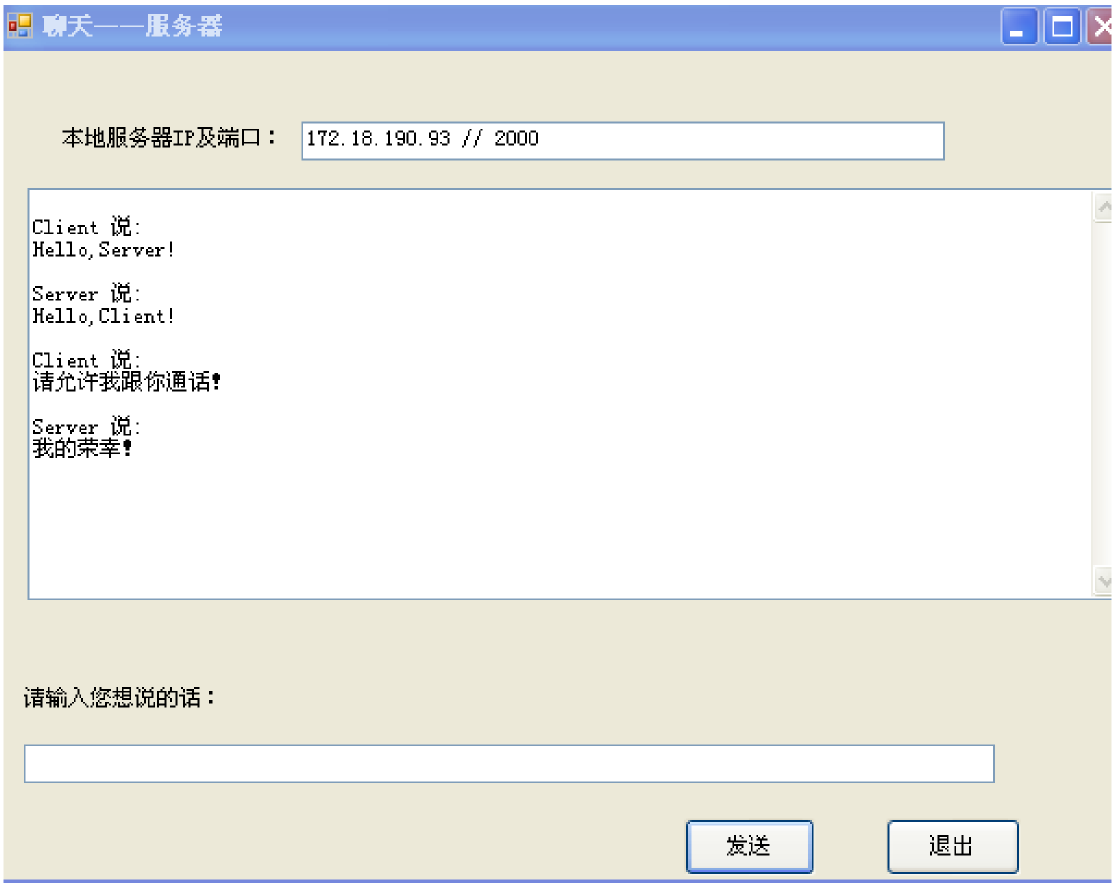
基于 Microsoft Visual Stadio 2005 C++ 环境下

采用 Winsock 编程原理实现服务器——客户端通讯的简单程序

运行整体效果

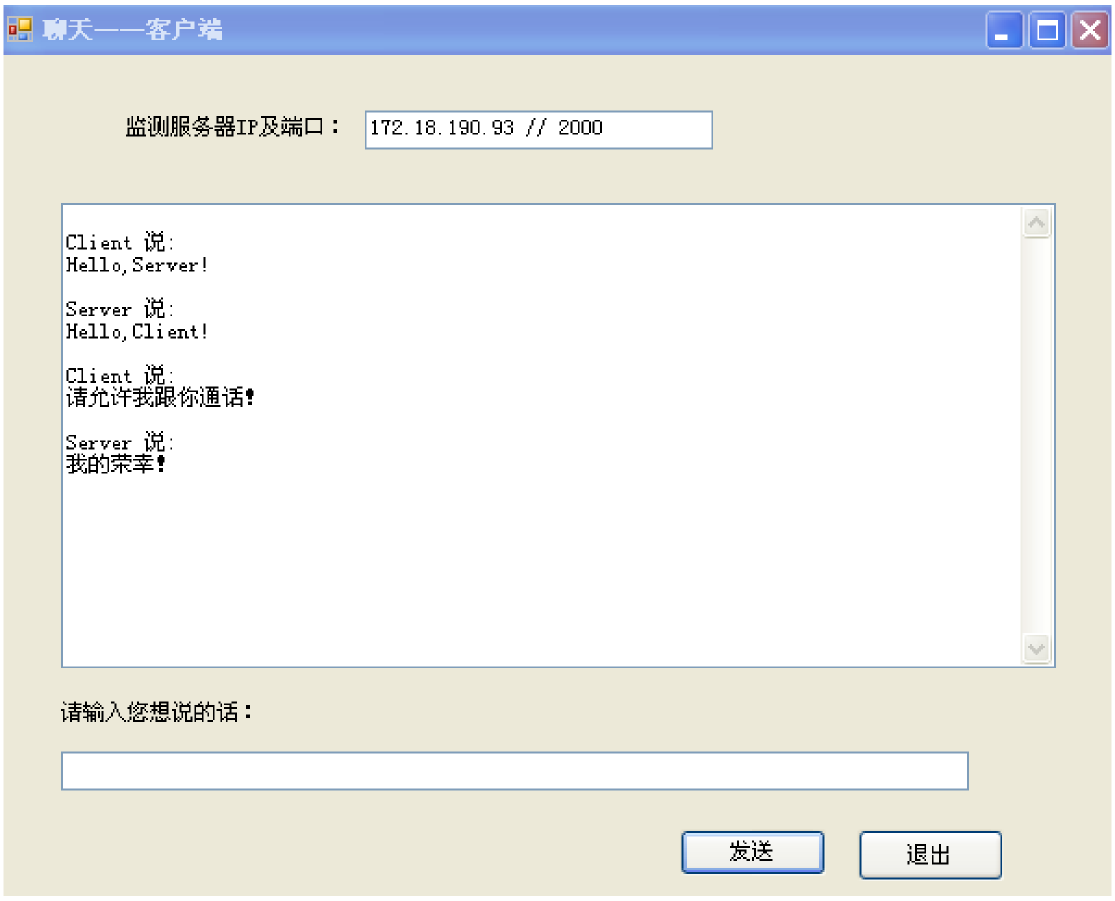


本地服务器IP及端口： 172.18.190.93 // 2000

```
Client 说:
Hello,Server!

Server 说:
Hello,Client!

Client 说:
请允许我跟你通话!

Server 说:
我的荣幸!
```

