yesNETWARE.TOC000yesyesyesyesNetware Help FileTRUENetwareyesyes10/11/98

Netware 3x
Netware 3x Functions
Netware 3x Errors

Netware 4x
Netware 4x Functions
Netware 4x Errors

AddExtender(filename)
LastError( )
Net101
NetInfo(requestcode)

Troubleshooting

n3Attach(server-name , user-name, password)
n3CaptureEnd(port-number)
n3CaptureGet(port-number)
n3CapturePrt(server-name, queue-name, port-number, flags)
n3ChgPassword(server-name, user-name, old password, new-password)
n3Detach(server-name)
n3DirAttrGet(dirname)
n3DirAttrSet(dirname, attribs, mode)
n3DirTimeGet
n3DrivePath(local-name)
n3DriveStatus(local-name)
n3FileAttrGet(filename)
n3FileAttrSet(filename, attribs, mode)
n3FileTimeGet
n3GetMapped(server-name)
n3GetNetAddr(server-name, flags)
n3GetUser(server-name)
n3GetUserId(server-name, user-name, format)
n3Logout(server-name)
n3Map(net-path, local-name)
n3MapDelete(local-name)
n3MapDir(net-path, local-name)
n3MapRoot(net-path, local-name)
n3MemberDel(server-name, group-name ,user-name)
n3MemberGet(server-name, group-name ,user-name)
n3MemberSet(server-name, group-name , user-name)
n3MsgSend(server-name, message , user-name)
n3MsgSendAll(server-name, message)
n3ServerInfo(server-name, request #)
n3ServerList(request)
n3UserGroups(server-name, user-name)
n3Version( )

n4Attach(server-name)
n4CaptureEnd(port-number)
n4CaptureGet(port-number)
n4CapturePrt(server-name, queue-name, port-number, flags)
n4ChgPassword(server-name, user-name, old password, new-password)
n4Detach(server-name)
n4DirAttrGet(dirname)
n4DirAttrSet(dirname, attribs, mode)
n4DirTimeGet
n4DrivePath(local-name)
n4DriveStatus(local-name)
n4FileAttrGet(filename)
n4FileAttrSet(filename, attribs, mode)
n4FileTimeGet
n4GetContext
n4GetMapped(server-name)
n4GetNetAddr(server-name, flags)
n4GetUser(server-name)
n4GetUserId(server-name, user-name, format)

n4Login(user-name, password, context, tree)
n4Logout( )
n4LogoutTree
n4Map(net-path, local-name)
n4MapDelete(local-name)
n4MapDir(net-path, local-name)
n4MapRoot(net-path, local-name)
n4MemberDel(group-name ,user-name)
n4MemberGet(group-name ,user-name)
n4MemberSet(group-name , user-name)
n4MsgSend(server-name, message , user-name)
n4MsgSendAll(server-name, message)
n4ObjectList(context, parent, class, mask)
n4ObjectInfo(context, object, request)
n4ObjectProps(context, object, attribute)
n4NameConvert(context, object, format)
n4ServerInfo(server-name, request #)
n4ServerList(request#)
n4SetContext(context, tree)
n4UserGroupEx
n4UserGroups(server-name, user-name)
n4Version( )

WIL Netware Extender Help File

These Windows Interface Language network extenders provide standard support for Novell networks. They may be used in addition with other extenders, such as the Windows for WorkGroups Multinet extender or with each other.

The **Netware extenders** help you to attach, map and manipulate network processes within your Windows System.

Netware 3x functions and **Netware 4x functions** are designed specifically for the separate versions of Netware. These functions can be **used within WIL scripts** or can be **compiled into WIL executables**.

Netware 3x and **Netware 4x** extenders are available for both Windows 3.1 and Windows 95. Windows NT requires the 32 bit Netware 3x or 4x extenders be used in conjunction with the Netware Client 32.

Technical support is available for registered users. If you can't find what you're looking for, or you're having problems with your WIL scripts, be sure to look at the **troubleshooting** section, which has solutions to many of the problems you're likely to encounter.

About WIL Extenders

WIL extender Dlls are special Dlls designed to extend the built-in function set of the WIL processor. These Dlls typically add functions not provided in the basic WIL set, such as network commands for particular networks (Novell, Windows for WorkGroups, LAN Manager and others), MAPI, TAPI, and other important Application Program Interface functions as may be defined by the various players in the computer industry from time to time. These Dlls may also include custom built function libraries either by the original authors, or by independent third party developers. (An Extender SDK is available). Custom extender Dlls may add nearly any sort of function to the WIL language, from the mundane network math or database extensions, to items that can control fancy peripherals, including laboratory or manufacturing equipment.

WIL extenders must be installed separately. Up to 10 extender Dlls may be added. The total number of added items may not exceed 200 functions and constants. The **AddExtender** function must be executed before attempting to use any functions in the extender library. The **AddExtender** function should be only executed once in each WIL script that requires it.

Using WIL Extenders

Accessing the additional functionality available in WIL Extender Dlls is simple. At the top of each script in which WIL Extender commands are to be used add the appropriate extender with the **AddExtender** command.

```
AddExtender(extender filename)
```

The WIL interpreter will search for the the extender DLL's. If no path is specified in the **AddExtender** statement, the WIL interpreter will search the current directory, the windows directory and on directories on the path. The extender DLL's must be available or the **AddExtender** line will return an error. In general, when you run a large exe with embedded extenders, it will extract the extenders to the same directory the compiled exe is in.

The **AddExtender** function should only be executed once for each extender dll in each WIL script that requires it. Per script you can add up to 10 extender Dlls or a combined total of 200 functions.

For example to use both of the NetWare Extenders, two AddExtender lines would appear in the script.

```
AddExtender("wwn3x32i.dll")    ;for the Netware 3 Client32
AddExtender("wwn4x32i.dll")    ;for the Netware 4 Client32
AddExtender("wwn3x16i.dll")    ;for the Netware 3 16-bit Client
AddExtender("wwn4x16i.dll")    ;for the Netware 4 16-bit Client
AddExtender("wwn3y16i.dll")    ;for the Netware 3 IntraNetware Client
AddExtender("wwn4y16i.dll")    ;for the Netware 4 IntraNetware Client
```


Both Netware 3x and Netware 4x have their own specific extenders with separate but similar command sets. Netware 3x functions and Netware 4x functions are designed specifically for the separate versions of Netware. The functions can be used within WIL scripts or can be compiled into WIL executables.

They may be used in addition with other extenders, such as the Windows for WorkGroups Multinet extender or with each other.

Compiling with WIL Extenders

The WinBatch+Compiler has two options for compiling scripts into executables, Large EXE for Standalone PC's and Small EXE for Networked PC's. When any extender functions are used in a script, the corresponding extender must be compiled into the executable, or placed where the executable can access it.

The Large Standalone EXE option of the Compiler has an additonal button. The EXTENDERS button displays a list of extenders which can be chosen and compiled into a Standalone EXE option. More than one extender may be chosen. When a Standalone EXE is launched on a PC it looks for the necessary dll's in the current directory, on the path and in the Windows directory. If the Dll's are not found, they are automatically written into the current directory. If for some reason, they cannot be written to that directory (perhaps the directory is set to be Read Only), the large compiled file will not run.

The DLLs can also be copied into a directory on a computers PATH and the compiled EXE will find them there and run. The Compiler has a Small EXE for Networked PC's option that takes advantage of this.
The DLLs need to be placed on the PATH only once. Subsequent EXE files installed on this same machine can be compiled under the Small EXE option.

It is generally recommended to use the filename without including the path in an AddExtender statement. However, if it is preferable to point to the current directory, (directory in which the WIL executable resides), it can be done.

```
AddExtender( StrCat( DirHome( ), "wwn3x32i.dll" ))
```

Contacting Wilson WindowWare

Wilson WindowWare, Inc.
5421 California Ave. SW
Seattle, WA 98136 USA

Orders: (800) 762-8383
Voice: (206) 938-1740
Fax: (206) 935-7129

Email: info@windowware.com

Registered users of our software get manuals, technical support, use of Wilson WindowWare on-line information services, and special offers on new versions of Wilson WindowWare products.

- Technical Support
- About WIL Extenders

How to get Technical Support

The Wilson WindowWare website is an excellent technical resource. Access to the entire Technical Support Database is at your fingertips. In the Technical Support area use the keyword search to find answers to common problems, alternate scripting methods, and sample code. Or join the Wilson WindowWare Web BBS, a new Web forum. The BBS provides an outlet for registered users to share their experiences with other users.

See the information on registering your copy (found in the WinBatch.hlp or the WinEdit.hlp files) if you haven't done so yet.

The latest versions of our software are available on-line. The places here may change at any time -- check your installation sheet for the most recent addresses.

Internet Web page: http://www.windowware.com

Internet Technical Support Articles & Web BBS: http://techsupt.windowware.com

Internet FTP: ftp.windowware.com in /wwwftp/wilson

- Contacting Wilson WindowWare
- About WIL Extenders
- Troubleshooting

WIL Netware Extender Help File

About this Help File

This extender adds certain network capability to the Windows Interface Language (WIL) processing engine. Please refer to the **WIL Reference Manual** for an introduction to WIL, as well as for complete documentation of the many functions available in WIL and the programs that use it. This help file includes only topics and functions which are exclusive to this particular WIL Extender.

Notational Conventions

Throughout this manual, we use the following conventions to distinguish elements of text:

ALL-CAPS
: Used for filenames.

Boldface
: Used for important points, programs, function names, and parts of syntax that must appear as shown.

`system`
: Used for items in menus and dialogs, as they appear to the user.

```
Small fixed-width
```
: Used for WIL sample code.

Italics
: Used for emphasis, and to liven up the documentation just a bit.

Acknowledgments

This network extender developed by Morrie Wilson and Richard Merit.

Documentation written by Tina Browning.

Netware 3x

Selecting the appropriate extender for use with your system can be a little tricky. Both Netware 3 and Netware 4 have their own extenders and own set of functions. In addition, the extender must match the Windows operating system.

These WIL extenders provide standard support for Novell 3.x networks.

