To .columnCount
            PartData(0, i) = .ChartData(0, i) '列标签
        Next
        For i = pt To pt + 4
            k = k + 1
            PartData(k, 0) = .ChartData(i, 0) '行标签
            For j = 1 To .columnCount
                PartData(k, j) = .ChartData(i, j)
            Next
        Next
    End With

    XARR = Application.Index(PartData, 0, 2)
    XMAX = Application.Max(XARR)
    XMIN = Application.Min(XARR)
    YARR = Application.Index(PartData, 0, 3)
    YMAX = Application.Max(YARR)
    YMIN = Application.Min(YARR)
    With UserForm6.MSChart1
        .ChartType = MSChart1.ChartType
        .Plot.UniformAxis = False
        .ChartData = PartData
        With .Plot.Axis(VtChAxisIdX).ValueScale
```

```
            .Auto = False
            .Maximum = XMAX
            .Minimum = XMIN
            .MajorDivision = 10
            .MinorDivision = 2
        End With
        .Plot.Axis(VtChAxisIdX).Pen.VtColor.Red = 255
        With .Plot.Axis(VtChAxisIdY).ValueScale
            .Auto = False
            .Maximum = YMAX
            .Minimum = YMIN
            .MajorDivision = 10
            .MinorDivision = 2
        End With
    End With
    UserForm6.Show
End Sub

Private Sub UserForm_Initialize()
Dim arr()
arr = Range("a1:b200")
With MSChart1
    .TitleText = "变化趋势图"
    .Title.VtFont.VtColor.Blue = 255
    .Plot.Axis(VtChAxisIdX).AxisTitle.Text = "时间（单位：秒）"
    With .Plot.SeriesCollection(1)
        .Pen.Width = 20
        .SeriesMarker.Auto = False
        .SeriesMarker.Show = True
        .DataPoints(-1).Marker.style = VtMarkerStyle3dBall
        .DataPoints(-1).Marker.Size = 120
        .DataPoints(-1).Marker.FillColor.Set 0, 0, 100
    End With
    With .Plot.Axis(VtChAxisIdX).ValueScale
        .Auto = False
        .Maximum = 10
        .Minimum = 0
        .MajorDivision = 6
        .MinorDivision = 0
    End With
    With .Plot.Axis(VtChAxisIdY).ValueScale
        .Auto = False
        .Maximum = 400
        .Minimum = -800
```

```
            .MajorDivision = 12
            .MinorDivision = 0
        End With
        .AllowSeriesSelection = False
        .Plot.UniformAxis = False
        .ChartType = VtChChartType2dXY
        .ChartData = arr
End With
End Sub
```

2.6 图表保存和打印

MSChart 控件在保存图表和打印这方面基本是空白，唯一提供的方法是 EditCopy 方法，该方法以 Windows 图元文件格式将当前图表的图片复制到剪贴板中，同时将创建图表使用的数据复制到剪贴板中。（保存图表和打印在附件窗体 20）

2.6.1 保存图表

在 VB6 中要保存图表文件是非常容易的，用剪贴板对象 Get 数据：Clipboard.GetData，然后使用 SavePicture 函数就可以把图表保存到磁盘中。但在 VBA 中却很麻烦，因为 VBA 的剪贴板只支持文本，无法获取剪贴板上的图片数据。好在可用 OLE 技术（Object Linking and Embedding，对象连接与嵌入）。
保存为 BMP 格式，直接使用 OleCreatePictureIndirect 实现访问标准图片对象（stdPicture），使用的接口为 IPicture，接口的 GUID 为："{7BF80980-BF32-101A-8BBB-00AA00300CAB}"。使用 OleCreatePictureIndirectIPic 得到的 stdPicture 对象可用 SavePicture 函数保存到磁盘，也可以直接赋值给 PictureBox、Image 等带有 Picture 属性的控件使用，而无需使用磁盘中转，然后用 Loadpicture 导入。

保存为 BMP 格式完全代码如下：