请输入您想说的话：

服务器整体图片



172.18.190.93 // 2000

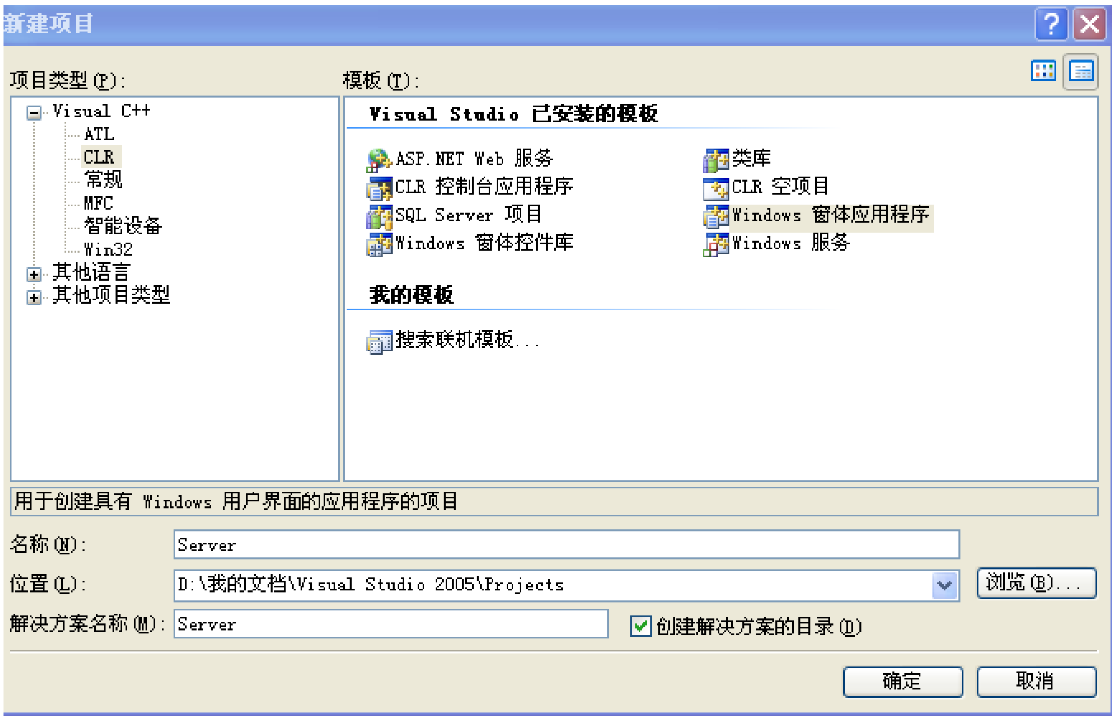
```
Client 说:
Hello,Server!

Server 说:
Hello,Client!

Client 说:
请允许我跟你通话!

Server 说:
我的荣幸!
```

客户端整体图片

服务器编写

1. 新建 Windows 窗体应用程序 Server

文件——新建——项目，进入新建项目，选 CLR、Windows 窗体应用程序，在下面名称一项输入 Server，保存位置自己定，如下图 1 所示


新建项目

项目类型(P):
- Visual C++
  - ATL
  - CLR
  - 常规
  - MFC
  - 智能设备
  - Win32
- 其他语言
- 其他项目类型

模板(T):

Visual Studio 已安装的模板

ASP.NET Web 服务	类库
CLR 控制台应用程序	CLR 空项目
SQL Server 项目	Windows 窗体应用程序
Windows 窗体控件库	Windows 服务

我的模板

搜索联机模板...

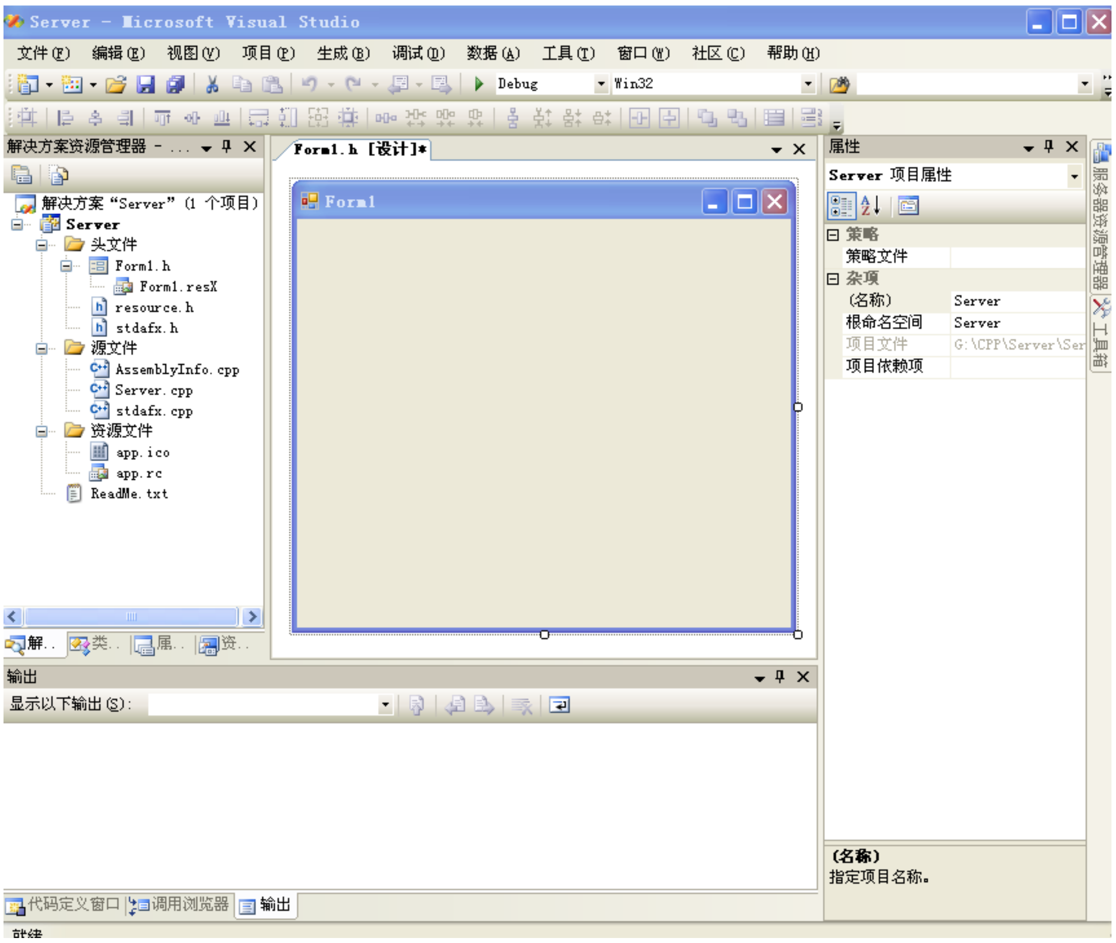
用于创建具有 Windows 用户界面的应用程序的项目

名称(N):	Server
位置(L):	D:\我的文档\Visual Studio 2005\Projects
解决方案名称(M):	Server

[✓] 创建解决方案的目录(D)

图 1

单击确定后，进入 Server 界面，如下图 2 所示


Server - Microsoft Visual Studio

文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 数据(A) 工具(T) 窗口(W) 社区(C) 帮助(H)
Debug
Win32

解决方案资源管理器 - ...

- 解决方案"Server"(1 个项目)
  - Server
    - 头文件
      - Form1.h
        - Form1.resX
      - resource.h
      - stdafx.h
    - 源文件
      - AssemblyInfo.cpp
      - Server.cpp
      - stdafx.cpp
    - 资源文件
      - app.ico
      - app.rc
    - ReadMe.txt

属性

Server 项目属性

策略	
策略文件	
杂项	
(名称)	Server
根命名空间	Server
项目文件	G:\CPP\Server\Ser
项目依赖项	

(名称)
指定项目名称。

图 2

从图 2 可以清楚地看到新建 Server 包含的一些资源，中间是开发的界面，默认为 Form1，右边是一些调协的属性！我们想要调试后得到什么样的界面效果，直接到 Form1 中进行设置即可，右边有个工具箱，自动隐藏，当然也可设置显示，如下图 3 所示