- Netware 3x Functions Errors
- Netware 4x Functions Errors
- Using WIL Extenders
- Compiling with WIL Extenders

For WIN 3.1 and WFW 3.11

wwn3x16i.dll
Other required DLL's: nwcalls.dll

With Windows 3.1 and Windows for WorkGroups use the wwn3x16i.dll, 16 bit Netware 3 extender. In addition to the extender dll, a current version of nwcalls.dll is necessary for full functionality.

```
AddExtender("wwn3x16i.dll")
```

Windows 95 and Windows NT require separate extenders.

For Win95 with Win 3.1 Netware Client
wwn3z32i.dll
Other required DLL's: nwcalls.dll, wwn3z16i.dll

For Windows 95 with a Windows 3.1 Netware Client (not Windows NT) the extender is wwn3z32i.dll. This extender requires two other dll's, nwcalls.dll and wwn3z16i.dll.

```
AddExtender("wwn3z32i.dll")
```

For Win95/NT with Netware Client 32.
wwn3x32i.dll

For Windows 95 and INTEL versions of Windows NT running Netware Client 32 use the wwn3x32i.dll.

```
AddExtender("wwn3x32i.dll")
```

Other required DLL's: nwcalls.dll

This dll is designed to work with Netware Client 32. Netware Client 32 can be downloaded from: *http://netwire.novell.com/home/client/client32*

For WIN 3.1 and WFW 3.11 with the IntraNetware Client

With Windows 3.1 and Windows for WorkGroups use with the IntraNetware Client use the wwn3y16i.dll.

```
AddExtender("wwn3y16i.dll")
```

Netware 3x Functions

The following WIL functions are useful when using Network extenders.

AddExtender(filename)

LastError( )

Net101

NetInfo(requestcode)

The following functions for Netware 3x can be added with the wwn3x16i.dll,wwn3y16i.dll, wwn3z32i.dll or the wwn3x32i.dll.

n3Attach(server-name , user-name, password)

n3CaptureEnd(port-number)

n3CaptureGet(port-number)

n3CapturePrt(server-name, queue-name, port-number, flags)

n3ChgPassword(server-name, user-name, old password, new-password)

n3Detach(server-name)

n3DirAttrGet(dirname)

n3DirAttrSet(dirname, attribs, mode)

n3DirTimeGet(dirname, time-field)

n3DrivePath(local-name)

n3DriveStatus(local-name)

n3FileAttrGet(filename)

n3FileAttrSet(filename, attribs, mode)

n3FileTimeGet(filename, time-field)

n3GetMapped(server-name)

n3GetNetAddr(server-name, flags)

n3GetUser(server-name)

n3GetUserId(server-name, user-name, format)

n3Logout(server-name)

n3Map(net-path, local-name)

n3MapDelete(local-name)

n3MapDir(net-path, local-name)

n3MapRoot(net-path, local-name)

n3MemberDel(server-name, group-name ,user-name)

n3MemberGet(server-name, group-name ,user-name)

- Netware 3x Errors
- Netware 4x Functions Errors
- Troubleshooting
- Using WIL Extenders
- Compiling with WIL Extenders

n3MemberSet(server-name, group-name , user-name)
n3MsgSend(server-name, message , user-name)
n3MsgSendAll(server-name, message)
n3ServerInfo(server-name, request #)
n3ServerList(request)
n3UserGroups(server-name, user-name)
n3Version( )

Netware 4x

Selecting the appropriate extender for use with your system can be a little tricky. Both Netware 3 and Netware 4 have their own extenders and own set of functions. In addition, the extender must match the Windows operating system.

These WIL extenders provide standard support for Novell 4.x networks.

For WIN 3.1 and WFW 3.11

wwn4x16i.dll
Other required DLL's: nwcalls.dll, nwnet.dll, nwlocale.dll.

With Windows 3.1 and Windows for WorkGroups use the wwn4x16i.dll, 16 bit Netware 4 extender. In addition to the extender dll, a current version of nwcalls.dll, nwnet.dll and nwlocale.dll are necessary for full functionality.

```
AddExtender("wwn4x16i.dll")
```

For Windows 95 and INTEL versions of Windows NT running Netware Client 32 use the wwn4x32i.dll.

```
AddExtender("wwn4x32i.dll")
```

For Win95/NT with Netware Client 32.

wwn4x32i.dll
Other required DLL's: nwcalls.dll

This dll is designed to work with Netware Client 32. Netware Client 32 can be downloaded from: *http://netwire.novell.com/home/client/client32.*

For WIN 3.1 and WFW 3.11 with the IntraNetware Client

With Windows 3.1 and Windows for WorkGroups use with the IntraNetware Client use the wwn4y16i.dll.

```
AddExtender("wwn4y16i.dll")
```

Netware 4x Functions

The following WIL functions are useful when using Network extenders.

AddExtender(filename)

LastError( )

Net101

NetInfo(requestcode)

The following functions for Netware 4x can be added with the wwn4x16i.dll, wwn4y16i.dll or the wwn4x32i.dll .

n4Attach(server-name)

n4CaptureEnd(port-number)

n4CaptureGet(port-number)

n4CapturePrt(server-name, queue-name, port-number, flags)

n4ChgPassword(server-name, user-name, old password, new-password)

n4Detach(server-name)

n4DirAttrGet(dirname)

n4DirAttrSet(dirname, attribs, mode)

n4DirTimeGet(dirname, time-field)

n4DrivePath(local-name)

n4DriveStatus(local-name)

n4FileAttrGet(filename)

n4FileAttrSet(filename, attribs, mode)

n4FileTimeGet(filename, time-field)

n4GetContext(request)

n4GetMapped(server-name)

n4GetNetAddr(server-name, flags)

n4GetUser(server-name)

n4GetUserId(server-name, user-name, format)

n4Login(user-name, password, context, tree)

n4Logout( )

n4LogoutTree(context, tree)

n4Map(net-path, local-name)

n4MapDelete(local-name)

n4MapDir(net-path, local-name)

n4MapRoot(net-path, local-name)

- Netware 4x Errors
- Netware 3x Functions Errors
- Troubleshooting
- Using WIL Extenders
- Compiling with WIL Extenders

n4MemberDel(group-name ,user-name)
n4MemberGet(group-name ,user-name)
n4MemberSet(group-name , user-name)
n4MsgSend(server-name, message , user-name)
n4MsgSendAll(server-name, message)
n4NameConvert(context, object, format)
n4ObjectList(context, parent, class, mask)
n4ObjectInfo(context, object, request)
n4ObjectProps(context, object, attribute)
n4ServerInfo(server-name, request #)
n4ServerList(request#)
n4SetContext(context, tree)
n4UserGroups(server-name, user-name)
n4UserGroupEx(server-name, user-name, context)
n4Version( )

AddExtender(filename)

Installs a WIL extender Dll.

Syntax:

AddExtender(filename)

Parameters:

(s) filename	WIL extender Dll filename.

Returns:

(i)	**@TRUE** if function succeeded. **@FALSE** if function failed.

WIL extender Dlls are special Dlls designed to extend the built-in function set of the WIL processor. These Dlls typically add functions not provided in the basic WIL set, such as network commands for particular networks (Novell, Windows for WorkGroups, LAN Manager and others), MAPI, TAPI, and other important Application Program Interface functions as may be defined by the various players in the computer industry from time to time. These Dlls may also include custom built function libraries either by the original authors, or by independent third party developers. (An Extender SDK is available). Custom extender Dlls may add nearly any sort of function to the WIL language, from the mundane network, math or database extensions, to items that can control fancy peripherals, including laboratory or manufacturing equipment.

Use this function to install extender Dlls as required. Up to 10 extender Dlls may be added. The total number of added items may not exceed 200 functions and constants. The **AddExtender** function must be executed before attempting to use any functions in the extender library. The **AddExtender** function should be only executed once in each WIL script that requires it.

The documentation for the functions added are supplied either in a separate manual or disk file that accompanies the extender Dll.

Example:

```
; Add vehicle radar processing dll controlling billboard visible to
; motorists, and link to enforcement computers.
; The WIL Extender SPEED.DLL adds functions to read a radar speed
; detector(GetRadarSpeed) , put a message on a billboard visible to
; the motorist (BillBoard), take a video of the vehicle (Camera), and
; send a message to alert enforcement personnel (Alert) that a
; motorist in  violation along with a picture id number to help
; identify the offending vehicle and the speed which it was going.
;
AddExtender("SPEED.DLL")
BillBoard("Drive Safely")
While @TRUE
    ; Wait for next vehicle
    while GetRadarSpeed()<5 ; if low, then just radar noise
       Yield            ; wait a bit, then look again
    endwhile
    speed=GetRadarSpeed()      ; Something is moving out there
    if speed < 58
       BillBoard("Drive Safely")  ; Not too fast.
    else
```

```
        if speed < 63
          BillBoard("Watch your Speed")   ; Hmmm a hot one
        else
          if speed < 66
            BillBoard("Slow Down")  ; Tooooo fast
          else
            BillBoard("Violation  Pull Over")
            pictnum = Camera() ; Take Video Snapshot
            Alert(pictnum, speed); Pull this one over
          endif
        endif
      endif
    endwhile
```

See Also:

DllCall *(found in main WIL documentation)*

LastError( )

Returns the most-recent error encountered during the current WIL program.

Syntax:

LastError( )

Parameters:

None

Returns:

(i) most-recent WIL error code encountered.

In addition to the normal behavior of the LastError function documented in the WIL Reference Guide, if the most recent error occurred in a WIL Extender, then a number assigned by the Extender will be returned. The numbers are documented in the appendix of this Extender document.

It may be possible to obtain error numbers not documented. The "Notes" section of the WIL manual has been provided to allow you to keep records of undocumented error codes.