```
Private Declare Function GetClipboardData Lib "user32" (ByVal wFormat As Long) As Long
Private Declare Function CloseClipboard Lib "user32" () As Long
Private Declare Function OpenClipboard Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function OleCreatePictureIndirect Lib "olepro32.dll" (PicDesc As PicBmp, RefIID As GUID, ByVal fPictureOwnsHandle As Long, IPic As IPicture) As Long
Private Declare Function CLSIDFromString Lib "ole32" (ByVal lpsz As Long, pclsid As GUID) As Long
Private Const sIID_IPicture As String = "{7BF80980-BF32-101A-8BBB-00AA00300CAB}"
Private Const CF_BITMAP As Long = 2

Private Type PicBmp
    Size As Long
```

```
    type As Long
    hBmp As Long
    hPal As Long
    Reserved As Long
End Type

Private Type GUID
    Data1 As Long
    Data2 As Integer
    Data3 As Integer
    Data4(0 To 7) As Byte
End Type

Sub SaveClipToBMP(ByVal FileName As String)
    Dim Pic As PicBmp, IPic As IPicture
    Dim hBmp As Long, IID_IDispatch As GUID
    CLSIDFromString StrPtr(sIID_IPicture), IID_IDispatch
    If OpenClipboard(ByVal 0&) Then
        hBmp = GetClipboardData(CF_BITMAP)
        CloseClipboard
        If hBmp Then
            With Pic
                .Size = Len(Pic)
                .type = 1
                .hBmp = hBmp
            End With
            OleCreatePictureIndirect Pic, IID_IDispatch, 1, IPic
            SavePicture IPic, FileName
        End If
    End If
End Sub
```

保存为 BMP 格式时，文件非常大，可以考虑保存为 JPG 格式，虽然是损压缩，但文件大小可减小 10 倍以上。从 BMP 格式到 JPG 格式需要编解码，完整代码如下：

```
Private Declare Function OpenClipboard Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function CloseClipboard Lib "user32" () As Long
Private Declare Function GetClipboardData Lib "user32" (ByVal Format As Long) As Long
Private Declare Function GdiplusStartup Lib "gdiplus" (token As Long, inputbuf As
GdiplusStartupInput, Optional ByVal outputbuf As Long = 0) As Long
Private Declare Function GdiplusShutdown Lib "gdiplus" (ByVal token As Long) As Long
Private Declare Function GdipCreateBitmapFromHBITMAP Lib "gdiplus" (ByVal hbm As Long,
ByVal hPal As Long, BITMAP As Long) As Long
Private Declare Function GdipDisposeImage Lib "gdiplus" (ByVal Image As Long) As Long
```

```
Private Declare Function GdipSaveImageToFile Lib "gdiplus" (ByVal Image As Long, ByVal FileName As Long, clsidEncoder As GUID, encoderParams As Any) As Long
Private Declare Function CLSIDFromString Lib "ole32" (ByVal lpsz As Long, pclsid As GUID) As Long
Private Const CLSID_JPG As String = "{557CF401-1A04-11D3-9A73-0000F81EF32E}"
Private Const EncoderQuality As String = "{1D5BE4B5-FA4A-452D-9CDD-5DB35105E7EB}"
Private Const CF_BITMAP As Long = 2

Private Type GUID
    Data1 As Long
    Data2 As Integer
    Data3 As Integer
    Data4(0 To 7) As Byte
End Type

Private Type GdiplusStartupInput
    GdiplusVersion As Long
    DebugEventCallback As Long
    SuppressBackgroundThread As Long
    SuppressExternalCodecs As Long
End Type

Private Type EncoderParameter
    GUID As GUID
    NumberOfValues As Long
    type As Long
    value As Long
End Type

Private Type EncoderParameters
    count As Long
    Parameter As EncoderParameter
End Type

Sub SaveClipToJPG(ByVal FileName As String)
     Dim hMem As Long, BITMAP As Long
     Dim Quality As Long, lRet As Long
     Dim GDI_Token As Long
     Dim GpInput As GdiplusStartupInput
     Dim Params As EncoderParameters
     Dim JPGID As GUID, ECQID As GUID
     If OpenClipboard(ByVal 0&) = 0 Then Exit Sub
     hMem = GetClipboardData(CF_BITMAP)
     CloseClipboard
```