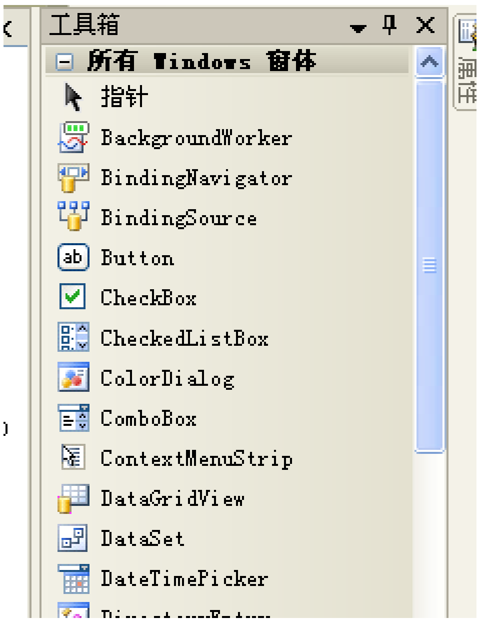
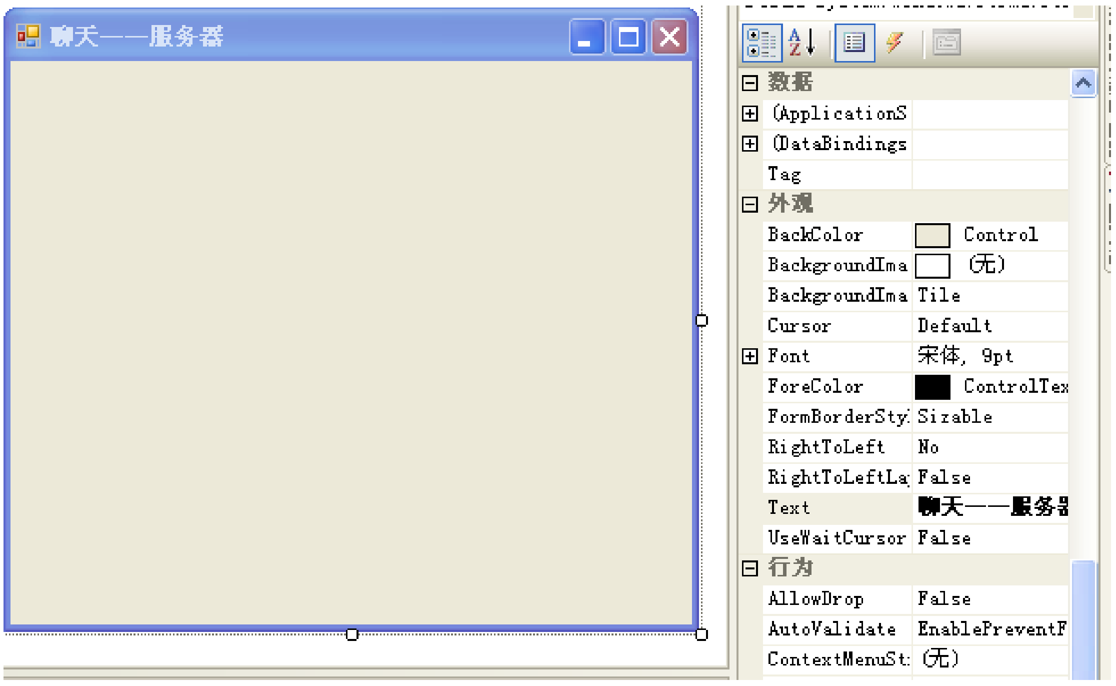


图 3

可以看到上面有很多功能选项！现在我们开始对 Form1 进行开发，将顶上“Form1”改为“聊天——服务器”，即单击 Form1 界面，在右边属性的 Text 项中的“Form1”更改为“聊天——服务器”即可，如下图 4 所示



数据	
(ApplicationS	
(DataBindings	
Tag	
外观	
BackColor	Control
BackgroundIma	(无)
BackgroundIma	Tile
Cursor	Default
Font	宋体, 9pt
ForeColor	ControlTex
FormBorderSty	Sizable
RightToLeft	No
RightToLeftLa	False
Text	聊天——服务器
UseWaitCursor	False
行为	
AllowDrop	False
AutoValidate	EnablePreventF
ContextMenuSt	(无)

图 4

设置 IP 和端口显示的窗口，选取 TextBox 选项，在界面上拉出一条窗口，如下图 5 所示

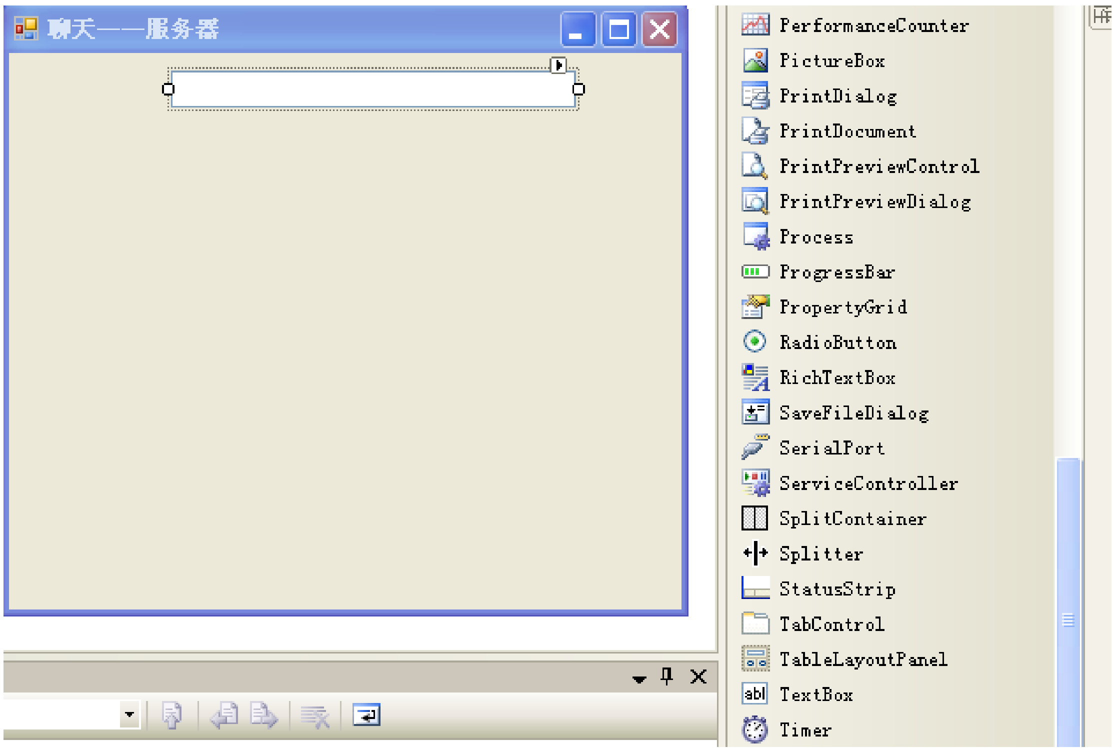
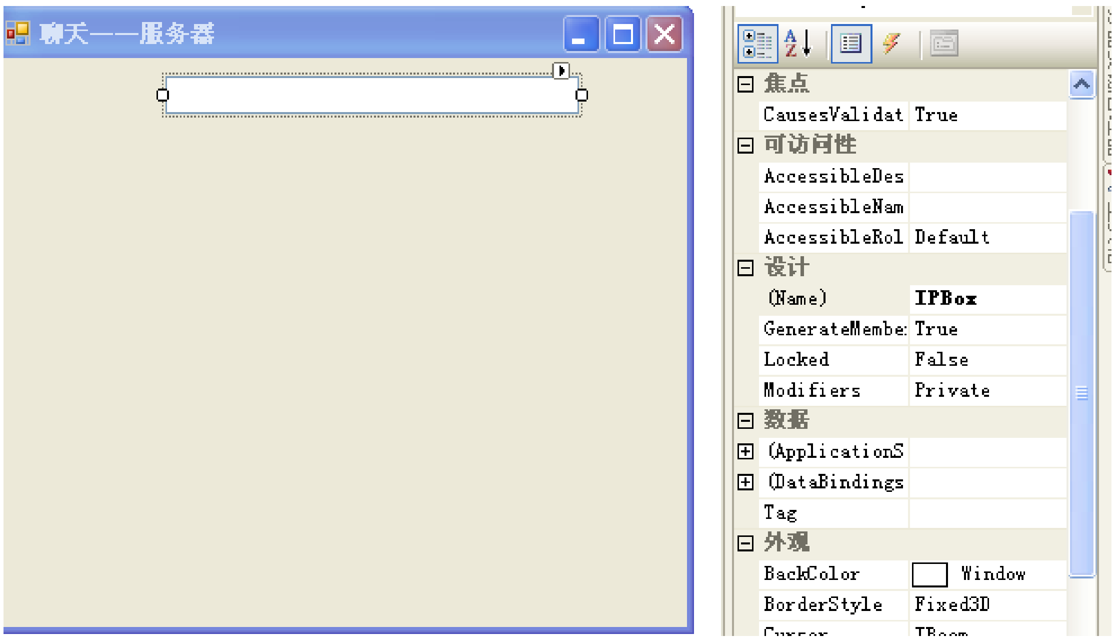


