

Socket Programming with MFC

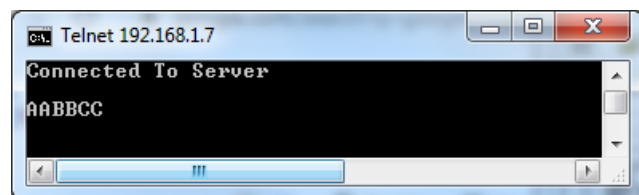
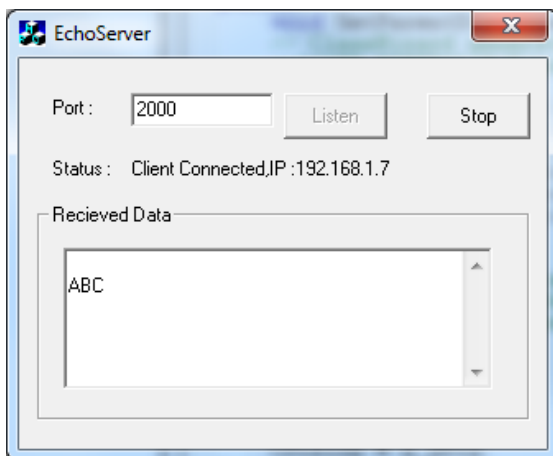
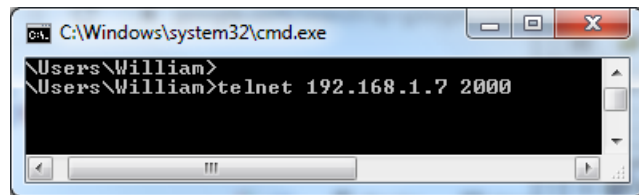
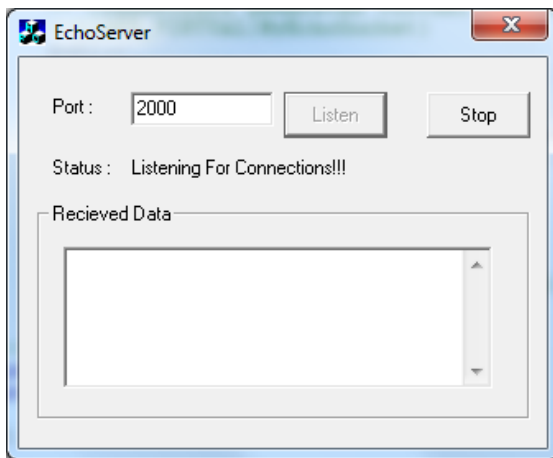
10-26-2020 William Xu

这个程序显示最简单的 socket client/server 通讯方式. 从这个最简单的程序出发我们可以开发出各种复杂的通讯程序.

这是一个用 MFC 程序 (VC ++ 6.0) 建立的 Server 程序 (EchoServer), 和它通信的 Client 程序是 Windows 7 中现成的 telnet 程序.

```
>telnet computer-IP telnet-Port <CR>
```

```
>telnet 192.168.1.7 2000 <CR>
```



This program can be downloaded from:

<https://www.codeproject.com/Articles/12209/Socket-Programming-with-MFC-Part-1>

下面描述它的工作原理

1. 用 CAsyncSocket socket class 建立自己的 MyEchoSocket class. 这里的虚拟功能全部被重写:

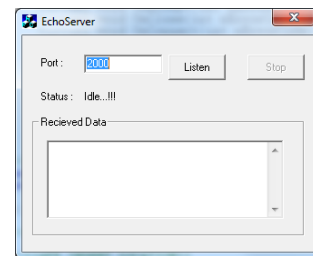
```
virtual void OnAccept(int nErrorCode);  
virtual void OnClose(int nErrorCode);  
virtual void OnConnect(int nErrorCode);  
virtual void OnOutOfBandData(int nErrorCode);  
virtual void OnReceive(int nErrorCode);  
virtual void OnSend(int nErrorCode);
```

2. 用 MyEchoSocket class 建立两个 class, 用来听 (Listen) 和连接 (Connect)
m_sListener,
m_sConnected.

因为这两个都是源于 socket class, 所以它们在听到, 收到的时候都会产生 Events.