```
    If hMem = 0 Then Exit Sub
    GpInput.GdiplusVersion = 1
    lRet = GdiplusStartup(GDI_Token, GpInput)
    If lRet <> 0 Then Exit Sub
    GdipCreateBitmapFromHBITMAP hMem, 0, BITMAP
    CLSIDFromString StrPtr(EncoderQuality), ECQID
    Quality = 100
    With Params
        .count = 1
        With .Parameter
            .GUID = ECQID
            .NumberOfValues = 1
            .type = 4
            .value = VarPtr(Quality)
        End With
    End With
    CLSIDFromString StrPtr(CLSID_JPG), JPGID
    GdipSaveImageToFile BITMAP, StrPtr(FileName), JPGID, Params
    GdipDisposeImage BITMAP
    GdiplusShutdown GDI_Token
End Sub
```

我们这里只写了 2 种格式的保存代码，应该已经足够使用了。在窗体中用 GetSaveAsFilename 方法获得文件名和保存路径，根据文件名后缀判断保存为 BMP 还是 JPG 格式，代码如下：

```
Private Sub CommandButton1_Click()
    Dim InitName, FileFilter, FileName
    MSChart1.EditCopy
    InitName = Format(Now, "yyyy-mm-dd hhmmss") '预设文件名
    FileFilter = "JPG 图片(*.jpg), *.jpg,BMP 图片(*.bmp),*.bmp"
    FileName = Application.GetSaveAsFilename(InitName, FileFilter)
    If FileName <> False Then
        If UCase(Right(FileName, 3)) = "JPG" Then
            SaveClipToJPG FileName   '保存为 JPG 格式图片
        Else
            SaveClipToBMP FileName '保存为 BMP 格式图片
        End If
    End If
End Sub
```

除了把图表保存到磁盘上，还可以粘贴到 EXCEL 工作表指定位置。代码简单得多了：

```
Sub PasteToExcelPicture(ByVal pictruename As String)
    Dim Pic As Shape
```

```
    For Each Pic In ActiveSheet.Shapes
        If Pic.Name = pictruename Then Pic.Delete
    Next
    With ActiveSheet.Pictures.Paste
        .Name = pictruename
        .Placement = xlFreeFloating
        .ShapeRange.LockAspectRatio = msoTrue
        .Top = Range("a1").Top
        .Left = Range("a1").Left
        .Height = 400
    End With
End Sub
```

除了保存图表的图片，其实还可以保存图表的数据：ActiveSheet.Paste。

把图表保存到 EXCEL 工作表，在窗体中的代码如下：

```
Private Sub CommandButton3_Click()
    MSChart1.EditCopy
    PasteToExcelPicture "test" '图表粘贴到 EXCEL
'    ActiveSheet.Paste '粘贴为数据
End Sub
```

2.6.2 打印图表

在窗体上直接打印，用：Me.PrintForm，该方法还会打印窗体上的控件，因此打印前用代码隐藏一下不需要打印的控件。我一般是复制到 EXCEL 工作表中打印，可预览，可调节边界：

```
Private Sub CommandButton2_Click() '打印
'    Me.PrintForm
    Dim AspectRatio!, PlotH!, PlotW!
    MSChart1.EditCopy
    With MSChart1.Plot.LocationRect
        PlotH = .Max.Y - .Min.Y
        PlotW = .Max.X - .Min.X
        AspectRatio = PlotH / PlotW
    End With
    PasteToExcelPicture "test" '图表粘贴到 EXCEL
    PrintPageSetup IIf(AspectRatio > 1, 1, 2)
    Me.Hide
    ActiveSheet.PrintOut Preview:=True
    Me.Show
End Sub
```

我先计算图表控件绘图区的高宽比，来自动决定横向还是纵向纸张，然后使用 PrintPageSetup 打印设置。PrintPageSetup 过程的代码如下：

```
Sub PrintPageSetup(ByVal Orient As Long)
    With ActiveSheet.PageSetup
        .HeaderMargin = Application.InchesToPoints(0.1)
        .TopMargin = Application.InchesToPoints(0.12)
        .LeftMargin = Application.InchesToPoints(0.1)
        .RightMargin = Application.InchesToPoints(0.1)
        .BottomMargin = Application.InchesToPoints(0.12)
        .FooterMargin = Application.InchesToPoints(0.1)
        .FitToPagesWide = 1
        .FitToPagesTall = 1
        .CenterVertically = 1
        .CenterHorizontally = 1
        .Orientation = Orient '横向 2;纵向 1
    End With
End Sub
```

——END——