rtChart User Guide

r2 Scientific

July 2021

Contents

Getting Started
This guide and the accompanying videos will have you up and running in no time at all. Purchase rtChart today and start creating highly visual products within an hour.

To get started you will need the Windows SDK which is free from Microsoft. Download it from:

https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk/

The SDK has header files you will need to use rtChart.

All the code in this guide is from the sample program described in the Show Me video. The entire dialog code and rtChart header can be downloaded from the web site.

About Colors
Most charts have 32 colors. The first 16 are defined as pleasant chart colors and the last 16 are all defined as red. All 32 values can be set by chart with the SetColor method available on most charts.

The chart's background color on most charts (not CMeterChart) can be changed to any value with the SetWindowBackgroundColor method.

Setting Up the Framework

Add the rtChart.h header file and specify the framework. Load the library from your location.

```
#pragma once
#include "rtChart.h"
using namespace rtChartFramework;
// Use your own location for the library
#pragma comment (lib,"c://My Location//rtChart.lib")
```

Add pointers to the charts your application will use, in your class.

```
        private:
        // Create the pointers to the library objects
        CLineChart* pcLineChart;

        CLegend* pcLegend;
        CLegend* pcLegend2;

        CBarChart* pcBarChart;
        CBarChart* pcBarChart2;

        CPieChart* pcPieChart;
        CPieChart* pcPieChart2;
        CPieChart* pcPieChart3;
        CPieChart* pcPieChart4;
        CPieChart* pcPieChart5;
        CPieChart* pcPieChart6;

        CMeterChart* pcMeterChart;
        CMeterChart* pcMeterChart2;
        CMeterChart* pcMeterChart3;

        CMultiMeterChart* pcMultiMeterChart;
        CMultiMeterChart* pcMultiMeterChart2;

        CProgressBar* pcProgressBar;
```

In the programs .cpp file add ON_WM_INITIALIZECHART to the message map or catch WM_INITIALIZECHART in your winmain.

```
BEGIN_MESSAGE_MAP(Cr2ChartTestDlg, CDialog)
        ON_WM_PAINT()
        ON_WM_CREATE()
        ON_WM_TIMER()
        ON_WM_INITIALIZECCHART()
END_MESSAGE_MAP()
```

After you have created the charts (see next section) you need to push the WM_INITIALIZECHART into the message loop with:

```
PostMessage(WM_INITIALIZECCHART);
```

This will call OnInitializeChart defined as:

```
void OnInitializeChart(UINT nEvent, UINT wParam, LPARAM lParam);
```

This method can be used to set various chart options. All these commands are optional.

Example:

```
void Cr2ChartTestDlg::OnInitializeChart(UINT nEvent, UINT wParam, LPARAM lParam)
{
	TRACE(_T("Cr2ChartTestDlg::OnInitializeChart\n"));
	// Outline first multimeter window
	pcMultiMeterChart->ShowLocation(255, 0, 0);

	// Set some chart background colors
	pcBarChart->SetWindowBackgroundColor(163, 228, 215);
	pcLegend->SetWindowBackgroundColor(163, 228, 215);
	pcBarChart2->SetWindowBackgroundColor(252, 243, 207);
	pcLineChart->SetWindowBackgroundColor(252, 243, 207);
	pcLegend2->SetWindowBackgroundColor(252, 243, 207);
	pcMultiMeterChart2->SetWindowBackgroundColor(221, 153, 119);

	// Set a custom color on the first bar chart
	pcBarChart->SetColor(16, 255, 255, 0, 0);

	// Move the first bar chart
	pcBarChart->MoveChart(20, 300);

	// Fire up the timer
	nOneSecondTimer = SetTimer(1, 500, NULL);
}
```

The charts may be created in OnCreate or OnInitDialog for dialog boxes.

CLineChart

Constructor:
CLineChart(int Elements, int HorizontalStep, int HighValue, CRect cRect, int FontSize, CWnd Parent);*

Where:
Elements = The number of items to be charted (lines)
HorizontalStep = Number of pixels advanced as data points are added to the chart
HighValue = The highest value the chart will handle. Should you supply a value higher than HighValue the chart will draw a red circle at the current point. This will change back when the next data point is added to that element.
cRect = The rectangle where you want to place the line chart. Please note that the library may change the size of the chart if required.
FontSize = The size of the font used by the chart.
Parent = The CWnd of the current window.

