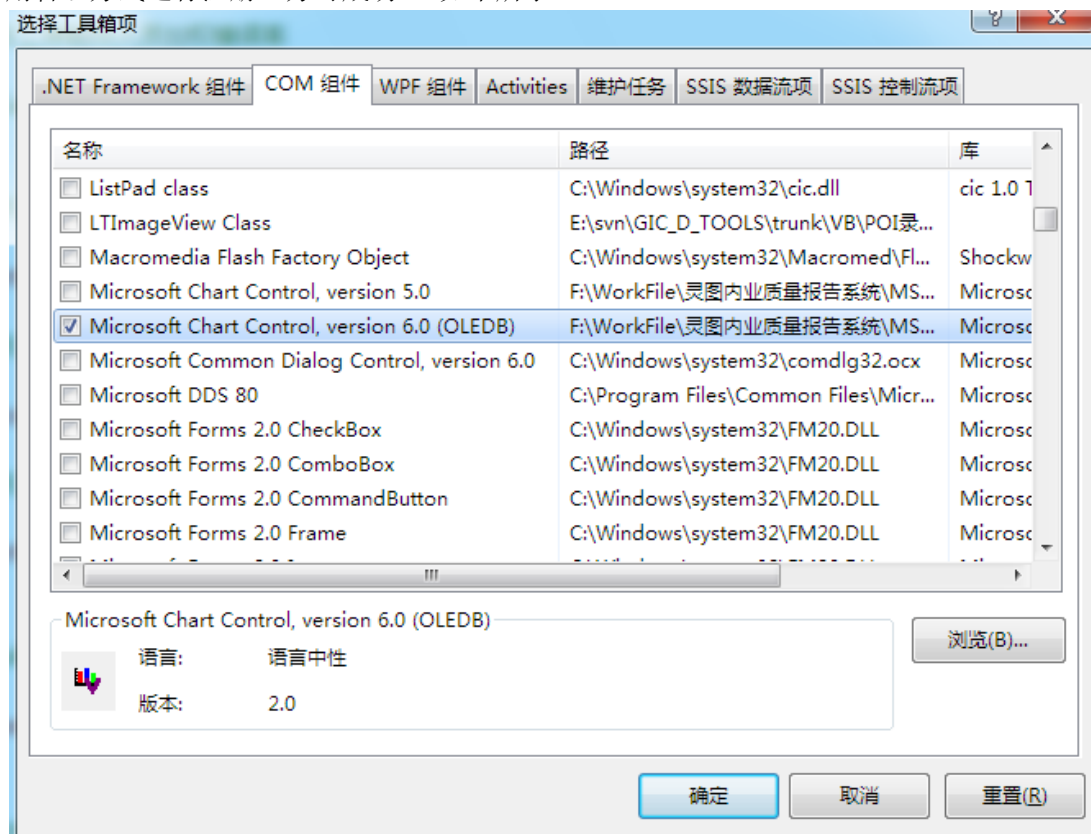


MFC 中 MSChart 的使用示例

最近由于工作的原因，接触了一些 MSChart 的东西。在查看的资料的时候发现，网上很多资料是关于在 VB 和 C# 中利用 MSChart 实现图表的绘制，但关于在 MFC 中利用 MSChart 实现图表绘制的资料甚少，因此在做的过程中遇到了很多问题，下面对 MFC 中使用 MSChart 进行简单的小结：