Example:

```
;Access script with some error checking
;
OnCancel="Exit"                       ; Setup default "cancel" processing

retcode = AddExtender("wwn3x32i.dll")      ;Load in Novell 3 extender
if retcode == 0
   ;This code should not even get the chance to execute.
   ;Fail-safe error checking here
   Message("Error","Failed to load Novell 3 extender")
endif

MyServer="\\DEPT07"
UserID="FRED"

ErrorMode(@OFF)        ;Tell WIL we want to handle errors in script

:TRYPSWD
OnCancel = "goto DETACH"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
OnCancel = "exit"
retcode = n3Attach(MyServer, UserID, Pswd)
if retcode == 0
   errcode=LastError()
   if errcode == 128
      Message("Bad Password Error","Bad password supplied for Userid %UserID%")
      goto TRYPSWD
   endif
   Message("Login Error %errcode%","Login Failure")
   if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
   exit
endif

; Find drive to map.  But don't use W, X, Y, or Z just to
; make it more interesting.
drives = DiskScan(0)
for I=1 to 4
   nono = strcat( num2char( char2num("V") + I) , ":")
   a = ItemLocate( nono,  drives, " ")
   if a!=0 then drives = ItemDelete(a, drives, " ")
next

if ItemCount(drives, " ") == 0
   Message("Error", "No drives available for mapping")
   if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
   exit
 endif

usedrive=ItemExtract(1,drives," ")

n3Map("\\DEPT07\SYS\Excel", usedrive)
errcode=LastError()
if errcode != 0      ; Map Failue
   Message("Map Error %errcode%","Map to %usedrive% failed")
   if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
   exit
endif

OrigDir=DirGet()
DirChange(strcat(usedrive,"\"))
RunWait("EXCEL.EXE","/E")
errcode = LastError()
if errcode != 0
```

```
   Message("RunWait Failed ???","Errorcode=%errcode%")
   ;drop thru to disconnect
endif
DirChange(OrigDir)
n3MapDelete(usedrive)

:DETACH
; Just in case user has other mappings to server, only
; detach (logout) from server if no other mappings exist
if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
exit

:CANCEL
%OnCancel%
Message("Error","Oncancel variable improperly set up")
exit
```

See Also:

Debug, ErrorMode *(both found in main WIL documentation)*

Net101

All network functionality for WIL is performed via "WIL Extenders", add-on Dlls for WIL, which contain Network commands for assorted networks.

NetInfo is the only WIL network function. It returns the types of the networks currently active on the local machine, and can be used to help determine which network extenders should be loaded in multi-network environments.

Documentation for the various network extenders are found either in a manual for a particular extender or in an associated disk file.

See Also:

NetInfo, AddExtender, DllCall *(found in main WIL documentation)*

NetInfo(requestcode)

Determines network(s) installed.

Syntax:

NetInfo(requestcode)

Parameters:

(i) requestcode	0 for primary network name. 1 for secondary subnet list.

Returns:

(s)	Primary network name for request code 0, or Secondary network list for request code 1.

Use this function to determine the network type(s) running on a workstation. When running in a mixed network environment, it may be important to be able to determine the types of networks running on a workstation so as to be able to load the appropriate network extender Dlls and issue the corresponding commands.

NetInfo(0) will return the name of the primary network, or will return "**MULTINET**" , which indicates the Windows multinet driver is active and the secondary subnet list should be queried. **NetInfo**(0) will return one of the following strings:

NetInfo(0) return values:

NONE	No network installed
MULTINET	Multinet driver installed, see subnet codes.
MSNET	Microsoft Network
LANMAN	LAN Manager
NETWARE	Novell NetWare
VINES	Banyan Vines
10NET	10 Net
LOCUS	Locus
SUNPCNFS	SUN PC NFS
LANSTEP	LAN Step
9TILES	9 Tiles
LANTASTIC	Lantastic
AS400	IBM AS/400
FTPNFS	FTP NFS
PATHWORK	DEC PathWorks
OTHER1	Other (code 1)
OTHER2	Other(code 2)
UNKNOWN	Other (unknown)

If **NetInfo**(0) returned "**MULTINET**" then **NetInfo**(1) will return one or more of the following in a space delimited list:

NetInfo(1) return values:

NONE	No networks active
MSNET	Microsoft Network
LANMAN	LAN Manager

WINNET	Windows Network (Windows for Workgroups, etc)
NETWARE	Novell Netware
VINES	Banyan Vines
OTHER2	Other (code 0x20)
OTHER4	Other (code 0x40)
OTHER8	Other (code 0x80)

32 Bit Windows

NetInfo(0) will always return the string "WINNT" for 32 bit Windows platforms, regardless of whether the platform is Windows95 or Windows NT.

Under Windows 95, NetInfo(1) will return a list of installed network client ID's, delimited with the standard file delimiter (by default, a tab).

Possible client ID's, with their corresponding descriptions, are:

Client ID	**Description**
3OPEN	3Com 3+Open (all versions)
3SHARE	3Com 3+Share (all versions)
DLR	IBM OS/2 LAN Server (versions below 1.2)
DLR12	IBM OS/2 LAN Server (version 1.2)
DLR13	IBM OS/2 LAN Server (versions 1.2, 1.3, and 1.2 without /API)
DLR13CSD	IBM OS/2 LAN Server (version 1.3 CSD 5015/5050)
DLR20	IBM OS/2 LAN Server (version 2.0)
FTPNFS	FTP Software NFS Client (InterDrive 95)
LANMAN	Microsoft Real Mode LAN Manager
LANT5	Artisoft LANtastic (version 5.X and above)
MSNET	Real mode MS-Net Compatible
NETWARE3	Novell NetWare (Workstation Shell 3.X [NETX])
NETWARE4	Novell NetWare (Workstation Shell 4.0 and above [VLM])
NOVELL32	Novell NetWare Client 32
NWREDIR	Client for NetWare Networks
PATHWKS	DEC PATHWORKS (versions below 4.0)
PATHWKS40	DEC PATHWORKS (version 4.x)
PCLP	IBM PC LAN Program (all versions)
PCNFS50	SunSoft PC-NFS (version 5.0)
VINES552	Banyan DOS/Windows 3.1 client
VREDIR	Client for Microsoft Networks

Under Windows NT, NetInfo(1) will return a list of installed network provider ID's, delimited with the standard file delimiter (by default, a tab).

Possible providers, with their corresponding descriptions, are:

Provider ID	**Description**
LanmanWorkstation	Microsoft Windows Network
NetWareWorkstation	NetWare Services
NWCWorkstation	NetWare or Compatible Network

Example:

```
a=NetInfo(0)
if a=="MULTINET"
   b=NetInfo(1)
   count=ItemCount(b," ")
   Message("Multinet supporting  %count% networks", b)
else
   Message("Installed Network", a)
endif
```

See Also:

Net101, AddExtender, DllCall *(found in main WIL documentation)*

n3Attach(server-name , user-name, password)

Performs a NetWare Attach to an individual file server.

Syntax:

n3Attach(server-name , user-name, password)

Parameters:

(s) server-name	name of a network file server.
(s) user-name	name of the current user.
(s) password	password required to access server OR " " for no password.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Performs a NetWare Attach to an individual file server. Does NOT run any login scripts.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn3x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n3Attach(MyServer, UserID, Pswd)
n3Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n3MapDelete("Q:")
if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
```

See Also:

n3Detach

n3CaptureEnd(port-number)

Ends a printer capture.

Syntax:

n3CaptureEnd(port-number)

Parameters:

(i) port-number	number from 1 to 9 (where 1 = "LPT1", etc.)

Returns:

(i)	**@TRUE** if successful.

Example:

```
AddExtender("wwn3x32i.dll")
;Assuming user is attached to and logged into server
;
a=n3CapturePrt("\\CHICAGO","PRINTERC",1,(Tabsize=8%@TAB%Numcopies=2")
RunWait("notepad.exe", "/p readme.txt")
n3CaptureEnd(1)
```

See Also:

n3CapturePrt

n3CaptureGet(port-number)

Returns the name of the print queue associated with a printer capture.

Syntax:

n3CaptureGet(port-number)

Parameters:

(i) port-number	number from 1 to 9 (where 1 = "LPT1", etc.).

Returns:

(i)	name of a print queue.

Example:

```
AddExtender("wwn3x32i.dll")
;Assuming user is attached to and logged into server
queue = n3CaptureGet(1)
Message("LPT1 is captured to", queue)
```

See Also:

n3CaptureEnd, n3CapturePrt

n3CapturePrt(server-name, queue-name, port-number, flags)

Captures a local printer port to a Netware printer queue.

Syntax:

n3CapturePrt(server-name, queue-name, port-number, flags)

Parameters:

(s) server-name	name of a network file server.
(s) queue-name	name of a network printer queue.
(s) port-number	number from 1 to 9 (where 1 = "LPT1", etc.).
(s) flags	a tab-delimited list of keyword=value pairs,

Returns:

(i)	**@TRUE** if successful.

Flags denotes a tab-delimited list of keyword=value pairs. For example;

"key1=value1%@TAB%key2=value2%@TAB%key3=value3..."

Valid keywords are:

JobDescription

Null-terminated ASCII description of the contents or purpose of the job. The NetWare DOS Requester uses only 13 bytes of this member. This member is used only by the DOS Requester and OS/2.

JobControlFlags

Set of queue job control flags affecting the way a queue server processes a queue job. This member is used only by the DOS Requester and OS/2. Returns 0 under DOS/Windows. Under OS/2 JobControlFlags is defined as 1024 (print interrupted capture), and bits 0, 1, and 2 must be 0.

TabSize

Returns a value between 1 and 18 inclusive, indicating tab size. The default setting is 8.

NumCopies

Number of copies of the captured file the printer prints (The maximum number of copies is 255 for netx.com, and 65536 for OS/2 and the DOS Requester). The default setting is 1. If NumCopies is 0, nothing prints.

PrintFlags

Value	Flag Name	Description
4	Release Job	If set, the print job is released for printing if the capture is interrupted by a loss of connection time to the server.
8	Suppress Form Feed	If set, the print service suppresses automatic form feed after the print job is printed.
64	Text File	If set, tab size and other printer control sequences are interpreted by the print service. If not set, the job is interpreted as a byte stream.

128 Print Banner If set, the print service precedes the print job with a banner page.

MaxLines

Maximum lines per page.

MaxChars

Maximum characters per line.

FormName

Name of the form a user must mount in the printer to print files captured to the LPT device. If the form currently mounted in the printer differs from the form name returned in this field, the NetWare server console displays a message instructing the console operator to mount the correct form.

FormType

Type of form (0 to 255) a user must mount in the printer to print files captured to the LPT device. If the form currently mounted in the printer differs from the form type returned in this field, the NetWare server console displays a message instructing the console operator to mount the correct form. The default form is 0.

BannerText

13-byte string containing the name appearing on the bottom half of a banner page. All letters are upper case.

FlushCaptureTimeout

(0 to 3,640) starts counting down every time an application executes a print command (int 17h). When the time-out expires, the server flushes the capture file and queues it at a printer. If an application executes a second print command before the first time-out expires, the time-out starts over from the original value. Each tick of the capture time-out is approximately one second. The range is 1 to 1000. The default time-out is 0---no time-out. (Not valid in OS/2.)

FlushCaptureOnClose

When FlushCaptureOnClose is enabled (0), the server flushes the capture file when the application ends the capture of the default LPT device. (Default is 0.) Any other value means disabled. (Not valid in OS/2.)

Example:

