

原创 zhangxi8211 于 2016-03-17 16:22:13 发布 4576 收藏 10

源码分享 专栏收录该内容 1 订阅 2 篇文章 订阅专栏

近期由于项目需要，编写了一个基于MFC<sup>Q</sup>对话框的Windows服务程序，现把方法和大家分享一下，希望高手指正。程序实现的基本功能如下：

- 1、注册为Windows服务程序，随系统启动
- 2、作为服务端，提供客户端的连接（TCP）
- 3、有可视化<sup>Q</sup>界面，监视客户端的连接情况
- 4、具有系统托盘图标

- 4、具有系统托盘图标

最终运行界面为：



```

        MessageBox(NULL, _T("Couldn't open service manager"), szServiceName, MB_OK);
        return FALSE;
    }
    // Get the executable file path
    TCHAR szFilePath[MAX_PATH];
    ::GetModuleFileName(NULL, szFilePath, MAX_PATH);
    //创建服务
    SC_HANDLE hService = ::CreateService(hSCM, szServiceName, szServiceName,
        SERVICE_ALL_ACCESS, SERVICE_WIN32_OWN_PROCESS|SERVICE_INTERACTIVE_PROCESS,
        SERVICE_AUTO_START, SERVICE_ERROR_NORMAL,
        szFilePath, NULL, NULL, _T(""), NULL, NULL);
    if (hService == NULL)
    {
        :::CloseServiceHandle(hSCM);
        MessageBox(NULL, _T("Couldn't create service"), szServiceName, MB_OK);
        return FALSE;
    }
    :::CloseServiceHandle(hService);
    :::CloseServiceHandle(hSCM);
    return TRUE;
}

BOOL IsInstalled()
{
    BOOL bResult = FALSE;

    //打开服务控制管理器
    SC_HANDLE hSCM = ::OpenSCManager(NULL, NULL, SC_MANAGER_ALL_ACCESS);

    if (hSCM != NULL)
    {
        //打开服务
        SC_HANDLE hService = ::OpenService(hSCM, szServiceName, SERVICE_QUERY_CONFIG);
        if (hService != NULL)
        {
            bResult = TRUE;
            :::CloseServiceHandle(hService);
        }
        :::CloseServiceHandle(hSCM);
    }
    return bResult;
}

```

(4) 在主对话框中自定义消息: #define WM\_USER\_CLOSE\_WND(WM\_USER+111), 用于程序退出。在ServerSocket.h文件中, 添加CSocketTestDlg类的消息函数: afx\_msg void OnUserClose(WPARAM wParam,LPARAM lParam); 并在ServerSocket.cpp文件中添加消息映射ON\_MESSAGE(WM\_USER\_CLOSE\_WND,OnUserClose), OnUserClose函数的实现代码如下:

```

void CSocketTestDlg::OnUserClose(WPARAM wParam,LPARAM lParam)
{
    BOOL bVisible = (this->GetStyle() & WS_VISIBLE);
    if (bVisible)
    {
        ShowWindow(SW_HIDE);
    }
    CDialog::OnOK();
}

```

### 三、实现系统托盘图标

(1) 在ServerSocket.h文件中定义: #define UM\_TRAYNOTIFY(WM\_USER+110), 在CSocketTestDlg类中添加成员变量: NOTIFYICONDATA m\_nid, 添加消息定义函数: afx\_msg LRESULT OnTrayNotify(WPARAM wParam,LPARAM lParam); 在SocketTestDlg.cpp文件中添加消息映射: ON\_MESSAGE(UM\_TRAYNOTIFY,OnTrayNotify), OnTrayNotify函数代码如下:

```

LRESULT CSocketTestDlg::OnTrayNotify(WPARAM wParam,LPARAM lParam)
{
    UINT uMsg = (UINT)lParam;
    switch(uMsg)
    {
        case UM_BUTTONUP:
        {
            //右键处理
            CMenu menuTray;
            CPoint point;
            int id;
            GetCursorPos(&point);
            menuTray.LoadMenu(IDR_MENU_TRAY);
            id = menuTray.GetSubMenu(0)->TrackPopupMenu(TPM_RETURNCMD |
                TPM_LEFTBUTTON|TPM_RIGHTBUTTON, point.x, point.y, this);

            if (id)
            {
                CString strInfo;
                strInfo.Format(L"Menuid %d", id);
                LPCTSTR strtmp;
                strtmp = strInfo.GetBuffer(0);
                MessageBox(strtmp, L"test");
            }

            if (id)
            {
                switch(id){
                    case IDR_MENU_EXIT:
                    {
                        OnClose();
                        break;
                    }
                    case IDR_TRAY_RESTORE:
                    {
                        //窗口前缘显示
                        SetForegroundWindow();
                        ShowWindow(SW_SHOWNORMAL);
                        break;
                    }
                    default:
                    {
                        break;
                    }
                }
            }
            break;
        }
        case UM_LBUTTONDBLCLK:
        {
            SetForegroundWindow();
            ShowWindow(SW_SHOWNORMAL);
            break;
        }
        default:
        {
            break;
        }
    }
    return 0;
}

```

(2) 在CSocketTestDlg的构造函数和析构函数中添加如下代码

```

CSocketTestDlg::CSocketTestDlg(CWnd* pParent /*=NULL*/)
: CDialog(CSocketTestDlg::IDD, pParent)
{
    //{{AFX_DATA_INIT(CSocketTestDlg)
    // NOTE: the ClassWizard will add member initial
    //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    memset(&m_nid,0,sizeof(m_nid));
    m_nid.cbSize = sizeof(m_nid);
    m_nid.uFlags = NIF_ICON | NIF_TIP | NIF_MESSAGE;
}

CSocketTestDlg::~CSocketTestDlg()
{
    m_nid.hIcon = NULL;
    Shell_NotifyIcon(NIM_DELETE,&m_nid);
}