图 5

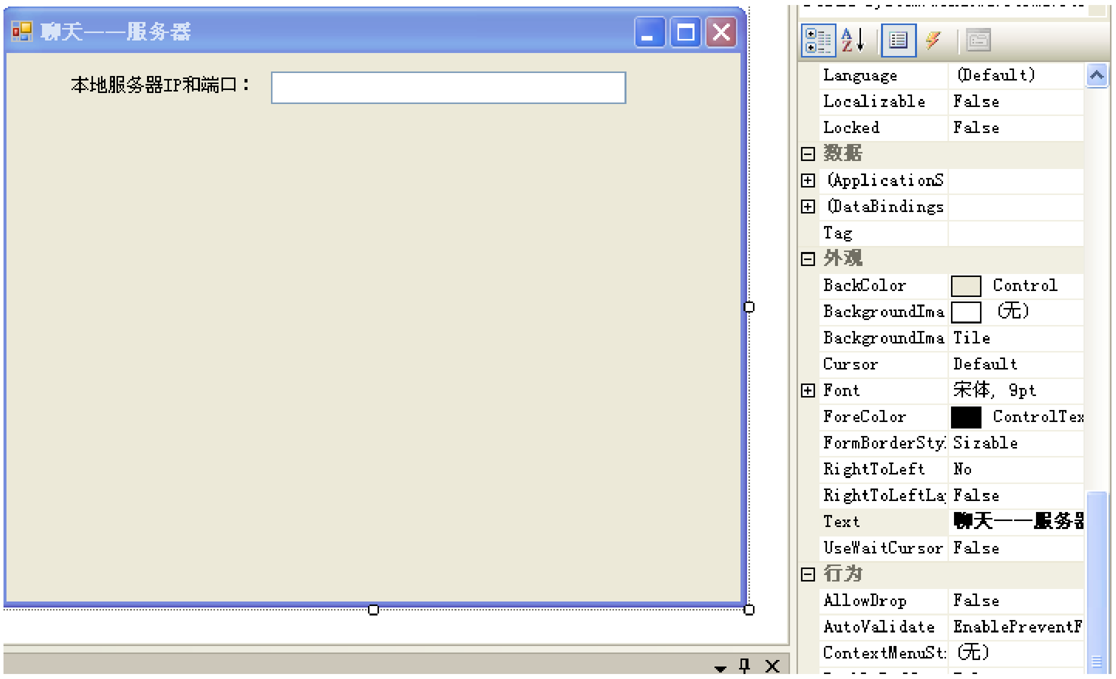
单击刚拉出的窗口，在右边属性项 Name 中将默认的 TextBox1 更改为 IPBox，如下图 6 所示



⊟ 焦点	
CausesValidat	True
⊟ 可访问性	
AccessibleDes	
AccessibleNam	
AccessibleRol	Default
⊟ 设计	
(Name)	**IPBox**
GenerateMembe:	True
Locked	False
Modifiers	Private
⊟ 数据	
⊞ (ApplicationS	
⊞ (DataBindings	
Tag	
⊟ 外观	
BackColor	▭ Window
BorderStyle	Fixed3D

图 6

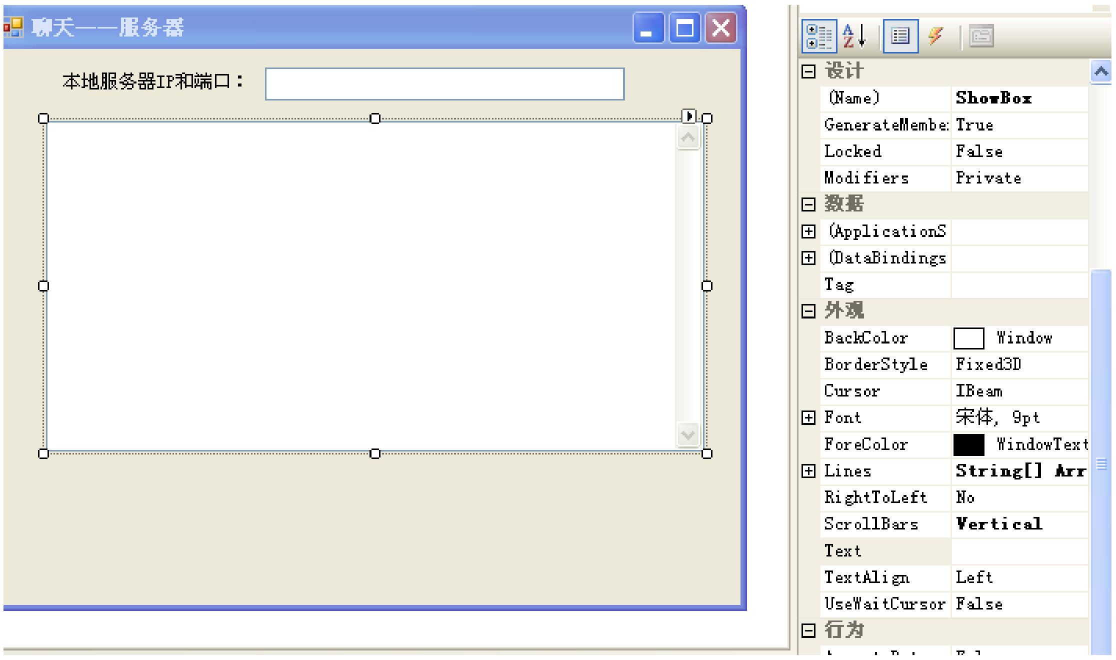
再从工具箱中选取 Label 项，在 IPBox 左边拉出一小窗口，默认文字为 Label1，点击它，在右边的属性中找到 Text 项，将默认的“Label1”更改为“本地服务器 IP 和端口：”，如下图 7 所示



Language	(Default)
Localizable	False
Locked	False
⊟ 数据	
⊞ (ApplicationS	
⊞ (DataBindings	
Tag	
⊟ 外观	
BackColor	□ Control
BackgroundIma	□ (无)
BackgroundIma	Tile
Cursor	Default
⊞ Font	宋体, 9pt
ForeColor	■ ControlTex
FormBorderSty	Sizable
RightToLeft	No
RightToLeftLa	False
Text	聊天——服务器
UseWaitCursor	False
⊟ 行为	
AllowDrop	False
AutoValidate	EnablePreventF
ContextMenuSt	(无)

图 7

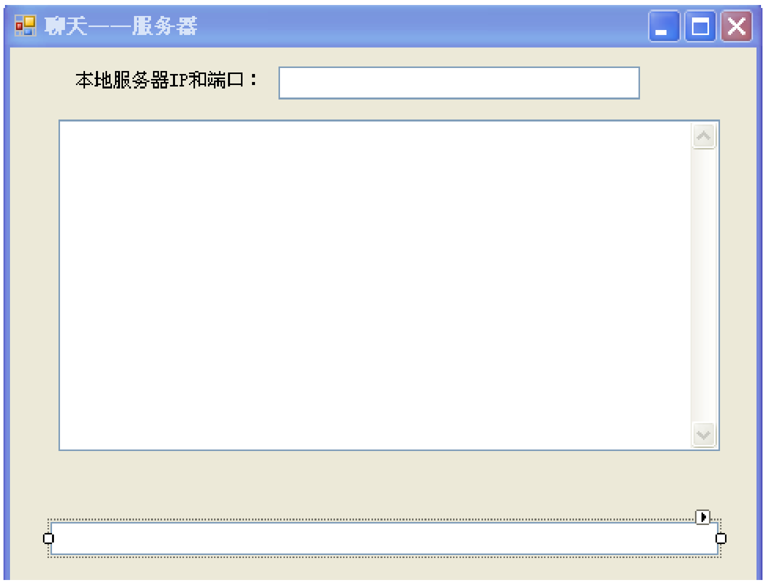
同样，选取 TextBox，拉出一个显示窗口，如下图 8 所示，在属性中设置 Name 为 ShowBox，ScrollBars 选为 Vertical（竖向输入）,Multiline 选为 True 可多项输入！

