## NeT命令

NET命令是功能强大的以命令行方式执行的工具。它包含了管理网络环境、服务、用户、登陆等Windows 98/NT/2000 中大部分重要的管理功能。使用它可以轻松的管理本地或者远程计算机的网络环境，以及各种服务程序的运行和配置。或者进行用户管理和登陆管理等。



**Net User**

作用：添加或更改用户帐号或显示用户帐号信息。

命令格式：Net user [username [password | \*] [options]] [/domain]

有关参数说明：

·键入不带参数的Net user查看计算机上的用户帐号列表

·username添加、删除、更改或查看用户帐号名

·password为用户帐号分配或更改密码

·提示输入密码

·/domain在计算机主域的主域控制器中执行操作。该参数仅在 Windows NT Server 域成员的 Windows NT Workstation 计算机上可用。默认情况下，Windows NT Server 计算机在主域控制器中执行操作。注意：在计算机主域的主域控制器发生该动作。它可能不是登录域。

例如：Net user ghq123查看用户GHQ123的信息。

**Net Localgroup**

作 用：添加、显示或更改本地组。

命令格式：Net localgroup groupname {/add [/comment:"text "] | /delete} [/domain]

有关参数说明：

·键入不带参数的Net localgroup显示服务器名称和计算机的本地组名称

·groupname要添加、扩充或删除的本地组名称

·/comment: "text "为新建或现有组添加注释

·/domain在当前域的主域控制器中执行操作，否则仅在本地计算机上执行操作

·name [ ...]列出要添加到本地组或从本地组中删除的一个或多个用户名或组名

·/add将全局组名或用户名添加到本地组中

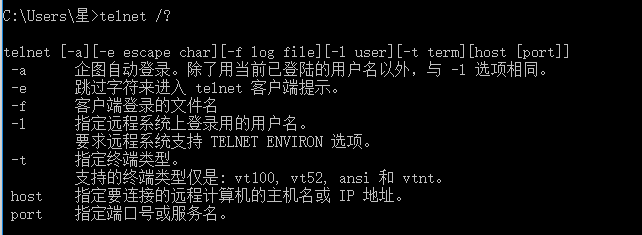
·/delete从本地组中删除组名或用户名

例如：Net localgroup ggg /add 将名为ggg的本地组添加到本地用户帐号数据库；

Net localgroup ggg 显示ggg本地组中的用户。

## TELNET命令

Telnet协议是TCP/IP协议族中的一员，是Internet远程登陆服务的标准协议和主要方式。它为用户提供了在本地计算机上完成远程主机工作的能力。在终端使用者的电脑上使用telnet程序，用它连接到服务器。终端使用者可以在telnet程序中输入命令，这些命令会在服务器上运行，就像直接在服务器的控制台上输入一样。可以在本地就能控制服务器。要开始一个telnet会话，必须输入用户名和密码来登录服务器。Telnet是常用的远程控制Web服务器的方法。



## 套接字编程函数

**加载套接字库并进行版本协商**

Int WSAStartup(WORD wVersionRequested,

//请求的版本号，低字节代表主版本，高字节代表副版本,一般我们用MAKEWORD（x,y）//宏来指定版本号,如：MAKEWORD（2,1）代表2.1的版本

LPWSADATA lpWSAData

//out,一个WSADATA结构体指针，用于接收实际加载的套接字 库版本号

)

**创建套接字**

SOCKET socket(int af, //指定协议族，也即网际域，对于windows平台总是AF\_INET 或 PF\_INET

Int type,//指定要创建的套接字类型

Int protocol//指定协议类型，我们一般设为0，让他根据我们前两个参数自动设置

）

**绑定到本机地址和端口**

int bind(SOCKET s, //已创建的套接字描叙符

const struct sockaddr FAR \*name, //要绑定的本机地址信息

Int namelen //第二个参数的长度

）

其中要注意第二个参数，由于第二个参数适用于所有网络协议，所以我们可以根据需要进行替换，如我们常常这样定义一个AF\_INET 的地址信息：

SOCKADDR\_IN SrvAddr;

SrvAddr.family=AF\_INET;

SrvAddr.port=hotns(666）; //其中666代表端口号

SrvAddr.sin\_addr.S\_un.S\_addr=htonl(192.168.1.101);

//或SrvAddr.sin\_addr.S\_un.S\_addr=INADDR\_ANY;(指绑定到本机的任一快网卡上，并且////其本身就是网络字节序，故无需转换，我推荐这种写法)

Inet\_addr和inet\_ntoa函数

Unsigned long Inet\_addr(sconst char FAR \* cp);

Inet\_addr可以把一个点分十进制表示的IP（如：192.168.1.101）转换为unsinged long 类型的数据，且该返回值可以直接赋值给S\_addr

Char FAR \* inet\_ntoa(struct in\_addr in);

Inet\_ntoa 从他函数的声明就知道他完成一个和inet\_addr相反的转换 。

**将指定的套接字设为监听模式**

Int listen(SOCKET s,int backlog);

第一个参数 s: 已经创建的套接字描述符

第二个参数 backlog 设置等待连接队伍的最大长度，注意：不是一个端口上可以同时连接的数目。

**Accept函数**

就收客服端发送的连接请求

SOCKET accept(

SOCKET s,

Struct sockaddr FAR \* addr,// 返回请求连接方的IP和端口信息

Int FAR \* addrlen

);

