MFC绘图工具High-speed Charting Control（VS2019）

前言

一、High-speed Charting Control

1.1 下载

1.2 添加到项目中

二、绘图测试

2.1 添加控件

2.2 编写程序

2.3 测试程序

三、困惑咨询

总结

前言

由于本人项目需要用到绘制实时柱状图，在网上看主要的博客都是介绍TeeChart。无意间发现了一个更适合的工具：High-speed Charting Control，简单做一下分享

提示：以下是本篇文章正文内容，下面案例可供参考

一、High-speed Charting Control

1.1 下载

在CSDN上有人分享过资源，直接可以查到。如果萌新没有积分的话可以在底下留言，我通过云盘的方式分享

1.2 添加到项目中

首先将资源中的Chart\_source复制到工程文件夹中（可能不同的资源里面名字不一样，这个文件就是该工具分装好的头文件和源文件）

然后在工程中：项目-添加现有项-Chart\_source文件夹全选添加

至此，已经完成了将资源添加进项目工程中

二、绘图测试

2.1 添加控件

High-speed Charting Control是基于Custom Control控件绘制，添加该控件后需要修改属性，主要有两处

（1）样式

  将默认的0x50010000修改为0x52010000

（2）类

  修改为ChartCtrl

2.2 编写程序

（1）在项目头文件（xxxDlg.h）中添加需要使用的库

#include "ChartCtrl\_source\ChartCtrl.h" //曲线基本设置头文件

#include "ChartCtrl\_source\ChartTitle.h" //曲线标题设置头文件

#include "ChartCtrl\_source\ChartBarSerie.h" //柱状图头文件

// MyChartDlg.h: 头文件

//

（2）在项目源文件（xxxDlg.cpp）中添加变量并绑定控件

  注意绑定控件要在对话框类下的DoDataExchange中，不然无法显示

CChartCtrl m\_ChartCtrl; //这里的成员变量名称根据实际需要命名即可

void CMyChartDlg::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Control(pDX, IDC\_CUSTOM1, m\_ChartCtrl);

}

（3）初始化坐标轴以及标题

BOOL CMyChartDlg::OnInitDialog()

{

CDialogEx::OnInitDialog();

// 将“关于...”菜单项添加到系统菜单中。

// IDM\_ABOUTBOX 必须在系统命令范围内。

ASSERT((IDM\_ABOUTBOX & 0xFFF0) == IDM\_ABOUTBOX);

ASSERT(IDM\_ABOUTBOX < 0xF000);

CMenu\* pSysMenu = GetSystemMenu(FALSE);

if (pSysMenu != nullptr)

{

BOOL bNameValid;

CString strAboutMenu;

bNameValid = strAboutMenu.LoadString(IDS\_ABOUTBOX);

ASSERT(bNameValid);

if (!strAboutMenu.IsEmpty())

{

pSysMenu->AppendMenu(MF\_SEPARATOR);

pSysMenu->AppendMenu(MF\_STRING, IDM\_ABOUTBOX, strAboutMenu);

}

}

// 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动

// 执行此操作

SetIcon(m\_hIcon, TRUE); // 设置大图标

SetIcon(m\_hIcon, FALSE); // 设置小图标

// TODO: 在此添加额外的初始化代码

//

//初始化坐标轴

CChartAxis\* pAxis = NULL;

pAxis = m\_ChartCtrl.CreateStandardAxis(CChartCtrl::BottomAxis); //底部坐标轴

//pAxis->SetAutomatic(true);

//根据具体数值自动调节坐标轴坐标，如果要固定坐标轴需要将其参数修改为false

pAxis->SetAutomatic(false);

pAxis->SetMinMax(0, 20); //固定x轴坐标为0-20

pAxis = m\_ChartCtrl.CreateStandardAxis(CChartCtrl::LeftAxis); //左边坐标轴

pAxis->SetAutomatic(false);

pAxis->SetMinMax(0, 100); //固定y轴坐标为0-100

//添加标题

TChartString str1;

str1 = \_T("High-speed Charting画图Demo");

m\_ChartCtrl.GetTitle()->AddString(str1);

//设置颜色

m\_ChartCtrl.GetTitle()->SetColor(RGB(255, 255, 255)); //标题字体白色

m\_ChartCtrl.GetLeftAxis()->SetTextColor(RGB(255, 255, 255)); //左坐标轴白色

m\_ChartCtrl.GetBottomAxis()->SetTextColor(RGB(255, 255, 255)); //底部坐标轴白色

m\_ChartCtrl.SetBorderColor(RGB(255, 255, 255)); //边框颜色白色

m\_ChartCtrl.SetBackColor(RGB(85, 85, 85)); //背景颜色深灰色

return TRUE; // 除非将焦点设置到控件，否则返回 TRUE

}

（4）绘制曲线

添加Button控件来启动定时器，定时器到时会自动更新绘图，参数给随机数

其中添加定时器的方法为：类向导-消息-添加WM\_TIMER

//点击按钮启动定时器

void CMyChartDlg::OnBnClickedTimerStart()

{

// TODO: 在此添加控件通知处理程序代码

SetTimer(1, 1000, NULL); //启动定时器1，计时1s

}

//定时器到时自动执行

void CMyChartDlg::OnTimer(UINT\_PTR nIDEvent)

{

// TODO: 在此添加消息处理程序代码和/或调用默认值

switch (nIDEvent)

{

case 1: //定时器1到时

{

srand((unsigned int)time(NULL)); //产生随机数种子，包含头文件#include<ctime>

double X1Values[15], Y1Values[15];

for (int i = 0; i < 15; i++)

{

X1Values[i] = i;

Y1Values[i] = rand() % 100 + 1; //y坐标随机产生1~100

}

CChartBarSerie\* pBarSerie3; //定义柱状图句柄

m\_ChartCtrl.SetZoomEnabled(true);

m\_ChartCtrl.RemoveAllSeries();//先清空

pBarSerie3 = m\_ChartCtrl.CreateBarSerie(); //绑定绘图变量

pBarSerie3->SetSeriesOrdering(poNoOrdering);//设置为无序

pBarSerie3->SetPoints(X1Values, Y1Values, 15);

}

break;

default:

break;

}

CDialogEx::OnTimer(nIDEvent);

}

2.3 测试程序

（1）执行时会报错

  这是因为VS2019做了优化，标准头文件从stdafx.h变为了pch.h，将High-speed Charting Control中源文件的库文件修改过来即可

（2）运行效果

三、困惑咨询

该控件的简单使用已经分享完毕了，现在有一个小困惑，恳请各位大佬能帮忙解答一下，每次Debug启动项目的时候，会弹出一个报警

点击忽略后，程序可以正常运行，不知道如何解决该问题

总结

本例只分享了柱状图的绘制，其他种类的图都可以仿照绘制，有问题可以留言交流~

————————————————

版权声明：本文为CSDN博主「爱学习的周Sir」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/puzzleZHOU/article/details/117730586