```
AddExtender("wwn3x32i.dll")
;Assuming user is attached to and logged into server
;
a=N3CapturePrt("\\CHICAGO","PRINTERC",1,"Tabsize=8%@TAB%Numcopies=2")
RunWait("notepad.exe", "/p readme.txt")
N3CaptureEnd(1)
```

See Also:

n3CaptureEnd

n3ChgPassword(server-name, user-name, old-password, new-password)

Changes a user's password.

Syntax:

n3ChgPassword(server-name, user-name, old-password, new-password)

Parameters:

(s) server-name	name of a network file server or empty string.
(s) user-name	name of the current user.
(s) old-password	the old password.
(s) new-password	a new password.

Returns:

(i)	**@TRUE** if successful.

In order to use n3ChgPassword to change the password, you must be logged into the specified server. If you have supervisor or equivalent rights, a blank string ("") can be specified for "old-password".

Passwords are NOT case-sensitive.

Example:

```
AddExtender("wwn3x32i.dll")
;
;Assuming user is attached to and logged into server
;
 N3ChgPassword("\\CHICAGO","joe","spam","maps")
```

n3Detach(server-name)

Logs out and detaches from one or all NetWare 3.x network file servers.

Syntax:

n3Detach(server-name)

Parameters:

(s) server-name	name of a network file server or empty string.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

This function will logout and detach a user from a Novell 3 server. If a file server name is specified, then the user will be logged out of and detach from that particular file server. If the server name is an empty string ("") then the user will be logged out of and detach from all NetWare 3.x file servers. This function does not affect attachments to NetWare 4 Directory Services servers.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn3x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n3Attach(MyServer, UserID, Pswd)
n3Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n3MapDelete("Q:")
if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
```

See Also:

n3Attach, n3Logout

n3DirAttrGet(dirname)

Returns NetWare directory attributes

Syntax:

n3DirAttrGet(dirname)

Parameters:

(s) dirname	directory pathname whose attributes are to be determined.

Returns:

(i)	the attributes of the specified directory pathname.

Returns a number which is the sum of all attributes set for the specified directory. Use the bitwise AND operator (&) to determine if a specific attribute is set. See below for a list of attribute constants.

Attribute Constants :

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:

```
AddExtender("wwn3x32i.dll")
dirname = "n:\public"
attrs = n3DirAttrGet(dirname)
If attrs & @attr_H
  Message(dirname, "is hidden")
Else
  Message(dirname, "is not hidden")
Endif
```

See Also:

n3DirAttrSet, n3FileAttrSet, n3FileAttrGet

n3DirAttrSet(dirname, attribs, mode)

Sets NetWare directory attributes

Syntax:

n3DirAttrSet(dirname, attribs, mode)

Parameters:

(s) dirname	directory pathname.
(s) attribs	one or more NetWare directory attribute constants (see below)
(s) mode	**@ON**, specified attributes are set. **@OFF**, specified attributes are removed.

Note: The 'execute-only' attribute cannot be removed.

If multiple attributes are specified, they should be combined using the bitwise OR operator.

Attribute Constants :

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:

```
AddExtender("wwn3x32i.dll")
dirname = "n:\public"
; set 'hidden' and 'system' attributes
n3DirAttrSet(dirname, @attr_H | @attr_Sy, @ON)
; remove 'shareable' attribute
n3DirAttrSet(dirname, @attr_Sh, @OFF)
```

See Also:

n3DirAttrGet, n3DirAttrSet, n3FileAttrSet, n3FileAttrGet

n3DirTimeGet(dirname, time-field)

Gets Netware time information for a directory.

Syntax:

n3DirTimeGet(dirname, time-field)

Parameters:

(s) dirname	must specify a single directory name (no wildcards).
(i) time-field	can be one of the following: 1 directory created 2 directory last modified 3 --- (not used) 4 directory last archived

Returns:

(s)	the requested directory time in YmdHms format (with a 4-digit year), or a blank string ("") if the requested time field is not set. (yyy:mm:dd:hh:mm:ss…)

Example:

```
dirtime=n3DirTimeGet( "C:\temp", 1)
message("Directory creation time is:", dirtime)
```

See Also:

n3FileTimeGet,

n3DrivePath(local-name)

Returns the network resource associated with the local-name.

Syntax:

n3DrivePath(local-name)

Parameters:

(s) local-name	local drive name.

Returns:

(s)	UNC network path associated with the local name.

This function will return a UNC-style network path that a local drive is mapped to. If there is no valid NetWare mapping, then an empty string will be returned.

Example:

```
AddExtender("wwn3x32i.dll")
netpath = n3DrivePath("Q:")
Message("Q: is mapped to", netpath)
```

See Also:

n3DriveStatus, n3GetMapped, n3ServerList

n3DriveStatus(local-name)

Returns a status code number indicating the type of connection, if any, associated with a local-name.

Syntax:

n3DriveStatus(local-name)

Parameters:

(s) local-name	local drive name.

Returns:

(i)	a status code bitmask.

This function returns information about a local drive. It can determine if the drive is unmapped, already mapped, or belongs to another network.

Bit Value	Bit Definition
0	Unmapped, free drive
1	Local Free Drive
2	Local Drive
4	Network Drive
8	PNW Drive
16	Netware Drive

Common status codes are:

0	Free Drive - Mapable
3	Free Drive - Mapable
6	Mapped local drive belonging to a Non-Novell network

Example:

```
AddExtender("wwn3x32i.dll")
for d = 0 to 25
   drive = strcat( num2char( char2num("A") + d ), ":" )
   stat = n3DriveStatus(drive)
   if stat == 0 || stat == 3
      Message(drive, "is a free, mapable drive")
   endif
   if stat == 23
      path = n3DrivePath(drive)
      Message(drive, "is a mapped Novell drive, mapped to %@CRLF%path")
   endif
next
```

See Also:

n3Drivepath, n3GetMapped, n3ServerList

n3FileAttrGet(filename)

Returns NetWare file attributes

Syntax:

n3FileAttrGet(filename)

Parameters:

(s) filename	a file name, which may include a full path, and which may *not* include wildcards.

Returns:

(i)	sum of all attributes set.

Returns a number which is the sum of all attributes set for the specified file. Use the bitwise AND operator (&) to determine if a specific attribute is set. See below for a list of attribute constants.

Attribute Constants :

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:

```
AddExtender("wwn3x32i.dll")
filename = "n:\public\nwadmin.exe"
attrs = n3FileAttrGet(filename)
If attrs & @attr_H
  Message(filename, "is hidden")
Else
  Message(filename, "is not hidden")
Endif
```

See Also:

n3FileAttrSet,

n3FileAttrSet(filename, attribs, mode)

Sets NetWare file attributes

Syntax:
n3FileAttrSet(filename, attribs, mode)

Parameters:

(s) filename	a file name, which may include a full path, and which may include wildcards
(i) attribs	one or more NetWare file attribute constants (see below for list).
(i) mode	@ON, specified attributes are set. @OFF, specified attributes are removed.

Returns:

(i)	always 1.

Note: The 'execute-only' attribute cannot be removed.

If multiple attributes are specified, they should be combined using the bitwise OR operator.

Attribute Constants :

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:
```
AddExtender("wwn3x32i.dll")
filename = "n:\public\nwadmin.exe"
; set 'hidden' and 'system' attributes
n3FileAttrSet(filename, @attr_H | @attr_Sy, @ON)
; remove 'shareable' attribute
n3FileAttrSet(filename, @attr_Sh, @OFF)
```

See Also:
n3FileAttrGet

n3FileTimeGet(filename, time-field)

Gets Netware time information for a file.

Syntax:

n3FileTimeGet(filename, time-field)

Parameters:

(s) filename	must specify a single file name (no wildcards).
(i) time-field	can be one of the following: 1 file created 2 file last modified 3 file last accessed 4 file last archived

Returns:

(s)	the requested file time in YmdHms format (with a 4-digityear), or a blank string ("") if the requested time field is not set.

Example:

```
filetime=n3FileTimeGet("c:\temp\temp.txt", 1)
Message("file created", filetime)
```

See Also:

n3DirTimeGet,

n3GetMapped(server-name)

Returns a tab delimited list of mapped Novell drives.

Syntax:

n3GetMapped(server-name)

Parameters:

(s) server-name	name of a network file server or empty string.

Returns:

(s)	tab delimited list of drives mapped to the specified server.

This function will interrogate drives A thru Z, and will return a list of drives mapped to Novell servers. If there is no valid NetWare mapping, then an empty string will be returned. If an empty string ("") is provided for the server name, then all valid mapped Novell drives will be returned.

Example:

```
addExtender("wwn3x32i.dll")
MappedDrives = n3GetMapped("\\SERV01")
Message("Mapped drives on \\SERV01", MappedDrives)
```

See Also:

n3DriveStatus, n3DrivePath

n3GetNetAddr(server-name, flags)

Returns the Internet work address of the current workstation.

Syntax:

n3GetNetAddr(server-name, flags)

Parameters:

(s) server-name	name of a network file server or empty string.
(i) flags	0 - reserved for future use.