3. 面板上有个按钮启动 (Listen) 它启动 socket m_sListener:

```
void CEchoServerDlg::OnButton1()  
{  
    UpdateData(TRUE);  
    m_sListener.Create(m_port);  
    if(m_sListener.Listen()==FALSE)  
    {  
        AfxMessageBox("Unable to Listen on that port, please try  
another port");  
        m_sListener.Close();  
        return;  
    }  
    m_status="Listening For Connections!!!";  
    UpdateData(FALSE);  
    m_listenbutton.EnableWindow(FALSE);  
    m_stopbutton.EnableWindow(TRUE);  
}
```



4. 下面一个问题要说明的是, 当 socket 的 Events 启动的时候如何通知 Dialog 主程序 (EchoServer)? 这是一个 trick. 这里面要用到一个功能称作:

Socket-Class. **SetParentDlg(this)**

`SetParentDlg()` 这个功能把 Parent Dialog 的 Instance 取下来放到 Socket Class 里. 当 Socket Class 的 Event 启动的时候就可以把它传到 Parent Dialog.

5. 在启动 Dialog 的时候要 Call 这个 `SetParentDlg()` 程序:

```
BOOL CEchoServerDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    ... ..

    // TODO: Add extra initialization here
    m_sListener.SetParentDlg(this);
    m_sConnected.SetParentDlg(this);

    ... ..
    m_status="Idle...!!!";
    m_port=2000;
    m_stopbutton.EnableWindow(FALSE);
    UpdateData(FALSE);
    return TRUE;
}
```

6. 在 Socket Class 已经有了 Parent Dialog 的 Instance, 当 Socket Event 启动的时候就可以把 Event 传到 Parent Dialog:

```
void MyEchoSocket::OnAccept(int nErrorCode)
{
    if(nErrorCode==0)
    {
        ((CEchoServerDlg*)m_pDlg)->OnAccept();
    }
    CAsyncSocket::OnAccept(nErrorCode);
}
```

注意 `m_pDlg` 是 `MyEchoSocket` 的 member; `->OnAccept()` 是 Parent Dialog 里的功能, 所以 Socket Class 的 Event 就能够传到 Parent Dialog.

If the MFC project is not able to download, we can create it by using the following source code.

How to:

- 1) Create MFC (VC++ 6.0) Dialog based program with Windows Sockets supported, named "EchoServer";
- 2) Replace your EchoServerDlg.cpp and EchoServerDlg.h two files with the same name files listed below.
- 3) Add MyEchoSocket.cpp and MyEchoSocket.h two files into "EchoServer" project.
- 4) Hit F7 to build the "EchoServer" project.
- 5) You will see the following 5 errors:
 - error C2065: 'IDC_BUTTON2' : undeclared identifier
 - error C2065: 'IDC_BUTTON1' : undeclared identifier
 - error C2065: 'IDC_EDIT1' : undeclared identifier
 - error C2065: 'IDC_EDIT2' : undeclared identifier
 - error C2065: 'IDC_STATUS' : undeclared identifier
- 6) On EchoServer dialog add two push buttons:
 - Listen - 'IDC_BUTTON1'
 - Stop - 'IDC_BUTTON2'
- 7) On EchoServer dialog add two edit boxes and one static string.
 - Edit Box(for Port) - 'IDC_EDIT1'
 - Edit Box, Multiline (for Receive Data) - 'IDC_EDIT2'
 - Static string(for status) - 'IDC_STATUS'
- 8) Hit F7 to re-build. It should be OK to generate executable program.

Source Code

```
// EchoServerDlg.cpp : implementation file
//
```

```
#include "stdafx.h"
#include "EchoServer.h"
#include "EchoServerDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
```

```

/////////////////////////////////////////////////////////////////
// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

    // Dialog Data
   //{{AFX_DATA(CAboutDlg)
    enum { IDD = IDD_ABOUTBOX };
    //}}AFX_DATA

    // ClassWizard generated virtual function overrides
   //{{AFX_VIRTUAL(CAboutDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX);  // DDX/DDV support
    //}}AFX_VIRTUAL

    // Implementation
protected:
   //{{AFX_MSG(CAboutDlg)
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
    //{{AFX_DATA_INIT(CAboutDlg)
    //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    //{{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////
// CEchoServerDlg dialog

```

```

CEchoServerDlg::CEchoServerDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CEchoServerDlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CEchoServerDlg)
    m_port = 0;
    m_recieveddata = _T("");
    m_status = _T("");
   //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CEchoServerDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CEchoServerDlg)
    DDX_Control(pDX, IDC_BUTTON2, m_stopbutton);
    DDX_Control(pDX, IDC_BUTTON1, m_listenbutton);
    DDX_Text(pDX, IDC_EDIT1, m_port);
    DDX_Text(pDX, IDC_EDIT2, m_recieveddata);
    DDX_Text(pDX, IDC_STATUS, m_status);
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CEchoServerDlg, CDialog)
   //{{AFX_MSG_MAP(CEchoServerDlg)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, OnButton1)
    ON_BN_CLICKED(IDC_BUTTON2, OnButton2)
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CEchoServerDlg message handlers

BOOL CEchoServerDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);