**Send函数**

通过一个已经建立连接的套接字发送数据

Int send(

SOCKET s,//注意：是以建立连接的套接字

Const char FAR \* buf,//目的地IP和端口信息

Int len,

Int falgs//该设置影响发送行为，我们一般设为0

)

**Recv 函数**

Int recv(

SOCKET s,

Char FAR \*buf,//发送数据的缓存地址

Int len,//发送数据长度

Int flags//该设置影响发送行为，我们一般设为0

)

**Connect 函数**

和一个特定的套接字建立连接

Int connect(

SOCKET s,

Const struct sockaddr FAR \* name,//目的地址信息

Int namelen

)

## URLDownloadToFile()

该函数的作用是：从指定URL地址读取内容并将读取到的内容保存到特定的文件里。

函数原形：

HRESULT URLDownloadToFile(

LPUNKNOWN pCaller,

LPCTSTR szURL,

LPCTSTR szFileName,

DWORD dwReserved,

LPBINDSTATUSCALLBACK lpfnCB

);

pCaller //控件的接口,

szURL //要下载的url地址，不能为空

szFileName //下载后保存的文件名

dwReserved //保留字段，必需为0

lpfnCB //下载进度状态回调

## OSVERSIONINFO

OSVERSIONINFO 结构体的定义：

typedef OSVERSIONINFOW OSVERSIONINFO;

typedef struct \_OSVERSIONINFOW {

DWORD dwOSVersionInfoSize; //指定该数据结构的字节大小

DWORD dwMajorVersion; //操作系统的主版本号 5代表2000以上版本

DWORD dwMinorVersion; //操作系统的副版本号 0代表win2000. 1代表winxp

DWORD dwBuildNumber; //操作系统的创建号

DWORD dwPlatformId; //操作系统ID号

WCHAR szCSDVersion[ 128 ]; // Maintenance string for PSS usage 关于操作系统的一些附加信息

} OSVERSIONINFOW, \*POSVERSIONINFOW, \*LPOSVERSIONINFOW, RTL\_OSVERSIONINFOW, \*PRTL\_OSVERSIONINFOW;

**dwMajorVersion :**

|  |  |
| --- | --- |
| Value | Meaning |
| 4 | The operating system is Windows NT 4.0, Windows Me, Windows 98, or Windows 95. |
| 5 | The operating system is Windows Server 2003 R2, Windows Server 2003, Windows XP, or Windows 2000. |
| 6 | The operating system is Windows Vista or Windows Server 2008. |

**dwMinorVersion :**

|  |  |
| --- | --- |
| Value | Meaning |
| 0 | The operating system is Windows Vista, Windows Server 2008, Windows 2000, Windows NT 4.0, or Windows 95. |
| 1 | The operating system is Windows XP. |
| 2 | The operating system is Windows Server 2003 R2, Windows Server 2003, or Windows XP Professional x64 Edition. |
| 10 | The operating system is Windows 98. |
| 90 | The operating system is Windows Me. |

**dwPlatformId :**

|  |  |
| --- | --- |
| Value | Meaning |
| VER\_PLATFORM\_WIN32\_NT  2 | The operating system is Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP, Windows 2000, or Windows NT. |
| VER\_PLATFORM\_WIN32\_WINDOWS  1 | The operating system is Windows Me, Windows 98, or Windows 95. |

**szCSDVersion:**

A null-terminated string, such as "Service Pack 3", that indicates the latest Service Pack installed on the system. If no Service Pack has been installed, the string is empty.

## 注册表操作

RegOpenKeyEx：打开一个指定的注册表键

LONG RegOpenKeyEx(

HKEY hKey, // 需要打开的主键的名称

LPCTSTR lpSubKey, //需要打开的子键的名称

DWORD ulOptions, // 保留，设为0

REGSAM samDesired, // 安全访问标记，也就是权限

PHKEY phkResult // 得到的将要打开键的句柄

)

RegCloseKey：关闭注册表项

LONG RegCloseKey(

HKEY hKey // 释放键的句柄

);

RegCreateKeyEx：创建注册表键，如果该键已经存在，则打开它（注册表键不区分大小写）

LONG RegCreateKeyEx(

HKEY hKey,

LPCTSTR lpSubKey,

DWORD Reserved,

LPTSTR lpClass,

DWORD dwOptions,

REGSAM samDesired,

LPSECURITY\_ATTRIBUTES lpSecurityAttributes,

PHKEY phkResult,

LPDWORD lpdwDisposition

);

RegDeleteKey：删除注册表键值

LONG RegDeleteKey(

HKEY hKey,

LPCTSTR lpSubKey

);

RegQueryValueEx：读取注册表键值

LONG WINAPI RegQueryValueEx(

HKEY hKey,

LPCTSTR lpValueName,

LPDWORD lpReserved,

LPDWORD lpType,

LPBYTE lpData,

LPDWORD lpcbData

);

RegSetValueEx：写入注册表键值

LONG RegSetValueEx(

HKEY hKey,

LPCTSTR lpValueName,

DWORD Reserved,

DWORD dwType,

CONST BYTE \*lpData,

DWORD cbData

);