Example:
```
CRect cRect = { 5, 5, 300, 150 };
pcLineChart = new CLineChart( 6, 20, 1000, cRect, 10, this);
```

Other CLineChart Methods

bool ShowLocation(byte byteRed, byte byteGreen, byte byteBlue);
ShowLocation draws a box around the chart window using the rgb value in the params.

Example:
```
pcLineChart->ShowLocation(255, 0, 0); // Draws a red box around the chart's window
```

bool SetColor(int nIndex, BYTE byteAlpha, BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
SetColor sets one of the 32 colors available to your specification specified by the params.

Example:
```
pcLineChart->SetColor(16, 255, 255, 0, 0); // Sets color 16 to red
```

bool SetWindowBackgroundColor(BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
Sets the chart background color to the RGB value of the params.

Example:
```
pcLineChart->SetWindowBackgroundColor(252, 243, 207);
```

bool MoveChart(int nX, int nY);
Move the chart to a new location.

Example:
```
pcLineChart->MoveChart(20, 300);
```

Called from anywhere after OnInitializeChart is called

void PlotChartValue(int nIndex, REAL dValue);
PlotChartValue draws the next datapoint on the chart. nIndex = the current element and dValus is the value to plot. REAL = float.

Example:

```
int n = rand() % 1200;
pcLineChart->PlotChartValue(i, (REAL)n);
```

CLegend

Constructor:
*CLegend(int Elements, int Orientation, int Format, int FontSize, CRect Rect, wchar_t** Labels, CWnd* Parent)*

Where:

Elements	= Number of items in the legend
Orientation	= HORIZONTAL or VERTICAL
Format	= CIRCLES or SQUARES
FontSize	= The size of the font used by the chart.
Rect	= The rectangle where you want to place the line chart. Please note that the library may change the size of the chart if required.
Labels	= Names of the elements
Parent	= The CWnd of the current window.

Example:
```
CRect cRect2 = { 5, 155, 296, 170 };
wchar_t* awszLegends[6] = { L"HP Server", L"Gateway", L"PLS", L"Graphics", L"Dell", L"Asus" };
pcLegend = new CLegend(6, HORIZONTAL, CIRCLES, 10, cRect2, awszLegends, this);
```

Other CLegend Methods

bool ShowLocation(byte byteRed, byte byteGreen, byte byteBlue);
ShowLocation draws a box around the chart window using the rgb value in the params.

Example:
```
pcLegend ->ShowLocation(255, 0, 0); // Draws a red box around the chart's window
```

bool SetColor(int nIndex, BYTE byteAlpha, BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
SetColor sets one of the 32 colors available to your specification specified by the params.

Example:
```
pcLegend ->SetColor(16, 255, 255, 0, 0); // Sets color 16 to red
```

bool SetWindowBackgroundColor(BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
Sets the chart background color to the RGB value of the params.

Example:
```
pcLegend ->SetWindowBackgroundColor(252, 243, 207);
```

bool MoveChart(int nX, int nY);
Move the chart to a new location.

Example:
```
pcLegend ->MoveChart(20, 300);
```

CBarChart
Constructor:
*CBarChart(int Elements, int Orientation, int HighValue, int FontSize, CRect cRect, wchar_t** Labels, CWnd* Parent);*

Where:

Elements	= The number of items to be charted (bars)
Orientation	= HORIZONTAL or VERTICAL
HighValue	= The highest value the chart will handle. Should you supply a value higher than HighValue the chart will draw a red border on the current bar. This will change back when the next data point is added to that element.
FontSize	= The size of the font used by the chart.
cRect	= The rectangle where you want to place the chart. Please note that the library may change the size of the chart if required.
Labels	= Names of the elements
Parent	= The CWnd of the current window.

Example:

```
cRect2 = { 5, 180, 370, 300 };
wchar_t* awszLabels[6] = { L"HP Server", L"Gateway", L"PLS", L"Graphics", L"Dell", L"Asus" };
pcBarChart = new CBarChart(6, HORIZONTAL, 500, 10, cRect2, awszLabels, this);
```

Other CBarChart Methods

bool ShowLocation(byte byteRed, byte byteGreen, byte byteBlue);
ShowLocation draws a box around the chart window using the rgb value in the params.

Example:
```
pcBarChart ->ShowLocation(255, 0, 0); // Draws a red box around the chart's window
```

bool SetColor(int nIndex, BYTE byteAlpha, BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
SetColor sets one of the 32 colors available to your specification specified by the params.