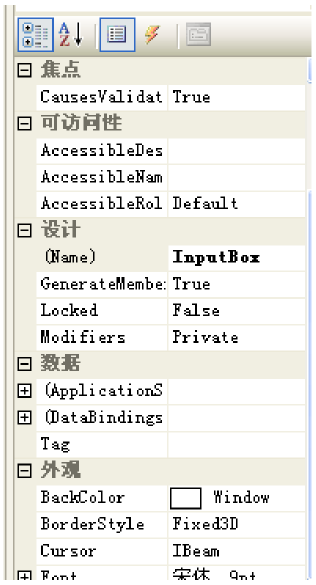


⊟ 设计	
(Name)	**ShowBox**
GenerateMembe	True
Locked	False
Modifiers	Private
⊟ 数据	
⊞ (ApplicationS	
⊞ (DataBindings	
Tag	
⊟ 外观	
BackColor	□ Window
BorderStyle	Fixed3D
Cursor	IBeam
⊞ Font	宋体, 9pt
ForeColor	■ WindowText
⊞ Lines	**String[] Arr**
RightToLeft	No
ScrollBars	**Vertical**
Text	
TextAlign	Left
UseWaitCursor	False
⊟ 行为	

图 8

再在 ShowBox 下用 TextBox 拉出一输入窗口，在属性项将 Name 设置为 InputBox，如下图 9 所示

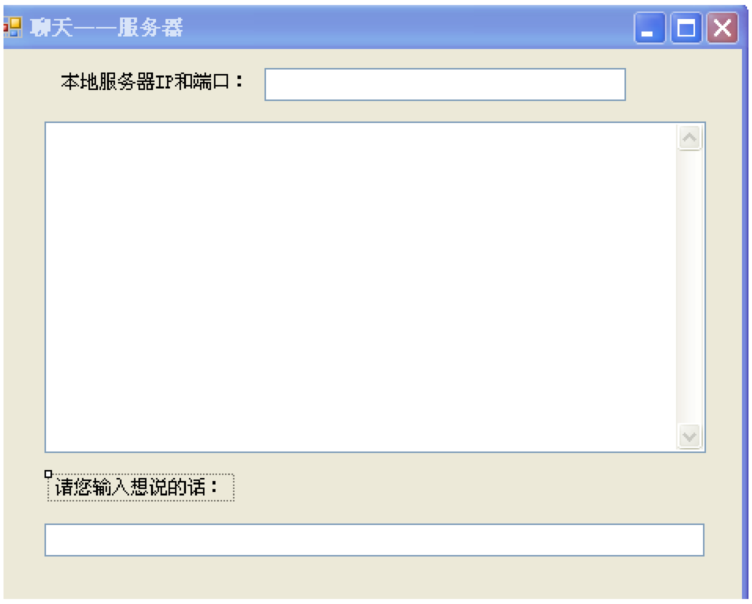


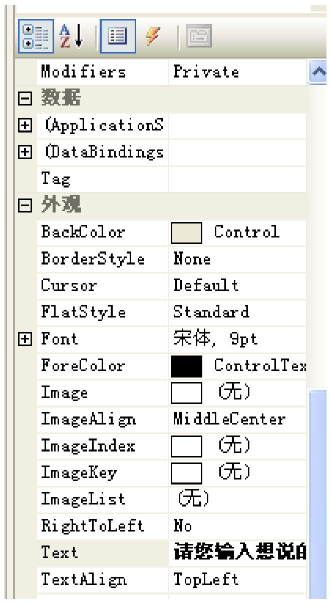


⊟ 焦点	
CausesValidat	True
⊟ 可访问性	
AccessibleDes	
AccessibleNam	
AccessibleRol	Default
⊟ 设计	
(Name)	**InputBox**
GenerateMembe	True
Locked	False
Modifiers	Private
⊟ 数据	
⊞ (ApplicationS	
⊞ (DataBindings	
Tag	
⊟ 外观	
BackColor	Window
BorderStyle	Fixed3D
Cursor	IBeam
⊞ Font	宋体, 9pt

图 9

在 ShowBox 与 InputBox 之间用 Label 拉出一窗口，在 Text 中将“Label”改为“请您输入想说的话：”，如下图 10 所示





Modifiers	Private
⊟ 数据	
⊞ (ApplicationS	
⊞ (DataBindings	
Tag	
⊟ 外观	
BackColor	Control
BorderStyle	None
Cursor	Default
FlatStyle	Standard
⊞ Font	宋体, 9pt
ForeColor	ControlTex
Image	(无)
ImageAlign	MiddleCenter
ImageIndex	(无)
ImageKey	(无)
ImageList	(无)
RightToLeft	No
Text	**请您输入想说的**
TextAlign	TopLeft

图 10

添加“发送”和“退出”按键，如下图 11 所示，选取 Button 项拉出两按键，分别在 Name 项将 Button1 更改为“SendButton”和“QuitButton”，而在 Text 项中将 Button1 分别更改为“发送”和“退出”!

