

MSChart 控件在 C++6.0 和 Microsoft visual studio 2005 中使用都需要额外安装控件，但是 Microsoft visual studio 2010 中已经内含此控件，所以此教程用于 Microsoft visual studio 2010 版本使用。

1: 新建 MFC 工程，此处可以选择对话框程序，如图 1 所示，其余设置可默认。



图 1

2: MSChart 控件位于工具箱下端的常规控件选项卡中，若是工具箱面板中未找到，可以右击工具箱→选择项→COM 组件中选择 Microsoft Chart Control, version5.0。控件位置及图标如图 2 所示。

注：此处除 Microsoft Chart Control, version5.0 外，还有 Microsoft Chart Control 6.0，所使用的是 version5.0，但添加的类却是 6.0，对于此处差别还不太清楚。



图 2

3: 添加 Microsoft Chart Control 控件到对话框程序中，如图 3 所示。

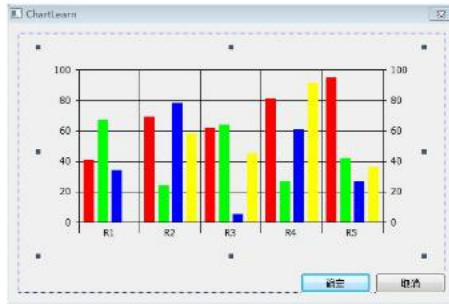


图 3

4: 点击 Microsoft Chart Control 控件, 右端出现属性窗口, 如图 4 所示, 在此处可以对于此控件进行设置, 下面对于各个控件属性含义进行具体讲解。

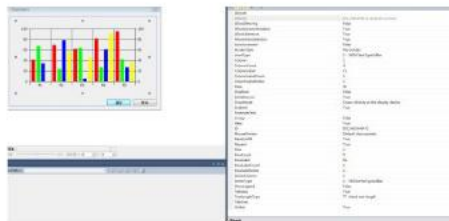


图 4

更改属性页内属性, 属性页位于属性列表上端快捷键最后一位, 属性页如图 5 所示。

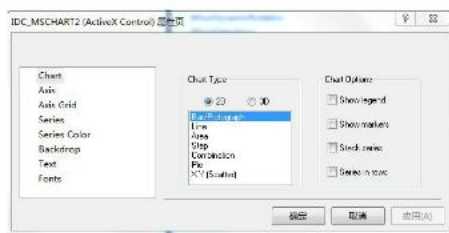


图 5

属性列表主要包括以下几个方面的设置:

Chart 主要包括对于图表类型的选择, 此例中主要使用的是 Line (2D), 选项 Show legend 为显示图例, 在此例中需要进行勾选。设置如图 6 所示。



图 6

Axis 主要是对于各个坐标轴的设置, 其中 Axis 可以进行 X Axis、Y Axis 及 Second Y Axis 三个轴的选择, 其中 Show scale 选项可以选择对应轴是否显示, Automatic scaling 可以对于坐标轴上最大值、最小值、标注标签间隔和每个间隔细分格数 (对于 X 及 Y 轴是不同的, X 轴若是 100 为一个格, 50 需要为一个小格, Y 轴 Major divisions 对应的是大格格数, Minor divisions 对应小格格数)。在对于此处设置之前, 需要将属性列表中的 RowCount 值更改, 此例需要将横轴分成 1000 份, 所以 RowCount 值改为 1000, 同时 Row 与 RowLabel 对应更改, 此类每 100 份表示为 1 格, 标注为 0.1 变化, 所以对应于下表进行更改。

Row	1	101	201	301	401	501	601	701	...
-----	---	-----	-----	-----	-----	-----	-----	-----	-----

RowLabel	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	...
----------	-----	-----	-----	-----	-----	-----	-----	-----	-----

同时将 RandomFill 选项改为 FALSE。

X 轴及 Y 轴设置如图 7 及图 8 所示，此处不需要 Second Y Axis，所以将其 Show scale 选项勾选掉。Color 可以修改对应轴的颜色，此处不进行解释。