Example:
```
pcBarChart->SetColor(16, 255, 255, 0, 0); // Sets color 16 to red
```

bool SetWindowBackgroundColor(BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
Sets the chart background color to the RGB value of the params.

Example:
```
pcBarChart ->SetWindowBackgroundColor(252, 243, 207);
```

```
bool MoveChart(int nX, int nY);
```
Move the chart to a new location.

Example:
```
pcBarChart ->MoveChart(20, 300);
```

Called from anywhere after OnInitializeChart is called

```
void PlotChartValue(int nIndex, REAL dValue);
```
PlotChartValue draws the next datapoint on the chart. nIndex = the current element and dValue is the value to plot. REAL = float.

Example:
```
int n = rand() % 1200;
pcBarChart ->PlotChartValue(i, (REAL)n);
```

CPieChart

Constructor:
CPieChart(CRect cRect, CWnd pP);*

Where:
cRect = The rectangle where you want to place the chart. Please note that the library may change the size of the chart if required.
Parent = The CWnd of the current window.

Example:
```
cRect2 = { 380, 30, 480, 130 };
pcPieChart = new CPieChart(cRect2, this);
```

Other Chart Methods

bool ShowLocation(byte byteRed, byte byteGreen, byte byteBlue);
ShowLocation draws a box around the chart window using the rgb value in the params.

Example:
```
pcPieChart ->ShowLocation(255, 0, 0); // Draws a red box around the chart's window
```

bool SetColor(int nIndex, BYTE byteAlpha, BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
SetColor sets one of the 32 colors available to your specification specified by the params.

Example:
```
pcPieChart ->SetColor(16, 255, 255, 0, 0); // Sets color 16 to red
```

bool SetWindowBackgroundColor(BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
Sets the chart background color to the RGB value of the params.

Example:
```
pcPieChart ->SetWindowBackgroundColor(252, 243, 207);
```

bool MoveChart(int nX, int nY);
Move the chart to a new location.

Example:
```
pcPieChart ->MoveChart(20, 300);
```

Called from anywhere after OnInitializeChart is called

void UpdatePieChart(int Elements, int Data[]);
UpdatePieChart draws the pie chart. Data[] = The value of each slice. Elements can be different for each update

Example:

```
anData[0] = rand() % 100;
anData[1] = rand() % 100;
anData[2] = rand() % 100;
anData[3] = rand() % 100;
anData[4] = rand() % 100;
anData[5] = rand() % 100;
pcPieChart->UpdatePieChart(6, anData);
```

CMeterChart

Constructor
CMeterChart(int HighValue, CRect cRect, int IDB_Pointer, CWnd* Parent);

Where:

HighValue	= The highest value the chart will handle.
cRect	= The rectangle where you want to place the chart. Please note that the library may change the size of the chart if required.
IDB_Pointer	= A reference ID for Pointer.bmp. Libraries do not contain resources. They rely on the calling program for resources. Load Pointer.bmp as a bitmap resource and pass its ID as IDB_Pointer.
Parent	= The CWnd of the current window.

Example:
```
cRect2 = { 380, 250, 580, 400 };
pcMeterChart = new CMeterChart(100, cRect2, IDB_BITMAP1, this);
```

Other Chart Methods

bool ShowLocation(byte byteRed, byte byteGreen, byte byteBlue);
ShowLocation draws a box around the chart window using the rgb value in the params.

Example:
```
pcMeterChart ->ShowLocation(255, 0, 0); // Draws a red box around the chart's window
```

bool MoveChart(int nX, int nY);
Move the chart to a new location.

Example:
```
pcMeterChart ->MoveChart(20, 300);
```

Called from anywhere after OnInitializeChart is called

void SetMeter(double dValue);
Set the pointer to this value.

Example:
```
int n = rand() % 1200;
pcMeterChart ->SetMeter((double)n);
```

CMultiMeter

Constructor
CMultiMeterChart(int Elements, int BarWidth, int HighValue, CPoint CenterPoint, int Radius, CWnd* Parent)

Where:

Elements	= The number of items to be charted (bars). Must be less than 13.
BarWidth	= The width of the chart bars. You will have to experiment with your chart to get the best results.
HighValue	= The highest value the chart will handle
CenterPoint	= The pivot point for bars to rotate.
Radius	= Outside radius for the tick lines. The library will try its best to pull this all together the way you wish. It may change the radius or pivot point for best results.
Parent	= The CWnd of the current window.