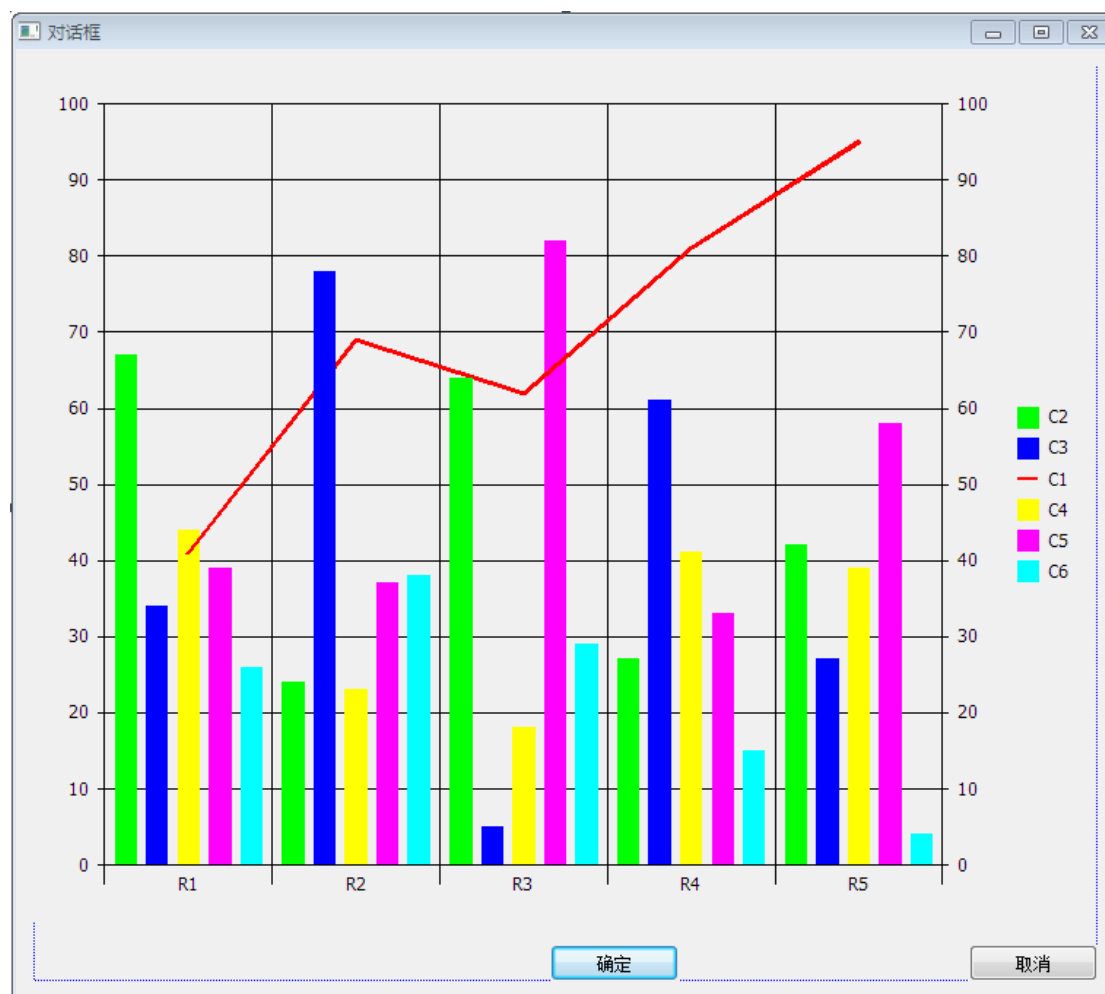
1. 添加 MSChart

点击 VS 中的“工具-选择工具箱-COM 组件”，找到 Microsoft Chart Control, version 6.0(OLEDB)并勾选(注：如果 COM 组件中不存在该项，便可自行安装，从网上下载 MSChart.exe 和 MSCHART20.OCX。首先安装 MSChart.exe，然后注册 MSCHART20.OCX，如果是 win7 下要用管理方式运行注册，方可成功)。如下所示：