Returns:

(i)	string of hex bytes in the format: x1x2x3x4:y1y2y3y4y5y6:z1z2

If a null string ("") is used for the server-name, the default server will be used.

The Internet work address of the current workstation is returned in the form of

x1x2x3x4:y1y2y3y4y5y6:z1z2

where:

x1x2x3x4	= 4-byte network address
y1y2y3y4y5y6	= 6-byte net node address (physical address of workstation's LAN board)
z1z2	= socket

Example:

```
AddExtender("wwn3x32i.dll")
;
;Assuming user is attached to and logged into server
;
 a=n3GetNetAddr("\\CHICAGO",0)
 message("Net Address is","%a%")
```

See Also:

n3GetUser, n3DrivePath, n3DriveStatus, n3ServerList, Environment *(found in main WIL documentation)*

n3GetUser(server-name)

Determines the currently logged on user name on the specified server.

Syntax:

n3GetUser(server-name)

Parameters:

(s) server-name	name of a network file server.

Returns:

(s)	a user name.

This function will return the currently logged on user name on a specified Novell server. If no user is logged on, a null string ("") will be returned.

Example:

```
AddExtender("wwn3x32i.dll")
;
;Assuming user is attached to and logged into server
;
Who = n3GetUser("\\DEPT07")
Message("I am logged into \\DEPT07 as", Who)
;
; Also Note this may work, depending on your setup
EnvWho = Environment("USER")
Message("Environment variable USER is",EnvWho)
```

See Also:

n3DrivePath, n3DriveStatus, n3GetUserId, n3ServerList, Environment *(found in main WIL documentation)*

n3GetUserId(server-name, user-name, format)

Returns the object ID corresponding to the specified user name.

Syntax:

n3GetUserId(server-name, user-name, format)

Parameters:

(s) server-name	name of a network file server or empty string.
(s) user-name	name of the current user.
(i) format	the format for the return; see below.

Returns:

(i)	object ID for specified user.

"Format" specifies the format in which the ID is returned:

Request #	Format
0	Decimal number, in internal Netware format
1	Hexadecimal string, such as used for mail directories

Example:

```
AddExtender("wwn3x32i.dll")

;Assuming user is attached to and logged into server

userid=N3GetUserId("\\CHICAGO","joe",0)
message("UserID is","%userid%")
```

See Also:

n3GetUser,n3DrivePath, n3DriveStatus, n3ServerList, Environment *(found in main WIL documentation)*

n3Logout(server-name)

Logs out but does not detach from one or all NetWare 3.x network file servers.

Syntax:

n3Logout(server-name)

Parameters:

(s) server-name	name of a network file server or empty string.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

This function will logout a user but not detach from a Novell 3 server. If a file server name is specified, then the user will be logged out of that particular file server. If the server name is an empty string ("") then the user will be logged out of all NetWare 3.x file servers. This function does not affect attachments to NetWare 4 Directory Services servers.

Example:

```
AddExtender("wwn3x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n3Attach(MyServer, UserID, Pswd)
n3Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n3MapDelete("Q:")
n3Logout(MyServer)
```

See Also:

n3Attach, n3Detach

n3Map(net-path, local-name)

Root maps a drive to a resource specified by a UNC pathname.

Syntax:

n3Map(net-path, local-name)

Parameters:

(s) net-path	UNC net resource.
(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Root maps a drive to a resource specified by a fully qualified UNC filename. Must be either logged into the server, or specified net resource must not require a login. If drive is already mapped to a Novell server, the prior mapping will take effect.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn3x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n3Attach(MyServer, UserID, Pswd)
n3Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n3MapDelete("Q:")
if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
```

See Also:

n3MapDelete, n3MapDir, n3DrivePath, n3DriveStatus,

n3MapDelete(local-name)

Removes a drive mapping.

Syntax:

n3MapDelete(local-name)

Parameters:

(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

This function removes a drive mapping.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn3x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n3Attach(MyServer, UserID, Pswd)
n3Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n3MapDelete("Q:")
if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
```

See Also:

n3Map, n3DrivePath, n3DriveStatus,

n3MapDir(net-path, local-name)

Maps a drive to a resource specified by a UNC pathname.

Syntax:

n3MapDir(net-path, local-name)

Parameters:

(s) net-path	UNC net resource.
(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Maps a drive to a resource specified by a fully qualified UNC filename. Must be either logged into the server, or specified net resource must not require a login. If drive is already mapped to a Novell server, the prior mapping will be deleted and the new mapping will take effect.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn3x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n3Attach(MyServer, UserID, Pswd)
n3MapDir("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n3MapDelete("Q:")
if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
```

See Also:

n3MapDelete, n3Map, n3MapRoot, n3DrivePath, n3DriveStatus,

n3MapRoot(net-path, local-name)

Root maps a drive to a resource specified by a UNC pathname.

Syntax:

n3MapRoot(net-path, local-name)

Parameters:

(s) net-path	UNC net resource.
(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Root maps a drive to a resource specified by a fully qualified UNC filename. Must be either logged into the server, or specified net resource must not require a login. If drive is already mapped to a Novell server, the prior mapping will take effect.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn3x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n3Attach(MyServer, UserID, Pswd)
n3MapRoot("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n3MapDelete("Q:")
if n3GetMapped(MyServer)=="" then n3Detach(MyServer)
```

See Also:

n3MapDelete, n3MapDir, n3DrivePath, n3DriveStatus,

n3MemberDel(server-name, group-name ,user-name)

Deletes the specified user from the specified group on the specified server.

Syntax:

n3MemberDel(server-name, group-name ,user-name)

Parameters:

(s) server-name	name of a network file server.
(s) group-name	name of the group.
(s) user-name	name of the current user.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Assuming that the person running this script has sufficient authority to delete users from the specified group, this function will delete the specified user from the group.

Example:

```
AddExtender("wwn3x32i.dll")
;
;Assuming operator is attached to and logged into server
;
n3MemberDel("\\DEPT07", "MAUI SALES", "BSMITH")
```

See Also:

n3MemberGet, n3MemberSet

n3MemberGet(server-name, group-name ,user-name)

Determines if the specified user is a member of the specified group on the specified server.

Syntax:

n3MemberGet(server-name, group-name ,user-name)

Parameters:

(s) server-name	name of a network file server.
(s) group-name	name of the group.
(s) user-name	name of the current user.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Assuming that the person running this script has sufficient authority to query members of the specified group, this function will allow the person to determine if the user is a member of the specified group or not.

Example:

```
AddExtender("wwn3x32i.dll")
;
;Assuming operator is attached to and logged into server
;
resp = n3MemberGet("\\DEPT07", "NOME SALES", "BSMITH")
if resp == 0 then resp2 = "is NOT"
else resp2 = "is"
Message("NOME SALES","BSMITH %resp2% a member.")
```

See Also:

n3MemberSet, n3MemberDel

n3MemberSet(server-name, group-name , user-name)

Sets the specified user as a member of the specified group on the specified server.

Syntax:

n3MemberSet(server-name, group-name , user-name)

Parameters:

(s) server-name	name of a network file server.
(s) group-name	name of the group.
(s) user-name	name of the current user.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Assuming that the person running this script has sufficient authority to add users to the specified group, this function will add the specified user to the group.

Example:

```
AddExtender("wwn3x32i.dll")
;
;Assuming operator is attached to and logged into server
;
n3MemberSet("\\DEPT07", "NOME SALES", "BSMITH")
```

See Also:

n3MemberDel, n3MemberGet

n3MsgSend(server-name, message , user-name)

Sends a message (max 56 characters) to the specified user.

Syntax:

n3MsgSend(server-name, message , user-name)

Parameters:

(s) server-name	name of a network file server.
(s) message	message to be sent.
(s) user-name	name of the user to whom the message is being sent.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Basic short-form e-mail. Only works if user is logged on.

Example:

```
AddExtender("wwn3x32i.dll")
Msg = AskLine("My Mailing List","Enter Short Message","")
n3MsgSend("\\DEPT07", Msg, "ERINP")
n3MsgSend("\\DEPT07", Msg, "KARINW")
n3MsgSend("\\DEPT07", Msg, "TINAB")
n3MsgSend("\\DEPT07", Msg, "CAROLK")
n3MsgSend("\\DEPT07", Msg, "LAURAW")
n3MsgSend("\\DEPT07", Msg, "DONNAW")
n3MsgSend("\\DEPT07", Msg, "CUPCAKE")
```

See Also:

n3MsgSendAll

n3MsgSendAll(server-name, message)

Sends a message (max 56 characters) to all logged on users.

Syntax:

n3MsgSendAll(server-name, message)

Parameters:

(s) server-name	name of a network file server.
(s) message	message to be sent.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Server-wide broadcasting to all logged in users. Actually only gets the first 300 or so.

Example:

```
AddExtender("wwn3x32i.dll")
n3MsgSendAll("\\DEPT07","Blue Chevy, License 237-EKL, Lights on in back lot")
```

See Also:

n3MsgSend

n3ServerInfo(server-name, request #)

Returns information on the specified Netware server.

Syntax:

n3ServerInfo(server-name, request #)

Parameters:

(s) server-name	name of a network file server or empty string.
(i) request #	name of the current user.

Returns:

(i)	the requested server information.

Request	Meaning
1	major NetWare version number
2	minor NetWare version number
3	revision number of the NetWare OS on NetWare server
4	maximum number of connections the server will support
5	highest number of connections simultaneously in use
6	number of connections the server currently has in use
7	maximum number of volumes the server will support
8	SFT level the server supports
9	TTS Level of NetWare server operating system

Example:

```
AddExtender("wwn3x32i.dll")
;Assuming user is attached to and logged into server
;
 a=n3ServerInfo("\\CHICAGO",7)
 Message("The server will support at most:","%a% volumes")
```

See Also:

n3GetUser,n3DrivePath, n3DriveStatus, n3ServerList, Environment *(found in main WIL documentation)*

n3ServerList(request)

Returns name of connected server

Syntax:

n3ServerList(request #)

Parameters:

(i) request #	0 - all connected servers (tab-delimited list) 1 -default server 2 - primary server 4 - all known servers (TAB-delimited list)

Returns:

The name of the connected server.