Example:
```
CPoint cPoint = { 300, 550 };
pcMultiMeterChart = new CMultiMeterChart(5, 4, 100, cPoint, 150, this);
```

Other Chart Methods

bool ShowLocation(byte byteRed, byte byteGreen, byte byteBlue);
ShowLocation draws a box around the chart window using the rgb value in the params.

Example:
```
pcMultiMeterChart ->ShowLocation(255, 0, 0); // Draws a red box around the chart's window
```

bool SetColor(int nIndex, BYTE byteAlpha, BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
SetColor sets one of the 32 colors available to your specification specified by the params.

Example:
```
pcMultiMeterChart ->SetColor(16, 255, 255, 0, 0); // Sets color 16 to red
```

bool SetWindowBackgroundColor(BYTE byteRed, BYTE byteGreen, BYTE byteBlue);
Sets the chart background color to the RGB value of the params.

Example:
```
pcMultiMeterChart ->SetWindowBackgroundColor(252, 243, 207);
```

bool MoveChart(int nX, int nY);
Move the chart to a new location.

Example:
```
pcMultiMeterChart ->MoveChart(20, 300);
```

Called from anywhere after OnInitializeChart is called

void SetMultiMeter(int nElement, double dValue);
SetMultiMeter draws the next datapoint on the chart. nIndex = the current element and dValue is the value to plot.

Example:
```
int n = rand() % 1200;
pcMultiMeterChart -> SetMultiMeter (2, (double)n);
```

CProgressBar

Constructor:
```
CProgressBar(CRect Rect, CWnd* Parent);
```

Where:
cRect = The rectangle where you want to place the bar. Please note that the library may change the size of the bar if required.
Parent = The CWnd of the current window.

Example:
```
cRect2 = { 5, 575, 400, 581 };
pcProgressBar = new CProgressBar(cRect2, this);
```

Called from anywhere after OnInitializeChart is called

```
void SetProgressBar(int nValue);
```
SetProgressBar Sets the width to nValue. (0 to ?, depends on size of cRect2)