```

(3) 在EBOOL CSocketTestDlg::OnInitDialog()函数中添加如图代码

```

// TODO: Add extra initialization here
m_nid.hWnd = GetSafeHwnd();
m_nid.cbSize = sizeof(m_nid);
m_nid.hIcon = m_hIcon;
CString strToolTip = _T("XXX系统");
CString strToolTip = _T("XXX系统");
CString strToolTip = _T("XXX系统");
Shell_NotifyIcon(NIM_ADD,&m_nid);

```

(4) 添加窗口的WM\_SIZE和WM\_CLOSE消息函数

```

void CSocketTestDlg::OnSize(UINT nType, int cx, int cy)
{
    CDialog::OnSize(nType, cx, cy);
    if(nType == SIZE_MINIMIZED){
        ShowWindow(SW_HIDE);
    }
}

void CSocketTestDlg::OnClose()
{
    BOOL bVisible = (this->GetStyle() & WS_VISIBLE);
    if (bVisible)
    {
        ShowWindow(SW_HIDE);
    }
    else
    {
        CDialog::OnOK();
    }
}

```

(5) 在资源管理器中添加一个菜单 (IDR\_MENU\_TRAY), 添加两个菜单项: "恢复窗口" (IDR\_TRAY\_RESTORE) 和"退出" (IDR\_TRAY\_EXIT)

### 四、实现网络服务

(1) 添加类CServerSocket, 派生于CSocket, 对该类进行类向导添加三个函数OnAccept(), OnClose(), OnReceive(), 函数中调用的函数都在主对话框中实现。

```

#include "SocketTestDlg.h"
// CServerSocket command target
class CSocketTestDlg;
class CServerSocket : public CSocket
{
public:
    // Attributes
public:
    // Operations
public:
    CServerSocket();
    virtual ~CServerSocket();
    CSocketTestDlg * m_main_dlg; //主对话框指针
// Overrides
public:
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CServerSocket)
public:
    virtual void OnAccept(int nErrorCode);
    virtual void OnClose(int nErrorCode);
    virtual void OnReceive(int nErrorCode);
    //}}AFX_VIRTUAL

    // Generated message map functions
    //{{AFX_MSG(CServerSocket)
    // NOTE - the ClassWizard will add and remove member functions here.
    //}}AFX_MSG
// Implementation
protected:
public:
}

void CServerSocket::OnAccept(int nErrorCode)
{
    m_main_dlg->AddClient(); //添加上线用户
    CSocket::OnAccept(nErrorCode);
}

void CServerSocket::OnClose(int nErrorCode)

```



里面TcpAgent模式应该正x以实现这个

C++实现单个客户端连接多个服务器 (T...  
sunwayland2021: HPocket库里面TcpAge  
nt模式应该正好实现这个需求

C++实现单个客户端连接多个服务器 (T...  
\\t-u-y)悲批: 你好, 我也是遇到这个问题  
了, 我有一个想法, 就是说能不能把服务...

C++实现单个客户端连接多个服务器 (T...  
sick\_boy: 您好, 我想请教一下这个是如何  
实现的? 我现在遇到类似的问题, 下位博...

您愿意向朋友推荐“博客详情页”吗？

强烈推荐 不推荐 一般般 推荐 强烈推荐

最新文章

Qt (mingw) 使用libevent网络库开发高性能  
服务端

C++实现单个客户端连接多个服务器 (TCP)

2022年 1篇

2021年 1篇

2016年 1篇

Windows 服务(附服务开发辅助工具)

weixin\_34005042的博客 182  
引子 近来在 Windows 下逛弄了一阵子的服务程序, 有在 C++ 下弄服务的, 也在 C# 下弄服务的, 感觉在 C# 下弄服务蛮简单的, C/C++ 的麻...

MFC系统服务开启停止代码

曲线人数 182  
// m\_service.cpp : 实现文件 // #include "stdafx.h" #include "MFCApplication1.h" #include "m\_service.h" #include "afxdialogex.h" // m\_service 对话框 m\_...

MFC程序版本自动升级更新

Kung's blog column 6503  
1、自动升级需要实现两个exe程序的交替启动, 需要一个主程序和一个升级程序; 在一个解决方案中创建两个项目, 一个是你的主程序, 另一个为你的开...

基于MFC框架的学生信息管理系统/学生成绩管理系统(带有图形化界面)

最新发布  
ambiguous\_ 的博客 1911  
系统背景: 本系统基于 Visual Studio2013的环境搭建, 运用MFC框架对系统进行设计和实现, 本系统带有图形化界面, 具有常用的用户登录注册功能, ...

MFC/C下控制windows服务——停止服务, 启动服务

C语言是最好的编程语言 5188  
MFC/C下控制windows服务——停止服务, 启动服务 操作系统: Windows7 64位 (经测试, Windows10不能用) 运行环境: Visual Studio 10 文件编辑...

如何利用MFC的嵌套类编写进程外com(对潘登民先生的com原理及应用的补充)

Jason的专栏 2454  
一般的com用例书上只给出了利用MFC的嵌套类编写进程内的com组件程序,对于进程外的com组件程序很少提及,就连潘登民先生的>所举的例子中也是...

C++建立 Windows 服务程序

修罗的博客 522  
新建控制台程序, 不需要界面 在stdafx.h中添加MFC支持: #include <afx.h> #include <afxwin.h> // MFC Core and standard components #include <afxex...

“相关推荐”对你有帮助么？

非常没帮助 没帮助 一般 有帮助 非常有帮助

©2022 CSDN 皮肤主题: 大白 设计师: CSDN官方博客 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004606号 京公网安备11010802021033 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务

中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照