Example:

```
AddExtender("wwn3x32i.dll")
server = n3ServerList(1)
Message("Default server", server)
```

See Also:

n3GetUser,n3DrivePath, n3DriveStatus, n3ServerInfo, Environment *(found in main WIL documentation)*

n3UserGroups(server-name, user-name)

Returns a list of groups to which the specified user belongs.

Syntax:

n3UserGroups(server-name, user-name)

Parameters:

(s) server-name	name of a network file server or empty string.
(i) user-name	name of the current user.

Returns:

(s)	a tab-delimited list of groups.

Example:

```
AddExtender("wwn3x32i.dll")
;
;Assuming user is attached to and logged into server
;
list=N3UserGroups("\\CHICAGO","JOE")
AskItemList("User belongs to these groups",list,@tab,@sorted,@single)
```

See Also:

n3GetUser,n3DrivePath, n3DriveStatus, n3ServerInfo, n3ServerList, Environment *(found in main WIL documentation)*

n3Version( )

Returns the version of this Extender DLL.

Syntax:

n3Version( )

Parameters:

none

Returns:

(i) the version of number of this extender Dll.

This function is used to check the version number of this Dll in cases where older DLL's exist and alternate processing is desirable. Version numbers of newer versions will be larger than that of older versions.

Example:

```
AddExtender("wwn3x32i.dll")
a=n3Version()
Message("Dll Version",a)
```

Netware 3x Error Appendix

- Troubleshooting

"120: Not attached to specified server"
"121: Unknown Error"
"122: Unknown user name"
"123: Unrecognised function"
"124: Drivers not loaded"
"125: Invalid Connection"
"126: No Servers Found"
"127: Unknown File Server"
"128: Incorrect Password"
"129: Use n4Login for Directory Services login"
"130: Use n4Logout for Directory Services logout"
"131: Illegal local drive letter"
"132: Resource not found"
"133: Volume not found"
"134: Invalid Directory Handle (Internal Error?)"
"135: Invalid Path"
"136: Invalid Drive Number (Internal Error?)"
"137: No such property (Internal Error?)"
"138: No such object (Internal Error?)"
"139: No delete privilege"
"140: User lookup failed"
"141: Bad UNC server name. Use \\SRVNAME "
"142: Searchdrive slot number out of range (0 - 16)"
"143: Localdrive already mapped"
"144: Already logged into server"
"145: Map Attempted. Cannot access net resource"
"210: Out of memory"
"213: Invalid request number"
"214: Connect list overflow"
"215: Invalid file name"
"216: Invalid file attributes"
"217: Invalid change mode"
"218: Specified NetWare path is invalid or inaccessible"
"219: Specified file is invalid or inaccessible"
"220: Error changing file attributes"
"221: Wildcards not allowed in filename"

"1000: WIL NetWare 3 Extender"

Introduction

WIL extender Dlls are special Dlls designed to extend the built-in function set of the WIL processor. These Dlls typically add functions not provided in the basic WIL set, such as network commands for particular networks (Novell, Windows for WorkGroups, LAN Manager and others), MAPI, TAPI, and other important Application Program Interface functions as may be defined by the various players in the computer industry from time to time. These Dlls may also include custom built function libraries either by the original authors, or by independent third party developers. (An Extender SDK is available). Custom extender Dlls may add nearly any sort of function to the WIL language, from the mundane network math or database extensions, to items that can control fancy peripherals, including laboratory or manufacturing equipment.

WIL extenders must be installed separately. Up to 10 extender Dlls may be added. The total number of added items may not exceed 100 functions and constants. The **AddExtender** function must be executed before attempting to use any functions in the extender library. The **AddExtender** function should be only executed once in each WIL script that requires it.

INSTALLATION - Using a Dll.

To use a WIL extender, at the top of each script in which you use network commands add the appropriate extender with the AddExtender command.

AddExtender(extender filename)

n4Attach(server-name)

Attaches to and authenticates a user with a Novell 4.x network file server.

Syntax:

n4Attach(server-name)

Parameters:

(s) server-name	name of a network file server.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Attaches to and authenticates a user with a Novell 4.x network file server. User must already be logged in to a Novell Directory Services connection.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
```

See Also:

n4Detach, n4Login

n4CaptureEnd(port-number)

Ends a printer capture.

Syntax:

n4CaptureEnd(port-number)

Parameters:

(i) port-number	number from 1 to 9 (where 1 = "LPT1", etc.)

Returns:

(i)	**@TRUE** if successful.

Example:

```
AddExtender("wwn4x32i.dll")
;Assuming user is attached to and logged into server
;
a=n4CapturePrt("\\CHICAGO","PRINTERC",1,(Tabsize=8%@TAB%Numcopies=2")
RunWait("notepad.exe", "/p readme.txt")
n4CaptureEnd(1)
```

See Also:

n4CapturePrt

n4CaptureGet(port-number)

Returns the name of the print queue associated with a printer capture.

Syntax:

n4CaptureGet(port-number)

Parameters:

(i) port-number	number from 1 to 9 (where 1 = "LPT1", etc.).

Returns:

(i)	an abbreviated name of a print queue relative to the current context, (eg, "Printer1.Print") instead of a canonical name (eg, "Printer1.Print.Sales").

Example:

```
AddExtender("wwn4x32i.dll")
;Assuming user is attached to and logged into server
queue = n4CaptureGet(1)
Message("LPT1 is captured to", queue)
```

See Also:

n4CaptureEnd, n4CapturePrt

n4CapturePrt(server-name, queue-name, port-number, flags)

Captures a local printer port to a Netware printer queue.

Syntax:

n4CapturePrt(server-name, queue-name, port-number, flags)

Parameters:

(s) server-name	name of a network file server.
(s) queue-name	name of a network printer queue.
(s) port-number	number from 1 to 9 (where 1 = "LPT1", etc.).
(s) flags	a tab-delimited list of keyword=value pairs,

Returns:

(i)	**@TRUE** if successful.

Flags denotes a tab-delimited list of keyword=value pairs. For example;

"key1=value1%@TAB%key2=value2%@TAB%key3=value3..."

Valid keywords are:

JobDescription =

Null-terminated ASCII description of the contents or purpose of the job. The NetWare DOS Requester uses only 13 bytes of this member. This member is used only by the DOS Requester and OS/2.

JobControlFlags =

Set of queue job control flags affecting the way a queue server processes a queue job. This member is used only by the DOS Requester and OS/2. Returns 0 under DOS/Windows. Under OS/2 JobControlFlags is defined as 1024 (print interrupted capture), and bits 0, 1, and 2 must be 0.

TabSize =

Returns a value between 1 and 18 inclusive, indicating tab size. The default setting is 8.

NumCopies =

Number of copies of the captured file the printer prints (The maximum number of copies is 255 for netx.com, and 65536 for OS/2 and the DOS Requester). The default setting is 1. If NumCopies is 0, nothing prints.

PrintFlags =

Value	Flag Name	Description
4	Release Job	If set, the print job is released for printing if the capture is interrupted by a loss of connection time tothe server.
8	Suppress Form Feed	If set, the print service suppresses automatic form feed
64	Text File	If set, tab size and other printer control sequences areinterpreted by the print service. If not set, the job is interpreted as a byte stream.

128	Print Banner	If set, the print service precedes the print job with a banner page.

NOTE: If you would like to set more than one of the previous PrintFlag values you can add the values together. For example, if you would like to "Suppress Form Feed" (value=8) and "Print Banner" (value=128) you would add those values together (128+8=136).

```
Example:   PrintFlags=136
```

MaxLines
Maximum lines per page.

MaxChars
Maximum characters per line.

FormName
Name of the form a user must mount in the printer to print files captured to the LPT device. If the form currently mounted in the printer differs from the form name returned in this field, the NetWare server console displays a message instructing the console operator to mount the correct form.

FormType
Type of form (0 to 255) a user must mount in the printer to print files captured to the LPT device. If the form currently mounted in the printer differs from the form type returned in this field, the NetWare server console displays a message instructing the console operator to mount the correct form. The default form is 0.

BannerText
13-byte string containing the name appearing on the bottom half of a banner page. All letters are upper case.

FlushCaptureTimeout
(0 to 3,640) starts counting down every time an application executes a print command (int 17h). When the time-out expires, the server flushes the capture file and queues it at a printer. If an application executes a second print command before the first time-out expires, the time-out starts over from the original value. Each tick of the capture time-out is approximately one second. The range is 1 to 1000. The default time-out is 0---no time-out. (Not valid in OS/2.)

FlushCaptureOnClose
When FlushCaptureOnClose is enabled (0), the server flushes the capture file when the application ends the capture of the default LPT device. (Default is 0.) Any other value means disabled. (Not valid in OS/2.)

Example:

```
AddExtender("wwn4x32i.dll")
;Assuming user is attached to and logged into server
;
a=n4CapturePrt("\\CHICAGO","PRINTERC",1,"Tabsize=8%@TAB%Numcopies=2")
RunWait("notepad.exe", "/p readme.txt")
n4CaptureEnd(1)
```

See Also:

n4CaptureEnd

n4ChgPassword(server-name, user-name, old-password, new-password)

Changes a user's password.

Syntax:

n4ChgPassword(server-name, user-name, old password, new password)

Parameters:

(s) server-name	name of a network file server or empty string.
(s) user-name	name of the current user.
(s) old password	the old password.
(s) new password	a new password.

Returns:

(i)	**@TRUE** if successful.

When changing passwords the "Old-password" must be specified for Directory Services connections.

For bindery connections, if you have supervisor or equivalent rights, a blank string ("") can be specified for "old-password".

Passwords ARE case-sensitive for Directory Services connections, but NOT for bindery connections.

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming user is attached to and logged into server
;
 N4ChgPassword("\\CHICAGO", "joe", "spam", "spasm")
```

n4Detach(server-name)

Detaches from one or all NetWare file servers.