Example:
```
pcProgressBar->SetProgressBar(nProgress);
```

```
// rtChart TestDlg.h : header file


#pragma once

#include "rtChart.h"

using namespace rtChartFramework;

// Use your own location for the library
#pragma comment (lib,"c://Projects//rtChart//x64//Debug//rtChart.lib")

// Cr2ChartTestDlg dialog
class Cr2ChartTestDlg : public CDialog
{
// Construction
public:
	Cr2ChartTestDlg(CWnd* pParent = NULL);	// standard constructor
	~Cr2ChartTestDlg();
// Dialog Data
#ifdef AFX_DESIGN_TIME
	enum { IDD = IDD_R2CHARTTEST_DIALOG };
#endif

	protected:
	// Timer stuff
	int nOneSecondTimer;
	void OnTimer(UINT_PTR nID);

	// Create the pointers to the library objects
	CLineChart* pcLineChart;

	CLegend* pcLegend;
	CLegend* pcLegend2;

	CBarChart* pcBarChart;
	CBarChart* pcBarChart2;

	CPieChart* pcPieChart;
	CPieChart* pcPieChart2;
	CPieChart* pcPieChart3;
	CPieChart* pcPieChart4;
	CPieChart* pcPieChart5;
	CPieChart* pcPieChart6;

	CMeterChart* pcMeterChart;
	CMeterChart* pcMeterChart2;
	CMeterChart* pcMeterChart3;
```

```
        CMultiMeterChart* pcMultiMeterChart;
        CMultiMeterChart* pcMultiMeterChart2;

        CProgressBar* pcProgressBar;

        int nProgress = 0;
        int nElements;

// Implementation
protected:
        HICON m_hIcon;

        // Generated message map functions
        afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
        afx_msg BOOL OnInitDialog();
        afx_msg void OnPaint();

        DECLARE_MESSAGE_MAP()
public:
        void OnInitializeChart(UINT nEvent, UINT wParam, LPARAM lParam);
};
```

```
// rtChart TestDlg.cpp : implementation file
//

#include "stdafx.h"
#include "r2Chart Test.h"
#include "r2Chart TestDlg.h"
#include "afxdialogex.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#endif


Cr2ChartTestDlg::Cr2ChartTestDlg(CWnd* pParent /*=NULL*/)
	: CDialog(IDD_R2CHARTTEST_DIALOG, pParent)
{
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

Cr2ChartTestDlg::~Cr2ChartTestDlg()
{
}


BEGIN_MESSAGE_MAP(Cr2ChartTestDlg, CDialog)
	ON_WM_PAINT()
	ON_WM_CREATE()
	ON_WM_TIMER()
	ON_WM_INITIALIZECCHART()
END_MESSAGE_MAP()


// Cr2ChartTestDlg message handlers
int Cr2ChartTestDlg::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
	TRACE(_T("Cr2ChartTestDlg::OnCreate\n"));
	if (CWnd::OnCreate(lpCreateStruct) == -1)
		return -1;
	return 0;
}


BOOL Cr2ChartTestDlg::OnInitDialog()
{
	TRACE(_T("Cr2ChartTestDlg::OnInitDialog\n"));
	CDialog::OnInitDialog();

	// LineChart
```

```
CRect cRect = { 5, 5, 300, 150 };
nElements = 6;
// CLineChart(int Elements, int HorizontalStep, int HighValue, CRect cRect, int FontSize, CWnd*
Parent);
pcLineChart = new CLineChart( nElements, 20, 1000, cRect, 10, this);

// Legend
CRect cRect2 = { 5, 155, 296, 170 };
nElements = 6;
wchar_t* awszLegends[6] = { L"HP Server", L"Gateway", L"PLS", L"Graphics", L"Dell", L"Asus" };
// CLegend(int Elements, int Orientation, int Format, int FontSize, CRect Rect, wchar_t** Labels,
CWnd* Parent);
pcLegend = new CLegend(nElements, HORIZONTAL, CIRCLES, 10, cRect2, awszLegends, this);

cRect2 = { 305, 5, 370, 140 };
pcLegend2 = new CLegend(nElements, VERTICAL, SQUARES, 10, cRect2, awszLegends, this);

// Bar Chart
cRect2 = { 5, 180, 370, 300 };
nElements = 6;
wchar_t* awszLabels[6] = { L"HP Server", L"Gateway", L"PLS", L"Graphics", L"Dell", L"Asus" };
// CBarChart(int Elements, int Orientation, int HighValue, int FontSize, CRect cRect, wchar_t**
Labels, CWnd* Parent);
pcBarChart = new CBarChart(nElements, HORIZONTAL, 500, 10, cRect2, awszLabels, this);

cRect2 = { 5, 310, 170, 400 };
nElements = 6;
pcBarChart2 = new CBarChart(nElements, VERTICAL, 500, 10, cRect2, awszLabels, this);

// Pie Chart
cRect2 = { 380, 30, 480, 130 };
// CPieChart(CRect cRect, CWnd* pP);
pcPieChart = new CPieChart(cRect2, this);

cRect2 = { 490, 30, 590, 130 };
pcPieChart2 = new CPieChart(cRect2, this);

cRect2 = { 600, 30, 700, 130 };
pcPieChart3 = new CPieChart(cRect2, this);

cRect2 = { 380, 140, 480, 240 };
pcPieChart4 = new CPieChart(cRect2, this);