```

```

if (pSysMenu != NULL)
{
    CString strAboutMenu;
    strAboutMenu.LoadString(IDS_ABOUTBOX);
    if (!strAboutMenu.IsEmpty())
    {
        pSysMenu->AppendMenu(MF_SEPARATOR);
        pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
    }
}

// Set the icon for this dialog. The framework does this automatically
// when the application's main window is not a dialog
SetIcon(m_hIcon, TRUE);           // Set big icon
SetIcon(m_hIcon, FALSE);        // Set small icon

// TODO: Add extra initialization here
m_sListener.SetParentDlg(this);
m_sConnected.SetParentDlg(this);
m_status="Idle...!!!";
m_port=2000;
m_stopbutton.EnableWindow(FALSE);
UpdateData(FALSE);
return TRUE; // return TRUE unless you set the focus to a control
}

void CEchoServerDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CEchoServerDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting
    }
}

```

```

        SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CEchoServerDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

void CEchoServerDlg::OnAccept()
{
    CString strIP;
    UINT port;
    if(m_sListener.Accept(m_sConnected))
    {
        m_sConnected.GetSockName(strIP,port);
        m_status="Client Connected,IP :"+ strIP;
        m_sConnected.Send("Connected To Server",strlen("Connected To Server"));
        UpdateData(FALSE);
    }

    else
    {
        AfxMessageBox("Cannoot Accept Connection");
    }
}

void CEchoServerDlg::OnClose()
{
    AfxMessageBox("Socket Is Closed!!!");
}

```



```

        OnButton2();

    }

void CEchoServerDlg::OnReceive()
{
    char *pBuf=new char [1025];
    CString strData;
    int iLen;
    iLen=m_sConnected.Receive(pBuf,1024);
    if(iLen==SOCKET_ERROR)
    {
        AfxMessageBox("Could not Recieve");
    }
    else
    {
        pBuf[iLen]=NULL;
        strData=pBuf;
        m_recieveddata.Insert(m_recieveddata.GetLength(),strData); //display in server
        UpdateData(FALSE);
        m_sConnected.Send(pBuf,iLen);
        //m_sConnected.ShutDown(0);
        delete pBuf;
    }
}

void CEchoServerDlg::OnButton1()
{
    // TODO: Add your control notification handler code here
    UpdateData(TRUE);
    m_sListener.Create(m_port);
    if(m_sListener.Listen()==FALSE)
    {
        AfxMessageBox("Unable to Listen on that port,please try another port");
        m_sListener.Close();
        return;
    }
    m_status="Listening For Connections!!!";
    UpdateData(FALSE);
    m_listenbutton.EnableWindow(FALSE);
    m_stopbutton.EnableWindow(TRUE);
}

void CEchoServerDlg::OnButton2()
{
    // TODO: Add your control notification handler code here

```

```

m_sConnected.Close();
m_sListener.Close();
m_status="Idle!!";
UpdateData(FALSE);
m_listenbutton.EnableWindow(TRUE);
m_stopbutton.EnableWindow(FALSE);

```

```

}

```

```

// EchoServerDlg.h : header file
//

```

```

#ifndef AFX_ECHOSERVERDLG_H__91CEEDDC_5EB4_4AE7_8175_D8A85EA9CA07__INCLUDED_
#define AFX_ECHOSERVERDLG_H__91CEEDDC_5EB4_4AE7_8175_D8A85EA9CA07__INCLUDED_

```

```

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

```

```

////////////////////////////////////
// CEchoServerDlg dialog
#include "MyEchoSocket.h"

```

```

class CEchoServerDlg : public CDialog
{
// Construction
public:
    void OnReceive();
    void OnClose();
    void OnAccept();
    CEchoServerDlg(CWnd* pParent = NULL);    // standard constructor

```

```

// Dialog Data
//{{AFX_DATA(CEchoServerDlg)
enum { IDD = IDD_ECHOSERVER_DIALOG };
CButton    m_stopbutton;
CButton    m_listenbutton;
int        m_port;
CString    m_recieveddata;
CString    m_status;
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CEchoServerDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
//}}AFX_VIRTUAL

```

```

// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
   //{{AFX_MSG(CEchoServerDlg)
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    afx_msg void OnButton1();
    afx_msg void OnButton2();
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
private:
    MyEchoSocket m_sListener;
    MyEchoSocket m_sConnected;
};

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif
// !defined(AFX_ECHOSERVERDLG_H__91CEEDDC_5EB4_4AE7_8175_D8A85EA9CA07__INCLUDED_)