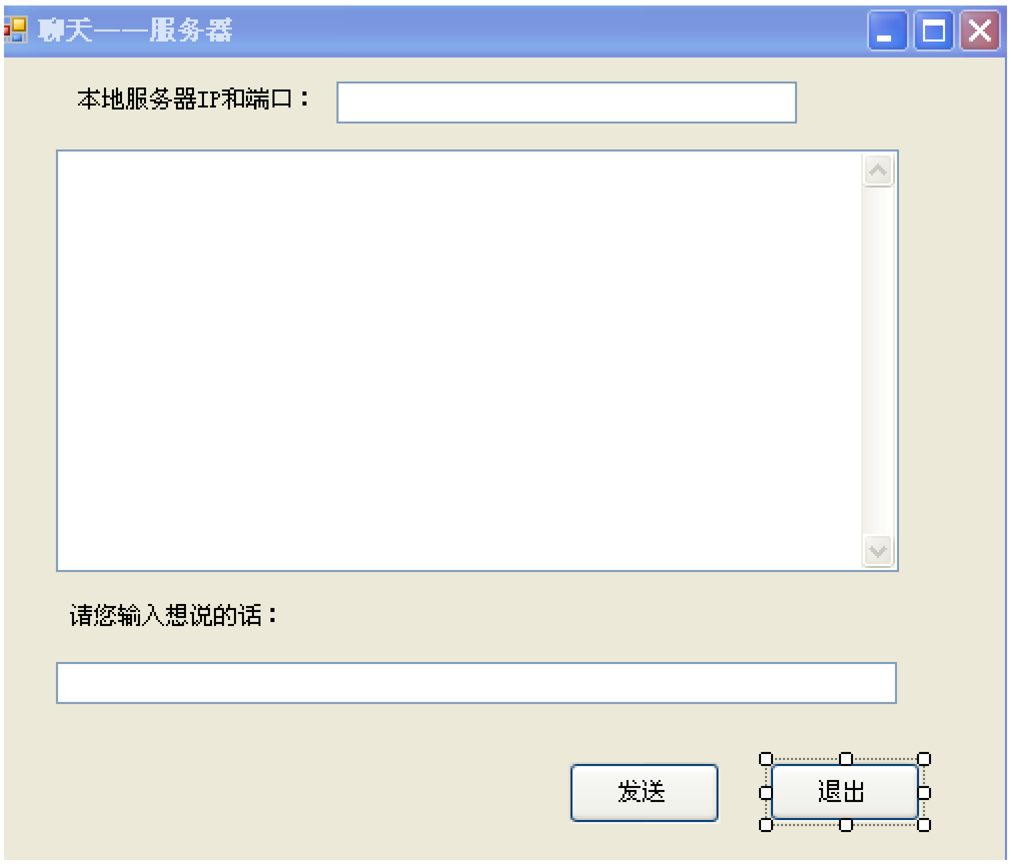
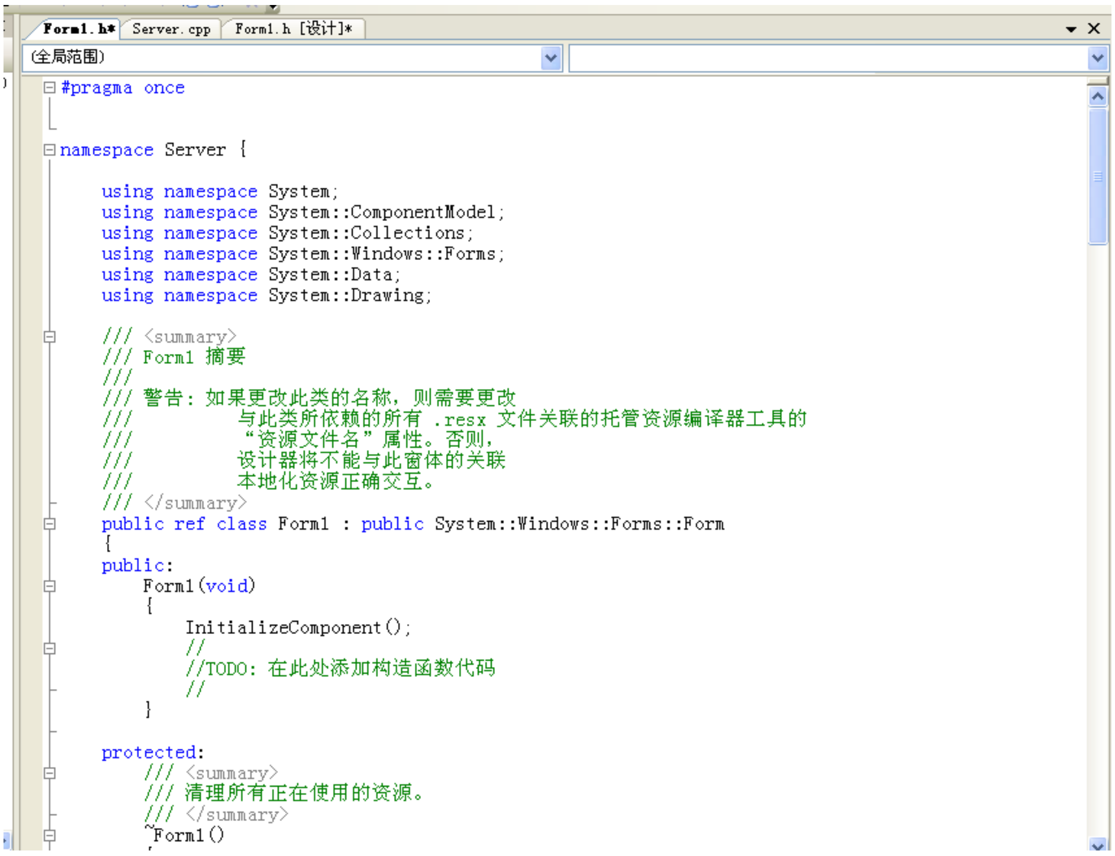


图 11

以上基本设置完了界面，调试后会出现此界面，类 QQ 的形式！
现在对代码进行编写，双击左边的 Server.cpp 文件，进入到里面，
在 using namespace Server;下加多一行代码，如下

```
using namespace Server;
using namespace System::Threading;
```

选取 Form1.h 文件，右击选取“查看代码”进入下窗口，如下图 12 所示



Form1.h* | Server.cpp | Form1.h [设计]*

(全局范围)

```
#pragma once

namespace Server {

	using namespace System;
	using namespace System::ComponentModel;
	using namespace System::Collections;
	using namespace System::Windows::Forms;
	using namespace System::Data;
	using namespace System::Drawing;

	/// <summary>
	/// Form1 摘要
	///
	/// 警告: 如果更改此类的名称，则需要更改
	///          与此类所依赖的所有 .resx 文件关联的托管资源编译器工具的
	///          “资源文件名”属性。否则，
	///          设计器将不能与此窗体的关联
	///          本地化资源正确交互。
	/// </summary>
	public ref class Form1 : public System::Windows::Forms::Form
	{
	public:
		Form1(void)
		{
			InitializeComponent();
			//
			//TODO: 在此处添加构造函数代码
			//
		}

	protected:
		/// <summary>
		/// 清理所有正在使用的资源。
		/// </summary>
		~Form1()
```