2. 在对话框中添加 MSChart 控件

这个比较简单，在此不再赘述，如下所示：



3.属性设置

MSChart 控件的属性及说明如下，可根据需要自行设置。

ChartAreas: 增加多个绘图区域，每个绘图区域包含独立的图表组、数据源，用于多个图表类型在一个绘图区不兼容时。

AlignmentOrientation: 图表区对齐方向，定义两个绘图区域间的对齐方式。

AlignmentStyle: 图表区对齐类型，定义图表间用以对其的元素。

AlignWithChartArea: 参照对齐的绘图区名称。

InnerPlotPosition: 图表在绘图区内的位置属性。

Auto: 是否自动对齐。

Height: 图表在绘图区内的高度（百分比，取值在 0-100）

Width: 图表在绘图区内的宽度（百分比，取值在 0-100）

X,Y: 图表在绘图区内左上角坐标

Position: 绘图区位置属性，同 InnerPlotPosition。

Name: 绘图区名称。

Axis: 坐标轴集合

Title: 坐标轴标题

TitleAlignment: 坐标轴标题对齐方式

Interval: 轴刻度间隔大小

IntervalOffset: 轴刻度偏移量大小

MinorGrid: 次要辅助线

MinorTickMark: 次要刻度线

MajorGrid: 主要辅助线

MajorTickMark: 主要刻度线

DataSourceID: MSChart 的数据源。

Legends: 图例说明。

Palette: 图表外观定义。

IsValueShownAsLabel: 是否显示数据点标签，如果为 `true`，在图表中显示每一个数据值

Label: 数据点标签文本

LabelFormat: 数据点标签文本格式

LabelAngle: 标签字体角度

Name: 图表名称

Points: 数据点集合

XValueType: 横坐标轴类型

上面的属性来自别处，可能有些并非 MFC 中的属性，但大致意思还是差不多，故在此一一列举。

杂项	
(About)	
(Name)	IDC_MSCHART1 (ActiveX Control)
AllowDithering	False
AllowDynamicRotation	True
AllowSelections	False
AllowSeriesSelection	True
AutoIncrement	False
BorderStyle	No border.
chartType	9 - VtChChartType2dCombination
Column	1
ColumnCount	6
ColumnLabel	C1
ColumnLabelCount	1
ColumnLabelIndex	1
Data	41
Data Field	
Data Source	<未绑定到数据源>
DataMember	
Disabled	False
DoSetCursor	True
DrawMode	Draws directly to the display device
Enabled	True
FootnoteText	
Group	False
Help	True
ID	IDC_MSCHART1
MousePointer	Default chart pointer.
OLEDragMode	0 - chOLEDragManual
OLEDropMode	0 - chOLEDropNone
RandomFill	True
Repaint	True
Row	1
RowCount	5
RowLabel	R1

chartType	9 - VtChChartType2dCombination
Column	1
ColumnCount	6

个人见解：一个 Chart 中可包含多个 Series，ColumnCount 为 Series 个数，可根据需要为不同的 Series 设置显示类型（SeriesType），例如：

```
m_Chart.GetPlot().GetSeriesCollection().GetItem(5).SetSeriesType(6);
```

Data Field	
Data Source	<未绑定到数据源>
DataMember	
Disabled	False
DoSetCursor	True
DrawMode	Draws directly to the display device
Enabled	True
FootnoteText	
Group	False
Help	True
ID	IDC_MSCHART1
MousePointer	Default chart pointer.
OLEDragMode	0 - chOLEDragManual
OLEDropMode	0 - chOLEDropNone
RandomFill	True
Repaint	True
Row	1
RowCount	5
RowLabel	R1
RowLabelCount	1
RowLabelIndex	1
SeriesColumn	3
SeriesType	6 - VtChSeriesType2dLine
ShowLegend	True
Tabstop	True
TextLengthType	TT virtual text length
TitleText	
Visible	True