cRect2 = { 490, 140, 590, 240 };
pcPieChart5 = new CPieChart(cRect2, this);

cRect2 = { 600, 140, 700, 240 };
pcPieChart6 = new CPieChart(cRect2, this);
```

```
	// Meter Chart
	cRect2 = { 380, 250, 580, 400 };
	// CMeterChart(int HighValue, CRect cRect, int IDB_Pointer, CWnd* Parent);
	pcMeterChart = new CMeterChart(100, cRect2, IDB_BITMAP1, this);

	cRect2 = { 590, 250, 720, 400 };
	pcMeterChart2 = new CMeterChart(100, cRect2, IDB_BITMAP1, this);

	cRect2 = { 590, 340, 720, 490 };
	pcMeterChart3 = new CMeterChart(100, cRect2, IDB_BITMAP1, this);

	// MultiMeter Chart
	CPoint cPoint = { 300, 550 };
	// CMultiMeterChart(int Elements, int BarWidth, int HighValue, CPoint CenterPoint, int Radius,
	CWnd* Parent)
	pcMultiMeterChart = new CMultiMeterChart(5, 4, 100, cPoint, 150, this);

	cPoint = { 560, 570 };
	pcMultiMeterChart2 = new CMultiMeterChart(5, 3, 200, cPoint, 120, this);

	// ProgressBar
	cRect2 = { 5, 575, 400, 581 };
	// CProgressBar(CRect Rect, CWnd* Parent);
	pcProgressBar = new CProgressBar(cRect2, this);


	PostMessage(WM_INITIALIZECCHART);



	return TRUE;  // return TRUE  unless you set the focus to a control
}

void Cr2ChartTestDlg::OnPaint()
{
		CDialog::OnPaint();
}

void Cr2ChartTestDlg::OnInitializeChart(UINT nEvent, UINT wParam, LPARAM lParam)
{
	TRACE(_T("Cr2ChartTestDlg::OnInitializeChart\n"));
	// Outline first multimeter window
	pcMultiMeterChart->ShowLocation(255, 0, 0);

	// Set some chart colors
	pcBarChart->SetWindowBackgroundColor(163, 228, 215);
//	pcLegend->SetWindowBackgroundColor(163, 228, 215);
```

```
	pcBarChart2->SetWindowBackgroundColor(252, 243, 207);
	pcLineChart->SetWindowBackgroundColor(252, 243, 207);
//	pcLegend2->SetWindowBackgroundColor(252, 243, 207);
	pcMultiMeterChart2->SetWindowBackgroundColor(221, 153, 119);

	// Set a custom color on the first bar chart (not really used)
	pcBarChart->SetColor(16, 255, 255, 0, 0);

	// Move the first bar chart (also not used)
//	pcBarChart->MoveChart(20, 300);

	// Fire up the timer
	nOneSecondTimer = SetTimer(1, 500, NULL);
}

void Cr2ChartTestDlg::OnTimer(UINT_PTR nID)
{
	// Feed the charts
	int i;
	int anData[12];
	for (i = 0; i < nElements; ++i)
	{
		int n = rand() % 1200;
		pcLineChart->PlotChartValue(i, n);
		n = rand() % 600;
		pcBarChart->PlotChartValue(i, (double)n);
		pcBarChart2->PlotChartValue(i, (double)n);
	}
		anData[0] = rand() % 100;
		anData[1] = rand() % 100;
		anData[2] = rand() % 100;
		anData[3] = rand() % 100;
		anData[4] = rand() % 100;
		anData[5] = rand() % 100;
		pcPieChart->UpdatePieChart(6, anData);

		anData[0] = rand() % 100;
		anData[1] = rand() % 100;
		anData[2] = rand() % 100;
		anData[3] = rand() % 100;
		anData[4] = rand() % 100;
		anData[5] = rand() % 100;
		pcPieChart2->UpdatePieChart(6, anData);

		anData[0] = rand() % 100;
		anData[1] = rand() % 100;
		anData[2] = rand() % 100;
		anData[3] = rand() % 100;
```

```
anData[4] = rand() % 100;
anData[5] = rand() % 100;
pcPieChart3->UpdatePieChart(6, anData);

anData[0] = rand() % 100;
anData[1] = rand() % 100;
anData[2] = rand() % 100;
anData[3] = rand() % 100;
anData[4] = rand() % 100;
anData[5] = rand() % 100;
pcPieChart4->UpdatePieChart(6, anData);

anData[0] = rand() % 100;
anData[1] = rand() % 100;
anData[2] = rand() % 100;
anData[3] = rand() % 100;
anData[4] = rand() % 100;
anData[5] = rand() % 100;
pcPieChart5->UpdatePieChart(6, anData);

anData[0] = rand() % 100;
anData[1] = rand() % 100;
anData[2] = rand() % 100;
anData[3] = rand() % 100;
anData[4] = rand() % 100;
anData[5] = rand() % 100;
pcPieChart6->UpdatePieChart(6, anData);

double d = rand() % 100;
pcMeterChart->SetMeter(d);

d = rand() % 100;
pcMeterChart2->SetMeter(d);

d = rand() % 100;
pcMeterChart3->SetMeter(d);

for (i = 0; i < 5; ++i)
{
        d = rand() % 100;
        pcMultiMeterChart->SetMultiMeter(i, d);
        d = rand() % 260;
        pcMultiMeterChart2->SetMultiMeter(i, d);

}
if (nProgress > 150)
        nProgress = 0;
pcProgressBar->SetProgressBar(nProgress);
```

```
                nProgress += 6;
}
```