图 12

将里面的代码按下面的代码编写

```
#pragma once


namespace Server {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Threading;
    using namespace System::Text;
    using namespace System::Net;
    using namespace System::Net::Sockets;
    using namespace System::IO;


    /// <summary>
    /// Form1 摘要
    ///
```

```
/// 警告: 如果更改此类的名称，则需要更改
///          与此类所依赖的所有.resx 文件关联的托管资源编译器工具的
///          “资源文件名”属性。否则，
///          设计器将不能与此窗体的关联
///          本地化资源正确交互。
/// </summary>
public ref class Form1 : public System::Windows::Forms::Form
{
    //定义一些全局变量
    Socket^ s;
    Socket^ temp;
    Thread^ thread;
    int port;
    String^ host;

public:
    Form1(void)
    {
        InitializeComponent();
        //
        //TODO: 在此处添加构造函数代码
        //
    }

protected:
    /// <summary>
    /// 清理所有正在使用的资源。
    /// </summary>
    ~Form1()
    {
        if (components)
        {
            delete components;
        }
    }
private: System::Windows::Forms::TextBox^  ShowBox;
private: System::Windows::Forms::TextBox^  InputBox;
private: System::Windows::Forms::TextBox^  IPBox;
private: System::Windows::Forms::Label^  label1;
private: System::Windows::Forms::Button^  SendButton;
private: System::Windows::Forms::Label^  label2;
private: System::Windows::Forms::Button^  QuitButton;
public:  String^ m_ShowText;
public:  String^ IP_ShowText;
```

```
    public:

    protected:

    private:
        /// <summary>
        /// 必需的设计器变量。
        /// </summary>
        System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
        /// <summary>
        /// 设计器支持所需的方法- 不要
        /// 使用代码编辑器修改此方法的内容。
        /// </summary>
        void InitializeComponent(void)
        {
            this->ShowBox = (gcnew System::Windows::Forms::TextBox());
            this->InputBox = (gcnew System::Windows::Forms::TextBox());
            this->label1 = (gcnew System::Windows::Forms::Label());
            this->SendButton = (gcnew System::Windows::Forms::Button());
            this->QuitButton = (gcnew System::Windows::Forms::Button());
            this->IPBox = (gcnew System::Windows::Forms::TextBox());
            this->label2 = (gcnew System::Windows::Forms::Label());
            this->SuspendLayout();
            //
            // IPBox
            //
            IPBox->Location = System::Drawing::Point(162, 38);
            IPBox->Name = L"IPBox";
            IPBox->ScrollBars =
System::Windows::Forms::ScrollBars::Horizontal;
            IPBox->Size = System::Drawing::Size(348, 21);
            IPBox->TabIndex = 2;
            IPBox->TextChanged += gcnew System::EventHandler(this,
&Form1::IPBox_TextChanged);
            //
            // ShowBox
            //
            this->ShowBox->Location = System::Drawing::Point(14, 74);
            this->ShowBox->Multiline = true;
            this->ShowBox->Name = L"ShowBox";
            this->ShowBox->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
            this->ShowBox->Size = System::Drawing::Size(594, 223);
            this->ShowBox->TabIndex = 2;
            this->ShowBox->TextChanged += gcnew System::EventHandler(this,
&Form1::ShowBox_TextChanged);
            //
            // InputBox
            //
            this->InputBox->Location = System::Drawing::Point(12, 375);
            this->InputBox->Name = L"InputBox";
            this->InputBox->Size = System::Drawing::Size(525, 21);
            this->InputBox->TabIndex = 3;
            this->InputBox->TextChanged += gcnew System::EventHandler(this,
&Form1::InputBox_TextChanged);
            //
            // label1
            //
            this->label1->AutoSize = true;
            this->label1->Location = System::Drawing::Point(10, 344);
            this->label1->Name = L"label1";
            this->label1->Size = System::Drawing::Size(113, 12);
            this->label1->TabIndex = 4;
            this->label1->Text = L"请输入您想说的话：";
            //
            // SendButton
            //
            this->SendButton->Location = System::Drawing::Point(369, 415);
            this->SendButton->Name = L"SendButton";
            this->SendButton->Size = System::Drawing::Size(71, 31);
            this->SendButton->TabIndex = 5;
            this->SendButton->Text = L"发送";
            this->SendButton->UseVisualStyleBackColor = true;
            this->SendButton->Click += gcnew System::EventHandler(this,
&Form1::SendButton_Click);
            //
            // QuitButton
            //
            this->QuitButton->Location = System::Drawing::Point(478, 415);
            this->QuitButton->Name = L"QuitButton";
            this->QuitButton->Size = System::Drawing::Size(73, 31);
            this->QuitButton->TabIndex = 6;
            this->QuitButton->Text = L"退出";
            this->QuitButton->UseVisualStyleBackColor = true;
            this->QuitButton->Click += gcnew System::EventHandler(this,
&Form1::QuitButton_Click);
            //
            // label2
            //
            this->label2->AutoSize = true;
            this->label2->Location = System::Drawing::Point(31, 41);
            this->label2->Name = L"label2";
            this->label2->Size = System::Drawing::Size(125, 12);
            this->label2->TabIndex = 8;
            this->label2->Text = L"本地服务器IP及端口：";
            //
            // Form1
            //
            this->AutoScaleDimensions = System::Drawing::SizeF(6, 12);
            this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
            this->ClientSize = System::Drawing::Size(634, 477);
            this->Controls->Add(this->label2);
            this->Controls->Add(IPBox);
            this->Controls->Add(this->QuitButton);
            this->Controls->Add(this->SendButton);
            this->Controls->Add(this->label1);
            this->Controls->Add(this->InputBox);
            this->Controls->Add(this->ShowBox);
            this->Name = L"Form1";
            this->Text = L"聊天——服务器";
            this->Load += gcnew System::EventHandler(this,
&Form1::Form1_Load);
            this->ResumeLayout(false);
            this->PerformLayout();

        }
#pragma endregion

        delegate void UpdateShowBox_Invoke();
        void UpdateShowBox()
        {
            ShowBox->AppendText(m_ShowText);
        }

        delegate void UpdateIPBox_Invoke();
        void UpdateIPBox()
        {
            IPBox->AppendText(IP_ShowText);
        }


        void DoWork(){
            port = 2000;//设置端口号
            //host = "172.18.190.93";//设置服务器地址为"172.18.190.93", 特例
型
            String^ host =
System::Net::Dns::GetHostAddresses(Environment::MachineName)[0]->ToString()
;//获取本地服务器的IP
            IPAddress^ ip = IPAddress::Parse(host);
            IPEndPoint^ ipe = gcnew IPEndPoint(ip,port);
            s = gcnew
Socket(AddressFamily::InterNetwork,SocketType::Stream,ProtocolType::Tcp);//
创建一个socket类
            s->Bind(ipe);//绑定端口
            IP_ShowText = host+" // "+port;
            this->Invoke(gcnew
UpdateIPBox_Invoke(this,&Form1::UpdateIPBox));//显示IP和端口号
            s->Listen(0);//开始监听
            temp = s->Accept();//为新建连接创建新的Socket
            //连接上后进行死循环, 避免断开连接
            while(1){
                try{
                    String^ recvStr = "";
                    array<Byte>^ recvBytes = gcnew array<Byte>(1024);
                    int bytes;
    bytes = temp->Receive(recvBytes,recvBytes->Length,SocketFlags::None);
//从客户端接收信息
                    recvStr = ncoding::Default->GetString(recvBytes,0,bytes);
//转换数据为字符串
                    m_ShowText = "\r\n"+"Client 说:"+"\r\n"+recvStr+"\r\n";
//加上换行符把客户端传来的信息显示出来
        this->Invoke(gcnew UpdateShowBox_Invoke(this,&Form1::UpdateShowBox));

                }

                catch(EndOfStreamException^ e)
                {
                }
                catch(IOException^ e)
                {
                    MessageBox::Show("I/O error");
                }
                }
            }


private: System::Void SendButton_Click(System::Object^  sender,
System::EventArgs^  e) {

            String^ sendStr = InputBox->Text;
            if(sendStr->Length > 0)
            {
        m_ShowText = "\r\n"+"Server 说:"+"\r\n"+sendStr+"\r\n";//加上换行符
            this->Invoke(gcnew
UpdateShowBox_Invoke(this,&Form1::UpdateShowBox));//本窗口显示发出去的内容
            array<Byte>^ bs = Encoding::Default->GetBytes(sendStr);
//将字符串转为二进制，支持中英文传输
            temp->Send(bs,bs->Length,SocketFlags::None);
//把当前的聊天内容发送给客户端
            InputBox->Text = "";
            }
         }
private: System::Void InputBox_TextChanged(System::Object^  sender,
System::EventArgs^  e) {
             }
private: System::Void Form1_Load(System::Object^  sender, System::EventArgs^
e) {
             thread = gcnew Thread(gcnew ThreadStart(this,&Form1::DoWork));
             thread->IsBackground = true;
             thread->Start();
         }
private: System::Void QuitButton_Click(System::Object^  sender,
System::EventArgs^  e) {
            temp->Close();
             s->Close();
             thread->Abort();
             Application::Exit();
         }
private: System::Void IPBox_TextChanged(System::Object^  sender,
System::EventArgs^  e) {
            }
private: System::Void ShowBox_TextChanged(System::Object^  sender,
System::EventArgs^  e) {
         }
};
}
```

按照服务器编写的方面，同理可进行客户端编写，先设置界面，再进行代码编写，部分代码是自动生成的，如界面设置的代码，要想设置界面按键的代码，双击它就会自动生成，且进入 Form1.h 中进行设置，客户端的代码如下所示：

先是 Client.cpp 中多加一行代码，如下

```
using namespace System::Threading;
using namespace Client;
```