4.代码示例

```
void DemoView::InitChart()
{
    // 设置标题
    m_Chart.SetTitleText("MSChart 示例 By gis_swb@163.com");

    // 下面两句改变背景颜色
```

```
m_Chart.GetBackdrop().GetFill().SetStyle(1);
m_Chart.GetBackdrop().GetFill().GetBrush().GetFillColor().Set(255, 255, 255);

// 设置SeriesType
m_Chart.SetSeriesType(6); //2dLine

// 6条曲线
m_Chart.SetColumnCount(6);

// 显示图例
m_Chart.SetShowLegend(TRUE);
m_Chart.SetColumn(1);
m_Chart.SetColumnLabel((LPCTSTR)"1号机");
m_Chart.SetColumn(2);
m_Chart.SetColumnLabel((LPCTSTR)"2号机");
m_Chart.SetColumn(3);
m_Chart.SetColumnLabel((LPCTSTR)"3号机");

m_Chart.SetColumn(4);
m_Chart.SetColumnLabel((LPCTSTR)"4号机");
m_Chart.SetColumn(5);
m_Chart.SetColumnLabel((LPCTSTR)"5号机");
m_Chart.SetColumn(6);
m_Chart.SetColumnLabel((LPCTSTR)"6号机");
// 栈模式
m_Chart.SetStacking(FALSE);

// Y轴设置
VARIANT var;
// 不自动标注Y轴刻度
m_Chart.GetPlot().GetAxis(1,var).GetValueScale().SetAuto(FALSE);
// Y轴最大刻度
m_Chart.GetPlot().GetAxis(1,var).GetValueScale().SetMaximum(100);
// Y轴最小刻度
m_Chart.GetPlot().GetAxis(1,var).GetValueScale().SetMinimum(0);
// Y轴刻度等分
m_Chart.GetPlot().GetAxis(1,var).GetValueScale().SetMajorDivision(5);
// 每刻度一个刻度线
m_Chart.GetPlot().GetAxis(1,var).GetValueScale().SetMinorDivision(1);
// Y轴名称
m_Chart.GetPlot().GetAxis(1,var).GetAxisTitle().SetText("小时");

// 第二Y轴属性
// 不自动标注第二Y轴
```

```
m_Chart.GetPlot().GetAxis(2,var).GetValueScale().SetAuto(FALSE);
// 第二Y轴最大刻度
m_Chart.GetPlot().GetAxis(2,var).GetValueScale().SetMaximum(10);
m_Chart.GetPlot().GetAxis(2,var).GetValueScale().SetMinimum(0);
m_Chart.GetPlot().GetAxis(2,var).GetValueScale().SetMajorDivision(5);
m_Chart.GetPlot().GetAxis(2,var).GetValueScale().SetMinorDivision(1);
m_Chart.GetPlot().GetAxis(2,var).GetAxisTitle().SetText("Test");
//Y轴名称排列方式
m_Chart.GetPlot().GetAxis(1,var).GetAxisTitle().GetTextLayout().SetOrientation(1);
m_Chart.GetPlot().GetAxis(2,var).GetAxisTitle().GetTextLayout().SetOrientation(1);

// 线色
m_Chart.GetPlot().GetSeriesCollection().GetItem(1).GetPen().GetVtColor().Set(0, 0, 255);
m_Chart.GetPlot().GetSeriesCollection().GetItem(2).GetPen().GetVtColor().Set(255, 0, 0);
m_Chart.GetPlot().GetSeriesCollection().GetItem(3).GetPen().GetVtColor().Set(0, 255, 0);
m_Chart.GetPlot().GetSeriesCollection().GetItem(4).GetPen().GetVtColor().Set(0, 0, 255);
m_Chart.GetPlot().GetSeriesCollection().GetItem(5).GetPen().GetVtColor().Set(255, 0, 0);
m_Chart.GetPlot().GetSeriesCollection().GetItem(6).GetPen().GetVtColor().Set(0, 255, 0);

// 线宽 (对点位图有效)
//m_Chart.GetPlot().GetSeriesCollection().GetItem(1).GetPen().SetWidth(50);