```

```

// MyEchoSocket.cpp : implementation file
//

#include "stdafx.h"
#include "EchoServer.h"
#include "MyEchoSocket.h"
#include "EchoServerDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// MyEchoSocket

MyEchoSocket::MyEchoSocket()
{
}

MyEchoSocket::~MyEchoSocket()

```

```

{
}

// Do not edit the following lines, which are needed by ClassWizard.
#if 0
BEGIN_MESSAGE_MAP(MyEchoSocket, CAsyncSocket)
    //{AFX_MSG_MAP(MyEchoSocket)
    //}AFX_MSG_MAP
END_MESSAGE_MAP()
#endif // 0

////////////////////////////////////
// MyEchoSocket member functions

void MyEchoSocket::OnAccept(int nErrorCode)
{
    // TODO: Add your specialized code here and/or call the base class
    if(nErrorCode==0)
    {
        ((CEchoServerDlg*)m_pDlg)->OnAccept();
    }
    CAsyncSocket::OnAccept(nErrorCode);
}

void MyEchoSocket::OnClose(int nErrorCode)
{
    // TODO: Add your specialized code here and/or call the base class
    if(nErrorCode==0)
    {
        ((CEchoServerDlg*)m_pDlg)->OnClose();
    }

    CAsyncSocket::OnClose(nErrorCode);
}

void MyEchoSocket::OnConnect(int nErrorCode)
{
    // TODO: Add your specialized code here and/or call the base class

    CAsyncSocket::OnConnect(nErrorCode);
}

void MyEchoSocket::OnOutOfBandData(int nErrorCode)
{
    // TODO: Add your specialized code here and/or call the base class

```

```

        CAsyncSocket::OnOutOfBandData(nErrorCode);
    }

void MyEchoSocket::OnReceive(int nErrorCode)
{
    // TODO: Add your specialized code here and/or call the base class
    if(nErrorCode==0)
    {
        ((CEchoServerDlg*)m_pDlg)->OnReceive();
    }

    CAsyncSocket::OnReceive(nErrorCode);
}

void MyEchoSocket::OnSend(int nErrorCode)
{
    // TODO: Add your specialized code here and/or call the base class

    CAsyncSocket::OnSend(nErrorCode);
}

void MyEchoSocket::SetParentDlg(CDialog *pDlg)
{
    m_pDlg=pDlg;
}

```

```

#ifndef AFX_MYECHOSOCKET_H__166D4120_2F94_4231_AE60_7C719E3EC05C__INCLUDED_
#define AFX_MYECHOSOCKET_H__166D4120_2F94_4231_AE60_7C719E3EC05C__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

// MyEchoSocket.h : header file
//

////////////////////

// MyEchoSocket command target

class MyEchoSocket : public CAsyncSocket
{
// Attributes
public:

// Operations
public:

```

```

        MyEchoSocket();
        virtual ~MyEchoSocket();

// Overrides
public:
    void SetParentDlg(CDialog *pDlg);
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(MyEchoSocket)
    public:
        virtual void OnAccept(int nErrorCode);
        virtual void OnClose(int nErrorCode);
        virtual void OnConnect(int nErrorCode);
        virtual void OnOutOfBandData(int nErrorCode);
        virtual void OnReceive(int nErrorCode);
        virtual void OnSend(int nErrorCode);
    }AFX_VIRTUAL

    // Generated message map functions
    //{AFX_MSG(MyEchoSocket)
        // NOTE - the ClassWizard will add and remove member functions here.
    }AFX_MSG

// Implementation
protected:
private:
    CDialog * m_pDlg;
};

////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif
// !defined(AFX_MYECHOSOCKET_H__166D4120_2F94_4231_AE60_7C719E3EC05C__INCLUDED_)



---




---



//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by EchoServer.rc
//
#define IDM_ABOUTBOX            0x0010
#define IDD_ABOUTBOX            100
#define IDS_ABOUTBOX            101
#define IDD_ECHOSERVER_DIALOG    102

```

```
#define IDR_MAINFRAME          128
#define IDC_EDIT1              1000
#define IDC_BUTTON1            1001
#define IDC_BUTTON2            1002
#define IDC_EDIT2              1003
#define IDC_STATUS              1005

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE  129
#define _APS_NEXT_COMMAND_VALUE   32771
#define _APS_NEXT_CONTROL_VALUE   1006
#define _APS_NEXT_SYMED_VALUE     101
#endif
#endif
```