Syntax:

n4Detach(server-name)

Parameters:

(s) server-name	name of a network file server.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

This function will detach a user from a Novell 3 or 4 file server. If a file server name is specified, then the user will be detached from that particular file server. If the server name is an empty string ("") then the user will be logged out of all NetWare file servers.

The user will not be logged out of Directory Services and may attach to other file servers.

Directory Services Monitored Connectons will not be detached. Detaches for Monitored Connections are ignored. The function will return @TRUE

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
```

See Also:

n4Attach, n4Logout

n4DirAttrGet(dirname)

Returns NetWare directory attributes

Syntax:

n4DirAttrGet(dirname)

Parameters:

(s) dirname	directory pathname whose attributes are to be determined.

Returns:

(i)	the attributes of the specified directory pathname.

Returns a number which is the sum of all attributes set for the specified directory. Use the bitwise AND operator (&) to determine if a specific attribute is set. See below for a list of attribute constants.

Attribute Constants :

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:

```
AddExtender("wwn4x32i.dll")
dirname = "n:\public"
attrs = n4DirAttrGet(dirname)
If attrs & @attr_H
  Message(dirname, "is hidden")
Else
  Message(dirname, "is not hidden")
Endif
```

See Also:

n4DirAttrSet, n4FileAttrSet, n4FileAttrGet

n4DirAttrSet(dirname, attribs, mode)

Sets NetWare directory attributes

Syntax:

n4DirAttrSet(dirname, attribs, mode)

Parameters:

(s) dirname	directory pathname.
(s) attribs	one or more NetWare directory attribute constants (see below for list).
(s) mode	**@ON**, specified attributes are set. **@OFF**, specified attributes are removed.

Note: The 'execute-only' attribute cannot be removed.

If multiple attributes are specified, they should be combined using the bitwise OR operator.

Attribute Constants :

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:

```
AddExtender("wwn4x32i.dll")
dirname = "n:\public"
; set 'hidden' and 'system' attributes
n4DirAttrSet(dirname, @attr_H | @attr_Sy, @ON)
; remove 'shareable' attribute
n4DirAttrSet(dirname, @attr_Sh, @OFF)
```

See Also:

n4DirAttrGet, n4FileAttrSet, n4FileAttrGet

n4DirTimeGet(dirname, time-field)

Gets Netware time information for a directory.

Syntax:

n4DirTimeGet(dirname, time-field)

Parameters:

(s) dirname	must specify a single directory name (no wildcards).
(i) time-field	can be one of the following: 1 directory created 2 directory last modified 3 --- (not used) 4 directory last archived

Returns:

(s)	the requested directory time in YmdHms format (with a 4-digit year), or a blank string ("") if the requested time field is not set. (yyy:mm:dd:hh:mm:ss…)

Example:

See Also:

n4DirAttrGet, n4FileTimeGet

n4DrivePath(local-name)

Returns the network resource associated with the local-name.

Syntax:

n4DrivePath(local-name)

Parameters:

(s) local-name	local drive name.

Returns:

(i)	network resource associated with the local name.

This function will return a UNC-style network path that a local drive is mapped to. If there is no valid NetWare mapping, then an empty string will be returned.

Example:

```
AddExtender("wwn4x32i.dll")
netpath = n4DrivePath("Q:")
Message("Q: is mapped to", netpath)
```

See Also:

n4DriveStatus, n4ServerList, n4GetMapped

n4DriveStatus(local-name)

Returns a status code number indicating the type of connection associated with a local-name.

Syntax:

n4DriveStatus(local-name)

Parameters:

(s) local-name	local drive name.

Returns:

(i)	a status code number.

This function returns information about a local drive. It can determine if the drive is unmapped, already mapped, or belongs to another network.

Bit Value	Bit Definition
0	Unmapped, free drive
1	Local Free Drive
2	Local Drive
4	Network Drive
8	PNW Drive
16	Netware Drive

Common status codes are:

0	Free Drive - Mapable
3	Free Drive - Mapable
6	Mapped local drive belonging to a Non-Novell network
23	Mapped drive belonging to Novell

Example:

```
AddExtender("wwn4x32i.dll")
for d = 0 to 25
   drive = strcat( num2char( char2num("A") + d ), ":" )
   stat = n4DriveStatus(drive)
   if stat == 0 || stat == 3
      Message(drive, "is a free, mapable drive")
   endif
   if stat == 23
      path = n4DrivePath(drive)
      Message(drive, "is a mapped Novell drive, mapped to %@CRLF%path")
   endif
next
```

See Also:

n4DrivePath, n4ServerList, n4GetMapped

n4FileAttrGet(filename)

Returns NetWare file attributes

Syntax:

n4FileAttrGet(filename)

Parameters:

(s) filename	a file name, which may include a full path, and which may *not* include wildcards.

Returns:

(i)	sum of all attributes set.

Returns a number which is the sum of all attributes set for the specified file. Use the bitwise AND operator (&) to determine if a specific attribute is set. See below for a list of attribute constants.

Attribute Constants:

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:

```
AddExtender("wwn4x32i.dll")
filename = "n:\public\nwadmin.exe"
attrs = n4FileAttrGet(filename)
If attrs & @attr_H
  Message(filename, "is hidden")
Else
  Message(filename, "is not hidden")
Endif
```

See Also:

n4FileAttrSet

n4FileAttrSet(filename, attribs, mode)

Sets NetWare file attributes

Syntax:

n4FileAttrSet(filename, attribs, mode)

Parameters:

(s) filename	a file name, which may include a full path, and which may include wildcards.
(i) attribs	one or more NetWare file attribute constants (see below for list).
(i) mode	**@ON**, specified attributes are set. **@OFF**, specified attributes are removed.

If multiple attributes are specified, they should be combined using the bitwise OR operator.

Note: The 'execute-only' attribute cannot be removed.

Attribute Constants:

@attr_Ro	Read-only
@attr_H	Hidden
@attr_Sy	System
@attr_X	eXecute-only
@attr_A	Archive-needed
@attr_Sh	Shareable
@attr_T	Transactional
@attr_P	Purge
@attr_Ri	Rename-inhibit
@attr_Di	Delete-inhibit
@attr_Ci	Copy-inhibit
@attr_Dm	Don't migrate
@attr_Ic	Immediate compress
@attr_Dc	Don't compress

Example:

```
AddExtender("wwn4x32i.dll")
filename = "n:\public\nwadmin.exe"
; set 'hidden' and 'system' attributes
n4FileAttrSet(filename, @attr_H | @attr_Sy, @ON)
; remove 'shareable' attribute
n4FileAttrSet(filename, @attr_Sh, @OFF)
```

SeeAlso:

n4FileAttrGet

n4FileTimeGet(filename, time-field)

Gets Netware time information for a file.

Syntax:

n4FileTimeGet(filename, time-field)

Parameters:

(s) filename	must specify a single file name (no wildcards).
(i) time-field	can be one of the following: 1 file created 2 file last modified 3 file last accessed 4 file last archived

Returns:

(s)	the requested file time in YmdHms format (with a 4-digityear), or a blank string ("") if the requested time field is not set.

Example:

See Also:

n4DirTimeGet,

n4GetContext(request)

Returns the current user's default context or tree.

Syntax:

n4GetContext(request)

Parameters:

(i) request	see below.

Returns:

(s)	0-request returns context or 1-request returns tree.

Request #

0	context
1	tree (32 bit version only)

Example:

```
addExtender("wwn4x32i.dll")
context = n4GetContext(0)
list=N4UserGroupEx("\\CHICAGO","JOE",context)
AskItemList("User belongs to these groups",list,@tab,@sorted,@single)
```

See Also:

n4UserGroupEx

n4GetMapped(server-name)

Returns a tab delimited list of mapped Novell drives.

Syntax:

n4GetMapped(server-name)

Parameters:

(s) server-name	name of a network file server or empty string.

Returns:

(s)	tab delimited list of drives mapped to the specified server

This function will interrogate drives A thru Z, and will return a list of drives mapped to Novell servers. If there is no valid NetWare mapping, then an empty string will be returned. If an empty string ("") is provided for the server name, then all valid mapped Novell drives will be returned.

Example:

```
addExtender("wwn4x32i.dll")
MappedDrives = n4GetMapped("\\SERV01")
Message("Mapped drives on \\SERV01", Mapped Drives)
```

See Also:

n4DrivePath, n4DriveStatus

n4GetNetAddr(server-name, flags)

Returns the Internet work address of the current workstation.

Syntax:

n4GetNetAddr(server-name, flags)

Parameters:

(s) server-name	name of a network file server or empty string.
(i) flags	0 - reserved for future use.

Returns:

(i)	string of hex bytes in the format: x1x2x3x4:y1y2y3y4y5y6:z1z2

If a null string ("") is used for the server-name, the default server will be used.

The Internet work address of the current workstation is returned in the form of

x1x2x3x4:y1y2y3y4y5y6:z1z2

where:

x1x2x3x4	= 4-byte network address
y1y2y3y4y5y6	= 6-byte net node address (physical address of workstation's LAN board)
z1z2	= socket

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming user is attached to and logged into server
;
 a=n4GetNetAddr("\\CHICAGO",0)
 message("Net Address is","%a%")
```

See Also:

n4GetUser, n4DrivePath, n4DriveStatus, n4ServerList, Environment *(found in main WIL documentation)*

n4GetUser(server-name)

Determines the currently logged on user name on the specified server.

Syntax:

n4GetUser(server-name)

Parameters:

(s) server-name	name of a network file server.

Returns:

(i)	a user name.

This function will return the currently logged on user name on a specified Novell server. If no user is logged on, a null string ("") will be returned.