//m_Chart.GetPlot().GetSeriesCollection().GetItem(2).GetPen().SetWidth(100);
//m_Chart.GetPlot().GetSeriesCollection().GetItem(3).GetPen().SetWidth(2);
//m_Chart.GetPlot().GetSeriesCollection().GetItem(4).GetPen().SetWidth(10);
//m_Chart.GetPlot().GetSeriesCollection().GetItem(5).GetPen().SetWidth(10);
//m_Chart.GetPlot().GetSeriesCollection().GetItem(6).GetPen().SetWidth(10);

// 设置第二Y轴是否可用
m_Chart.GetPlot().GetSeriesCollection().GetItem(1).SetSecondaryAxis(FALSE);
;
// 设置Item的SeriesType, 此处为2d柱状图
m_Chart.GetPlot().GetSeriesCollection().GetItem(1).SetSeriesType(1);

m_Chart.GetPlot().GetSeriesCollection().GetItem(2).SetSecondaryAxis(FALSE);
;
m_Chart.GetPlot().GetSeriesCollection().GetItem(2).SetSeriesType(1);
```

```

m_Chart.GetPlot().GetSeriesCollection().GetItem(3).SetSecondaryAxis(FALSE)
;
m_Chart.GetPlot().GetSeriesCollection().GetItem(3).SetSeriesType(1);

m_Chart.GetPlot().GetSeriesCollection().GetItem(4).SetSecondaryAxis(TRUE);
// 设置Item的SeriesType, 此处为2d折线图
m_Chart.GetPlot().GetSeriesCollection().GetItem(4).SetSeriesType(6);
m_Chart.GetPlot().GetSeriesCollection().GetItem(5).SetSecondaryAxis(TRUE);
m_Chart.GetPlot().GetSeriesCollection().GetItem(5).SetSeriesType(6);
m_Chart.GetPlot().GetSeriesCollection().GetItem(6).SetSecondaryAxis(TRUE);
m_Chart.GetPlot().GetSeriesCollection().GetItem(6).SetSeriesType(6);

// 数据点类型显示数据值的模式(对柱柱状图和点线图有效)
// 0: 不显示   1: 显示在柱状图外
// 2: 显示在柱状图内上方   3: 显示在柱状图内中间   4: 显示在柱状图内
下方
    m_Chart.GetPlot().GetSeriesCollection().GetItem(1).GetDataPoints().GetItem(-1)
).GetDataPointLabel().SetLocationType(1);
    m_Chart.GetPlot().GetSeriesCollection().GetItem(2).GetDataPoints().GetItem(-1)
).GetDataPointLabel().SetLocationType(1);
    m_Chart.GetPlot().GetSeriesCollection().GetItem(3).GetDataPoints().GetItem(-1)
).GetDataPointLabel().SetLocationType(1);
    m_Chart.GetPlot().GetSeriesCollection().GetItem(4).GetDataPoints().GetItem(-1)
).GetDataPointLabel().SetLocationType(1);
    m_Chart.GetPlot().GetSeriesCollection().GetItem(5).GetDataPoints().GetItem(-1)
).GetDataPointLabel().SetLocationType(1);
    m_Chart.GetPlot().GetSeriesCollection().GetItem(6).GetDataPoints().GetItem(-1)
).GetDataPointLabel().SetLocationType(1);
}

void DemoView::DrawChart()
{
    int nRowCount = 6;
    m_Chart.SetRowCount(nRowCount);

    VARIANT var;
    // 不自动标注X轴
    m_Chart.GetPlot().GetAxis(0,var).GetCategoryScale().SetAuto(FALSE);
    // 每刻度一个标注
    m_Chart.GetPlot().GetAxis(0,var).GetCategoryScale().SetDivisionsPerLabel(1);
    // 每刻度一个刻度线
    m_Chart.GetPlot().GetAxis(0,var).GetCategoryScale().SetDivisionsPerTick(1);