图 7

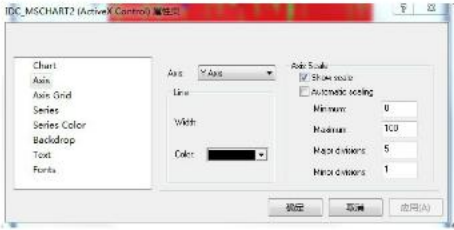


图 8

表格形状如图 9 所示。



图 9

Axis Grid 主要是对于表格中的网格线进行更改，此处为了更清晰的表示曲线，去除掉所有的网格线，故此例中将 Color 更改为白色即可。如图 10 所示。



图 10

Series 是对于图标中曲线个数进行设置，此处需要先对属性列表进行更改，ColumnCount 是曲线个数，此例共有两条曲线，故此处设置为 2，Column 及 ColumnLabel 对应设置，当 Column 为 1 时，设置 ColumnLabel 为  $y=200x$ ，Column 为 2 时，设置 ColumnLabel 为  $y=100x$ 。此处含义是当 x 轴对应 0.1 标签时，两个曲线分别对应于 20 及 10，但是其实 x 轴 0.1 对应于 100Row，这里有相对应关系。Statistics 选项卡中，主要可以画出最高最低限度、平均、标准及回归等曲线，此处不用勾选，设置如图 11 所示。

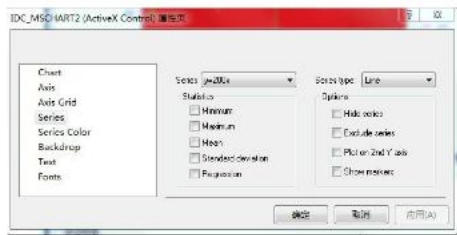


图 11

Series Color 对于曲线颜色设置, 此处将两条曲线分别设置为红色与绿色, 主要更改 Color 选项。

Backdrop 主要是对于背景的设置, 可以设置背景填充色以及边框填充, 此例将背景设置为蓝色, 设置如图 12 所示。

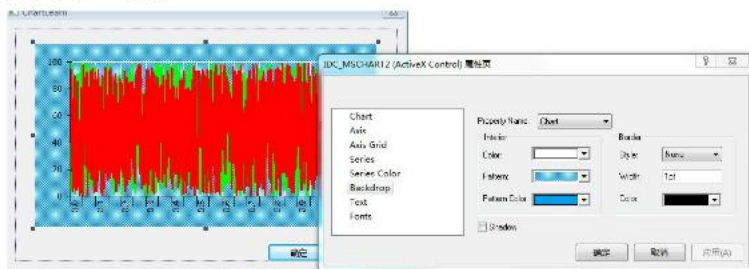


图 12

Text 及 Fonts 主要是对于标题、各轴标注等字体内容设置, 这些设置可以选择用语句进行, 所以此处只将标题字体 Title 改为仿宋, 各轴标签字体 Axis Label 是改为 Time New Roman。

对于属性页设置如上所述, 对于属性表中, 只需要再将 Enabled 选项改为 FALSE, 此处目的是在显示曲线时, 不可对控件进行操作。若图例过长从而显示不完全, 可以在属性页中, 更改 Fonts 选项卡中的 Legend 属性即可。

5: 添加 CMSChart 类到工程中, 点击类视图中的工程名, 右击选项卡中选择添加→类, 此处一定要注意, 在类视图中添加类和在项目选项卡中添加类是不同的, 之前也是因为这个原因而困扰很久, 此处位置如图 13 所示。

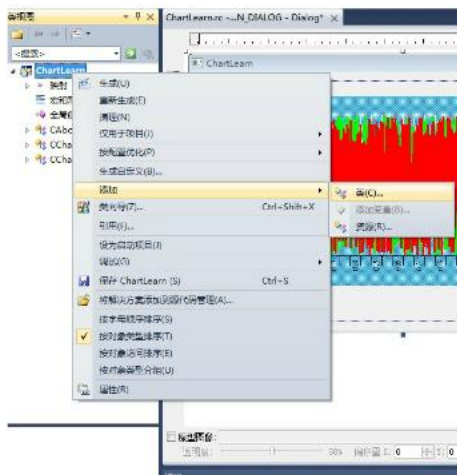


图 13

在类添加选项卡中双击 TypeLib 中的 MFC 类这个类型。在类型库中添加类向导中, 选择可用的类型库文件名为 Microsoft Chart Control 6.0 (SP4), 然后点击接口与生产的类中间的添加按钮, 设置如图 14 所示。之后点击完成即可。







图 15

此时编译会出现如图 16 错误。



图 16

点击错误，跳转至错误的位置，需要将 `m_chart1(0)` 中的 0 删除，此错误即可解除。具体原因未明。再编译即可通过。

7: 添加函数对于 MSChart 控件进行初始化设置，此时点击 `CChartLearnDlg` 类中的 `OnInitDialog()` 函数，在其末端添加代码，网上 C++6.0 在 MSChart 控件所使用的函数很多在此版本都是不可以的，所以此处所列的函数只为一些我所使用的函数。

