void CServerDlg::OnPaint()

{

if (IsIconic())

{

CPaintDC dc(this); // 用于绘制的设备上下文

SendMessage(WM\_ICONERASEBKGND, reinterpret\_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 使图标在工作区矩形中居中

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;

// 绘制图标

dc.DrawIcon(x, y, m\_hIcon);

}

else

{

CDialogEx::OnPaint();

}

}

//当用户拖动最小化窗口时系统调用此函数取得光标

//显示。

HCURSOR CServerDlg::OnQueryDragIcon()

{

return static\_cast<HCURSOR>(m\_hIcon);

}

void CServerDlg::OnEnChangeEdit1()

{

// TODO: 如果该控件是 RICHEDIT 控件，它将不

// 发送此通知，除非重写 CDialogEx::OnInitDialog()

// 函数并调用 CRichEditCtrl().SetEventMask()，

// 同时将 ENM\_CHANGE 标志“或”运算到掩码中。

// TODO: 在此添加控件通知处理程序代码

}

void CServerDlg::OnBnClickedBtnBrowser()

{

CFileDialog dlg(TRUE);

if (IDCANCEL == dlg.DoModal())

return;

m\_strFilePath = dlg.GetPathName();

UpdateData(FALSE);

}

//发送

void CServerDlg::OnBnClickedBtnSend()

{

UpdateData(TRUE);

if (m\_strFilePath.IsEmpty())

{

MessageBox(\_T("请您先选择需要发送的文件."), \_T("温馨提示"), MB\_OK);

return;

}

int nSel = m\_list.GetSelectionMark();

if (nSel < 0)

{

MessageBox(\_T("请您先选择需要发送的人."), \_T("温馨提示"), MB\_OK);

return;

}

//获取套接字

CClientSocket\*pSocket = (CClientSocket\*)m\_list.GetItemData(nSel);

//获取文件属性

WIN32\_FIND\_DATA wfd;

HANDLE hFinder = FindFirstFile(m\_strFilePath, &wfd);

FindClose(hFinder);

//先发文件属性

pSocket->Send(&wfd, sizeof(wfd));

//发送文件内容

CFile file;

if (FALSE == file.Open(m\_strFilePath, CFile::modeRead|CFile::typeBinary))

{

MessageBox(\_T("打开文件失败."), \_T("温馨提示"), MB\_OK);

return;

}

//循环发送

DWORD dwReadCount = 0;

while (dwReadCount < wfd.nFileSizeLow)

{

char \* buffer = new char[1025];

int nRead = file.Read(buffer, 1024);

buffer[nRead] = '\0';

//发送

pSocket->Send(buffer, nRead);

dwReadCount += nRead;

delete[] buffer;

}

//关闭文件

file.Close();

MessageBox(\_T("发送数据完毕"), \_T("温馨提示"), MB\_OK);

}

void CServerDlg::OnNMCustomdrawProgress(NMHDR \*pNMHDR, LRESULT \*pResult)

{

LPNMCUSTOMDRAW pNMCD = reinterpret\_cast<LPNMCUSTOMDRAW>(pNMHDR);

// TODO: 在此添加控件通知处理程序代码

\*pResult = 0;

}

void CServerDlg::OnTimer(UINT\_PTR nIDEvent)

{

CDialogEx::OnTimer(nIDEvent);

int pos = 0;

for (int i = 0; i<100; i++){

pos = pos + 1;

if (pos>100)

pos = 0;

m\_progress.SetPos(pos);

}

// TODO: 在此添加消息处理程序代码和/或调用默认值

CDialog::OnTimer(nIDEvent);

// TODO: 在此添加消息处理程序代码和/或调用默认值

}

void CServerDlg::OnLvnItemchangedList(NMHDR \*pNMHDR, LRESULT \*pResult)

{

LPNMLISTVIEW pNMLV = reinterpret\_cast<LPNMLISTVIEW>(pNMHDR);

// TODO: 在此添加控件通知处理程序代码

\*pResult = 0;

}