```



```
// X轴名称
m_Chart.GetPlot().GetAxis(0,var).GetAxisTitle().SetText("日期");
char buf[32];
srand( (unsigned)time( NULL ) );

for(int row = 1; row <= nRowCount; ++row)
{
    m_Chart.SetRow(row);
    sprintf(buf, "%d号?", row);

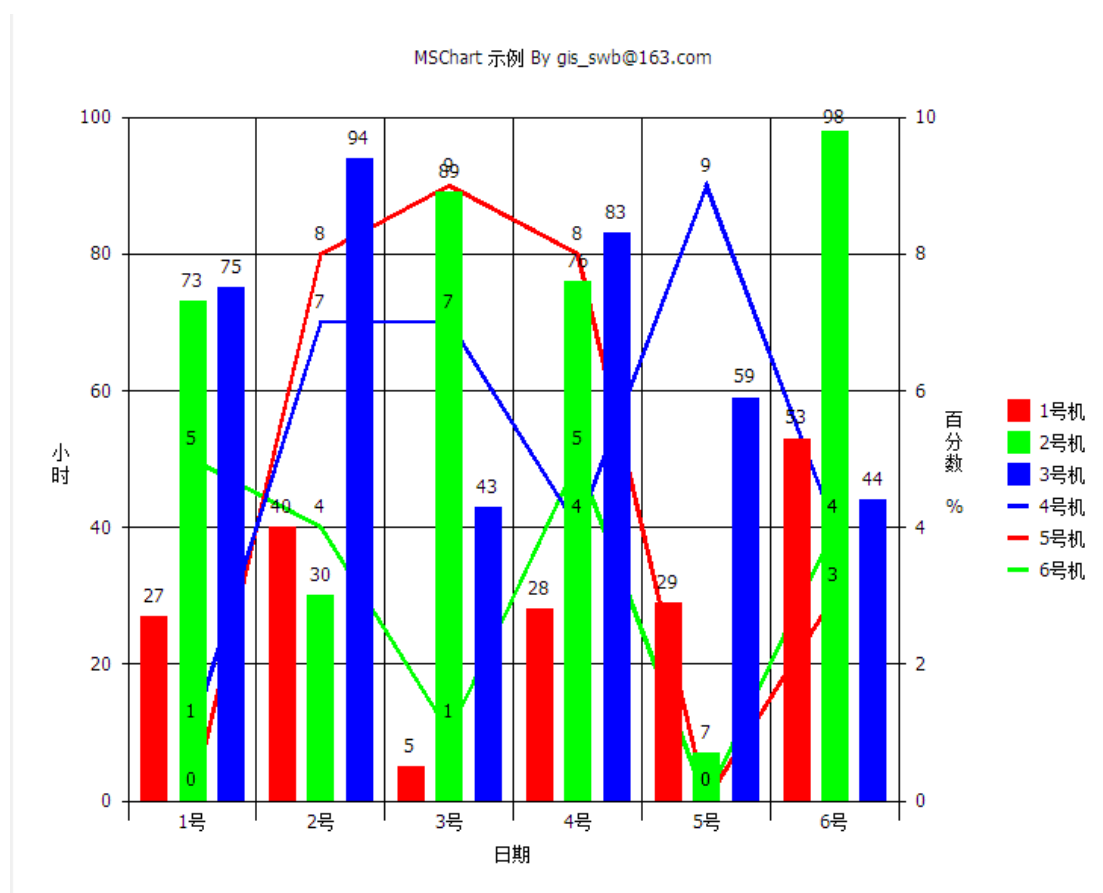
    m_Chart.SetRowLabel((LPCTSTR)buf);
    m_Chart.GetDataGrid().SetData(row, 1, rand() * 100 / RAND_MAX, 0);
    m_Chart.GetDataGrid().SetData(row, 2, rand() * 100 / RAND_MAX, 0);
    m_Chart.GetDataGrid().SetData(row, 3, rand() * 100 / RAND_MAX, 0);
    m_Chart.GetDataGrid().SetData(row, 4, rand() * 10 / RAND_MAX, 0);
    m_Chart.GetDataGrid().SetData(row, 5, rand() * 10 / RAND_MAX, 0);
    m_Chart.GetDataGrid().SetData(row, 6, rand() * 10 / RAND_MAX, 0);
}

m_Chart.Refresh();
}
```

5.代码说明

这段代码是实现在同一个 Chart 中绘制 2D 柱状图和 2D 折线图，并且柱状图使用左边的 Y 轴，折线图使用右侧的 Y 轴，此代码是在网上一个关于 MSChart 示例的基础上修改的，蓝色代码为修改和新增代码。

6.结果展示



7.相关下载

MSChart.exe: <http://pan.baidu.com/share/link?shareid=100879&uk=1798273438>

MSCHART20.OCX: <http://pan.baidu.com/share/link?shareid=100882&uk=1798273438>

Mschart Demo: <http://pan.baidu.com/share/link?shareid=100881&uk=1798273438>