再是Form1.h的代码，如下

```
#pragma once


namespace Client {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace System::Threading;
    using namespace System::Text;
    using namespace System::Net;
    using namespace System::Net::Sockets;
    using namespace System::IO;
    //using namespace System::Text::Encoding::Default.GetBytes;
    //using namespace System::Text::Encoding::Default::GetBytes;

    /// <summary>
    /// Form1 摘要
    ///
    /// 警告: 如果更改此类的名称，则需要更改
    ///          与此类所依赖的所有.resx 文件关联的托管资源编译器工具的
    ///          “资源文件名”属性。否则，
    ///          设计器将不能与此窗体的关联
    ///          本地化资源正确交互。
    /// </summary>
    public ref class Form1 : public System::Windows::Forms::Form
    {
    public:

        Socket^ s;
        Thread^ thread;

        Form1(void)
        {
```

```
			InitializeComponent();
			//
			//TODO: 在此处添加构造函数代码
			//
		}

	protected:
		/// <summary>
		/// 清理所有正在使用的资源。
		/// </summary>
		~Form1()
		{
			if (components)
			{
				delete components;
			}
		}
	private: System::Windows::Forms::TextBox^  CShowBox;
	private: System::Windows::Forms::TextBox^  CInputBox;
	private: System::Windows::Forms::TextBox^  IPBox;
	private: System::Windows::Forms::Label^  label1;
	private: System::Windows::Forms::Label^  label2;
	private: System::Windows::Forms::Button^  CSendButton;
	private: System::Windows::Forms::Button^  QuitButton;

	public:  String^ m_CShowText;
	public:  String^ IP_ShowText;

	protected:

	private:
		//void Click_Button();
		//int buttonCount;
		/// <summary>
		/// 必需的设计器变量。
		/// </summary>
		System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
		/// <summary>
		/// 设计器支持所需的方法- 不要
		/// 使用代码编辑器修改此方法的内容。
		/// </summary>
		void InitializeComponent(void)
```

```
           {
               this->CShowBox = (gcnew System::Windows::Forms::TextBox());
               this->CInputBox = (gcnew System::Windows::Forms::TextBox());
               this->label1 = (gcnew System::Windows::Forms::Label());
               this->CSendButton = (gcnew System::Windows::Forms::Button());
               this->QuitButton = (gcnew System::Windows::Forms::Button());
               this->IPBox = (gcnew System::Windows::Forms::TextBox());
               this->label2 = (gcnew System::Windows::Forms::Label());
               this->SuspendLayout();
               //
               // CShowBox
               //
               this->CShowBox->Location = System::Drawing::Point(32, 80);
               this->CShowBox->Multiline = true;
               this->CShowBox->Name = L"CShowBox";
               this->CShowBox->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
               this->CShowBox->Size = System::Drawing::Size(537, 251);
               this->CShowBox->TabIndex = 2;
               //
               // CInputBox
               //
               this->CInputBox->Location = System::Drawing::Point(32, 376);
               this->CInputBox->Name = L"CInputBox";
               this->CInputBox->Size = System::Drawing::Size(490, 21);
               this->CInputBox->TabIndex = 3;
               this->CInputBox->TextChanged += gcnew System::EventHandler(this,
&Form1::CInputBox_TextChanged);
               //
               // label1
               //
               this->label1->AutoSize = true;
               this->label1->Location = System::Drawing::Point(30, 349);
               this->label1->Name = L"label1";
               this->label1->Size = System::Drawing::Size(113, 12);
               this->label1->TabIndex = 4;
               this->label1->Text = L"请输入您想说的话：";
               //
               // CSendButton
               //
               this->CSendButton->Location = System::Drawing::Point(366, 418);
               this->CSendButton->Name = L"CSendButton";
               this->CSendButton->Size = System::Drawing::Size(79, 25);
               this->CSendButton->TabIndex = 5;
```

```
            this->CSendButton->Text = L"发送";
            this->CSendButton->UseVisualStyleBackColor = true;
            this->CSendButton->Click += gcnew System::EventHandler(this, 
&Form1::CSendButton_Click);
            // 
            // QuitButton
            // 
            this->QuitButton->Location = System::Drawing::Point(462, 418);
            this->QuitButton->Name = L"QuitButton";
            this->QuitButton->Size = System::Drawing::Size(79, 28);
            this->QuitButton->TabIndex = 6;
            this->QuitButton->Text = L"退出";
            this->QuitButton->UseVisualStyleBackColor = true;
            this->QuitButton->Click += gcnew System::EventHandler(this, 
&Form1::QuitButton_Click);
            // 
            // IPBox
            // 
            this->IPBox->Location = System::Drawing::Point(196, 30);
            this->IPBox->Name = L"IPBox";
            this->IPBox->ScrollBars = 
System::Windows::Forms::ScrollBars::Horizontal;
            this->IPBox->Size = System::Drawing::Size(188, 21);
            this->IPBox->TabIndex = 2;
            this->IPBox->TextChanged += gcnew System::EventHandler(this, 
&Form1::IPBox_TextChanged);
            // 
            // label2
            // 
            this->label2->AutoSize = true;
            this->label2->Location = System::Drawing::Point(65, 33);
            this->label2->Name = L"label2";
            this->label2->Size = System::Drawing::Size(125, 12);
            this->label2->TabIndex = 8;
            this->label2->Text = L"监测服务器IP及端口：";
            // 
            // Form1
            // 
            this->AutoScaleDimensions = System::Drawing::SizeF(6, 12);
            this->AutoScaleMode = 
System::Windows::Forms::AutoScaleMode::Font;
            this->ClientSize = System::Drawing::Size(600, 458);
            this->Controls->Add(this->label2);
            this->Controls->Add(this->IPBox);
            this->Controls->Add(this->QuitButton);
            this->Controls->Add(this->CSendButton);
            this->Controls->Add(this->label1);
            this->Controls->Add(this->CInputBox);
            this->Controls->Add(this->CShowBox);
            this->Name = L"Form1";
            this->Text = L"聊天——客户端";
            this->Load += gcnew System::EventHandler(this,
&Form1::Form1_Load);
            this->ResumeLayout(false);
            this->PerformLayout();

        }
#pragma endregion

    public:
        //调用CShowBox函数，实现窗口显示
        delegate void UpdateCShowBox_Invoke();
        void UpdateCShowBox()
        {
            CShowBox->AppendText(m_CShowText);
        }
        //调用IPBox函数，实现窗口IP和端口显示
        delegate void UpdateIPBox_Invoke();
        void UpdateIPBox()
        {
            IPBox->AppendText(IP_ShowText);
        }

        void CSendText(){
            int port = 2000;//设置端口号
            String^ host = "172.18.190.93";
//设置服务器地址为“.18.190.93”，只能预先知道服务器的IP
            IPAddress^ ip = IPAddress::Parse(host);
            IPEndPoint^ ipe = gcnew IPEndPoint(ip,port);
            s = gcnew
Socket(AddressFamily::InterNetwork,SocketType::Stream,ProtocolType::Tcp);
//创建一个Socket类
            s->Connect(ipe);
            IP_ShowText = host+" // "+port;
            this->Invoke(gcnew
UpdateIPBox_Invoke(this,&Form1::UpdateIPBox));//进行IP和端口显示

            while(1){
                try{
                    String^ recvStr;
                    array<Byte>^ recvBytes = gcnew array<Byte>(1024);
                    int bytes;
                    bytes =
s->Receive(recvBytes,recvBytes->Length,SocketFlags::None);
//从服务器接收所有聊天内容
                    recvStr =
Encoding::Default->GetString(recvBytes,0,bytes);//把接收的数据转为字符串
                    m_CShowText = "\r\n"+"Server 说:"+"\r\n"+recvStr+"\r\n";
                    this->Invoke(gcnew
UpdateCShowBox_Invoke(this,&Form1::UpdateCShowBox));//进行窗口显示

                }
                catch(EndOfStreamException^ e){}
                catch(IOException^ e)
                {
                    MessageBox::Show("I/O error");
                }
            }
        }

private: System::Void Form1_Load(System::Object^  sender, System::EventArgs^
e)  {

            thread = gcnew Thread(gcnew
ThreadStart(this,&Form1::CSendText));//创建线程
            thread->IsBackground = true;
            thread->Start();//启动线程
        }
private: System::Void CSendButton_Click(System::Object^  sender,
System::EventArgs^  e)  {
             String^ str = CInputBox->Text;
             if(str->Length > 0)//要是用户没输入，则不发送，即输入不能为空
             {
             m_CShowText = "\r\n"+"Client 说:"+"\r\n"+str+"\r\n";
             this->Invoke(gcnew
UpdateCShowBox_Invoke(this,&Form1::UpdateCShowBox));
//调用函数，将输入的话显示出来
             array<Byte>^ bs = Encoding::Default->GetBytes(str);
//将字符串转为二进制，避免传输过程不会出现乱码
             s->Send(bs,bs->Length,SocketFlags::None);//向服务器发送当前用户
说的话
             CInputBox->Text = "";//清空输入框
```

```
                }
            }
private: System::Void CInputBox_TextChanged(System::Object^  sender,
System::EventArgs^  e) {
            }
private: System::Void QuitButton_Click(System::Object^  sender,
System::EventArgs^  e) {
               //退出按键，关套接字、线程，退出界面
                s->Close();
                thread->Abort();
                Application::Exit();
            }
private: System::Void IPBox_TextChanged(System::Object^  sender,
System::EventArgs^  e) {
            }
};
}
```

以上基本完成服务器和客户端的编写，调试时要先运行服务器，再运行客户端，切勿相反，否则会出错，这跟socket的运行机制有关！
另外，代码是从开发环境中直接复制过来的，请自行对格式进行检查，代码是正确的！