注：此处一些设置可以在属性中直接更改。

(1) 添加标题。

```
m_chart1.put_TitleText(_T("一次曲线"));
```

(2) 更改标题颜色及大小。

```
CVcTitlevcTitle;  
CVcFontvcFont;  
CVcColorvcColor;
```

```
vcTitle = m_chart1.get_Title();  
vcFont = vcTitle.get_VtFont();  
vcColor = vcFont.get_VtColor();  
vcColor.Set(0, 255, 0);  
vcFont.put_Size(12);
```

(3) 坐标轴显示标签。

```
VARIANT var;  
CVcPlotvcPlot;  
CVcAxisvcAxis;  
CVcAxisTitlevcAxisTitle;
```

```
vcPlot = m_chart1.get_Plot();  
vcAxis = vcPlot.get_Axis(0, var);  
vcAxisTitle = vcAxis.get_AxisTitle();
```

```
vcAxiStitle.put_Text(T("时间 t (s)"));
```

函数 `get.Axis()` 中的 0 代表横轴，1 代表左纵轴，2 代表右纵轴。

(4) 设置坐标轴最大及最小值。此例中在属性中进行设置，所以没有使用。

```
vcplot = m_chart1.get_Plot();
```

```
vcaxis = vcplot.get_Axis(1, var);
```

```
CVcValueScale vcvalue1 = vcaxis.get_ValueScale();
```

```
CVcCategoryScale vccat1 = vcaxis.get_CategoryScale();
```

```
vccat1.put_Auto(FALSE);
```

```
vcvalue1.put_MajorDivision(20);
```

```
vcvalue1.put_Maximum(2000);
```

```
vcvalue1.put_Minimum(0);
```

(5) 输出点函数

```
CVcDataGridvcDataGrid = m_chart1.get_DataGrid();
```

```
vcDataGrid.SetData(i, 1, i, 0);
```

```
vcDataGrid.SetData(i, 2, 2*i, 0);
```

其中 `SetData` 函数中，第一位为 `Row`，其实也就是横轴位置，第二位为 `Column`，也就是对应曲线，第三位为数值，第四位为 0，具体含义未知。

此函数将输出点至于“确定”按钮对应函数中，所以需要双击确定按钮添加函数，如图 17 所示。

```
void CChartLearnDlg::OnBnClickedOk()  
{  
    // TODO: 在此添加控件通知处理程序代码  
    CDialogEx::OnOK();  
}
```

图 17

删掉 `CDialogEx::OnOK()` 语句

添加如下语句：

```
for(int i=1; i<=1000; i++)
```

```
{
```

```
    CVcDataGridvcDataGrid = m_chart1.get_DataGrid();
```

```
    vcDataGrid.SetData(i, 1, 0.2*i, 0);
```

```
    vcDataGrid.SetData(i, 2, 0.1*i, 0);
```

```
}
```

8: 执行函数。

F5 快捷键执行 `Debug`，显示如图 18 所示，当点击确定按钮时，会进行计算并显示，如图 19 所示。

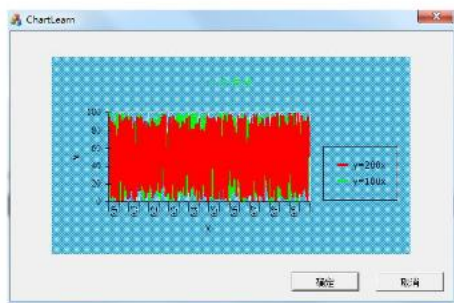


图 18

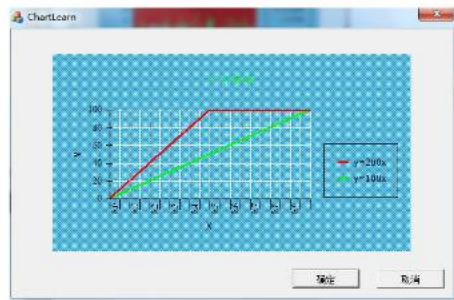


图 19

9: 关于显示数据来说, 其实每个点对应的是一个柱状图, 所以将 Row1 设置为 0, 而 Row101 设置为 0.1, 在表格中 0-0.1 其实已经画了 100 个数据, 不是单纯的对于 x 轴所标注的值进行绘图, 所选取的变量其实是 Row 数, 对于每个 Row 都有一个对应点数据值, 所以此处标注值与 Row 有对应关系, 但是二者一般情况下不同。