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming user is attached to and logged into server
;
Who = n4GetUser("\\DEPT07")
Message("I am logged into \\DEPT07 as", Who)
;
; Also Note this may work, depending on your setup
EnvWho = Environment("USER")
Message("Environment variable USER is",EnvWho)
```

See Also:

n4GetNetAddr, n4DrivePath, n4DriveStatus, n4ServerList, Environment *(found in main WIL documentation)*

n4GetUserId(server-name, user-name, format)

Returns the object ID corresponding to the specified user name.

Syntax:

n4GetUserId(server-name, user-name, format)

Parameters:

(s) server-name	name of a network file server or empty string.
(s) user-name	name of the current user.
(i) format	the format for the return; see below.

Returns:

(i)	object ID for specified user.

"Format" specifies the format in which the ID is returned:

Request #	Format
0	Decimal number, in internal Netware format
1	Hexadecimal string, such as used for mail directories

Example:

```
AddExtender("wwn4x32i.dll")

;Assuming user is attached to and logged into server

userid=N4GetUserId("\\CHICAGO","joe",0)
message("UserID is","%userid%")
```

See Also:

n4GetUser, n4DrivePath, n4DriveStatus, n4ServerList, Environment *(found in main WIL documentation)*

n4Login(user-name, password, context, tree)

Performs a login to Novell Directory Services.

Syntax:

n4Login(user-name, password, context, tree)

Parameters:

(s) user-name	name of the current user.
(s) password	password required to access Directory Services.
(s) context	desired login context.
(s) tree	desired Directory Services Tree.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Performs a login to Novell Directory Services. For Bindery based logins, use the Novell 3.x extender. This function does NOT run any login scripts, nor will it detach the user fron any existing servers.

If the user has a valid settings in the NET.CFG network configuration file, including the following:

```
PREFERRED SERVER   dir-svc-server
PREFERRED TREE     treename
CONTEXT NAME       contextname
```

and the defaults are to be used, then the **context** and **tree** parameters may be empty strings (""). Otherwise the context and tree parameters must be specified to login.

Example:

```
;Normal everyday server access script w/ login and logout
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Context="ACCOUNTING"
Tree="WORLDWIDEINC"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n4Login(UserID, Pswd, Context, Tree)
OrigDir=DirGet()
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
n4Logout()
```

See Also:

n4Logout, n4Attach, n4Detach

n4Logout( )

Performs a network logout.

Syntax:

n4Logout( )

Parameters:

None

Returns:

(i) **@TRUE** if successful;
@FALSE if unsuccessful.

The Logout function will log the user out of Directory Services and detach the user from all unnecessary fileservers - Both Directory Services 4.x servers and Bindery 3.x servers will be detached. The Monitored Connection to the Directory Services server will be retained.

Example:

```
;Normal everyday server access script w/ login and logout
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n4Login(UserID, Pswd, "", "")
OrigDir=DirGet()
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
n4Logout()
```

See Also:

n4Login, n4Detach

n4LogoutTree(context, tree)

Performs a network logout from the specified tree.

Syntax:

n4LogoutTree (context, tree)

Parameters:

(s) context	desired login context.
(s) tree	desired Directory Services Tree.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

The Logout function will log the user out of Directory Services "context" is a Directory Services context, or "" for the default context.
"tree" is a Directory Services tree, or "" for the default tree.
Note: the "tree" parameter is only supported in the 32 bit version.

This function terminates a client's connection to the network. Unlike the n4Logout function, it does not detach from any servers.

Returns @TRUE on success, or @FALSE if the user was not authenticated through Directory Services.

Example:

```
;Normal everyday server access script w/ login and logout
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
n4Login(UserID, Pswd, "", "")
OrigDir=DirGet()
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
n4Logout()
```

See Also:

n4Login, n4Detach, n4Logout

n4Map(net-path, local-name)

Root maps a drive to a resource specified by a UNC pathname.

Syntax:

n4Map(net-path, local-name)

Parameters:

(s) net-path	UNC net resource.
(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Root maps a drive to a resource specified by a fully qualified UNC filename. Must be either logged into the server, or specified net resource must not require a login.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
OrigDir=DirGet()
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
```

See Also:

n4DrivePath, n4DriveStatus, n4MapDelete, n4MapDir

n4MapDelete(local-name)

Removes a drive mapping.

Syntax:

n4MapDelete(local-name)

Parameters:

(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

This function removes a drive mapping.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
OrigDir=DirGet()
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
```

See Also:

n4Map, n4DrivePath, n4DriveStatus

n4MapDir(net-path, local-name)

Maps a drive to a resource specified by a UNC pathname.

Syntax:

n4MapDir(net-path, local-name)

Parameters:

(s) net-path	UNC net resource.
(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Maps a drive to a resource specified by a fully qualified UNC filename. Must be either logged into the server, or specified net resource must not require a login. If drive is already mapped to a Novell server, the prior mapping will be deleted and the new mapping will take effect.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
OrigDir=DirGet()
n4Attach(MyServer, UserID, Pswd)
n4MapDir("\\DEPT07\SYS\Excel", "Q:")
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
```

See Also:

n4MapDelete, n4MapRoot, n4Map, n4DrivePath, n4DriveStatus

n4MapRoot(net-path, local-name)

Root maps a drive to a resource specified by a UNC pathname.

Syntax:

n4MapRoot(net-path, local-name)

Parameters:

(s) net-path	UNC net resource.
(s) local-name	local drive name.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Root maps a drive to a resource specified by a fully qualified UNC filename. Must be either logged into the server, or specified net resource must not require a login. If drive is already mapped to a Novell server, the prior mapping will be deleted and the new mapping will take effect.

Example:

```
;Normal everyday server access script
;See example with "LastError()" in this help file
; for a more bullet-proof version
;
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Pswd=AskPassword("Login to Server %MyServer%", "Enter Password for %UserID%")
OrigDir=DirGet()
n4Attach(MyServer, UserID, Pswd)
n4MapRoot("\\DEPT07\SYS\Excel", "Q:")
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
```

See Also:

n4MapDelete, n4MapDir, n4Map, n4DrivePath, n4DriveStatus

n4MemberDel(group-name ,user-name)

Deletes the specified user from the specified group on the specified server.

Syntax:

n4MemberDel(group-name ,user-name)

Parameters:

(s) group-name	name of the group.
(s) user-name	name of the current user.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Assuming that the person running this script has sufficient authority to delete users from the specified group, this function will delete the specified user from the group.

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming operator is attached to and logged into server
;
n4MemberDel("MAUI SALES", "BSMITH")
```

See Also:

n4MemberGet, n4MemberSet

n4MemberGet(group-name ,user-name)

Determines if the specified user is a member of the specified group on the specified server.

Syntax:

n4MemberGet(group-name ,user-name)

Parameters:

(s) group-name	name of the group.
(s) user-name	name of the current user.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Assuming that the person running this script has sufficient authority to query members of the specified group, this function will allow the person to determine if the user is a member of the specified group or not.

n4MemberGet can accept distinguished names.

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming operator is attached to and logged into server
;
resp = n4MemberGet("NOME SALES", "BSMITH")
if resp == 0 then resp2 = "is NOT"
else resp2 = "is"
Message("NOME SALES","BSMITH %resp2% a member.")
```

See Also:

n4MemberSet, n4MemberDel

n4MemberSet(group-name , user-name)

Sets the specified user as a member of the specified group on the specified server.

Syntax:

n4MemberSet(group-name , user-name)

Parameters:

(s) group-name	name of the group.
(s) user-name	name of the current user.

Assuming that the person running this script has sufficient authority to add users to the specified group, this function will add the specified user to the group.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming operator is attached to and logged into server
;
n4MemberSet("NOME SALES", "BSMITH")
```

See Also:

n4MemberDel, n4MemberGet

n4MsgSend(server-name, message , user-name)

Sends a message (max 56 characters) to the specified user.

Syntax:

n4MsgSend(server-name, message , user-name)

Parameters:

(s) server-name	name of a network file server.
(s) message	message to be sent.
(s) user-name	name of the user to whom the message is being sent.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Example:

```
AddExtender("wwn4x32i.dll")
Msg = AskLine("My Mailing List","Enter Short Message","")
n4MsgSend("\\DEPT07", Msg, "ERINP")
n4MsgSend("\\DEPT07", Msg, "KARINW")
n4MsgSend("\\DEPT07", Msg, "TINAB")
n4MsgSend("\\DEPT07", Msg, "CAROLK")
n4MsgSend("\\DEPT07", Msg, "LAURAW")
n4MsgSend("\\DEPT07", Msg, "DONNAW")
n4MsgSend("\\DEPT07", Msg, "CUPCAKE")
```

See Also:

n4MsgSendAll

n4MsgSendAll(server-name, message)

Sends a message (max 56 characters) to all logged on users.

Syntax:

n4MsgSendAll(server-name, message)

Parameters:

(s) server-name	name of a network file server.
(s) message	message to be sent.

Returns:

(i)	**@TRUE** if successful; **@FALSE** if unsuccessful.

Server-wide broadcasting to all logged in users. Actually only gets the first 300 or so.

Example:

```
AddExtender("wwn4x32i.dll")
n4MsgSendAll("\\DEPT07","Blue Chevy, License 237-EKL, Lights on in back lot")
```

See Also:

n4MsgSend

n4NameConvert(context, object, format)

Converts an object name to a different format.

Syntax:

n4NameConvert(context, object, format)

Parameters:

(s) context	a NetWare context, or "" for the current context.
(s) object	the name of a NetWare object.
(i) format	specifies what format "object" will be converted to. (see below)

Returns:

(i)	the name of the new format.

"format" specifies what format "object" will be converted to, and is one of the following:

1	canonical	typed	(eg, "CN=Server.O=Office")
2	abbreviated	typed	(eg, "CN=Server")
3	canonical	typeless	(eg, "Server.Office")
4	abbreviated	typeless	(eg, "Server")
5	UNC		(eg, "\\Server")

If "object" contains a period ("."), it is assumed to be in canonical format; otherwise, it is assumed to be in abbreviated format. If "object" begins with two slashes "\\", it is assumed to be a UNC. If "object" is in an invalid format, it will be returned as-is, without being converted.

Example:

```
objname = n4NameConvert("", "\\Server", 3)
Message("Object name", objname)
```

See Also:

n4ObjectList, n4ObjectInfo, n4ObjectProps

n4ObjectList(context, parent, class, mask)

Lists objects in a Directory Services tree.

Syntax:

n4ObjectList(context, parent, class, mask)

Parameters:

(s) context	A NetWare context, or "" for the current context.
(s) parent	The object whose immediate sub-ordinates are to be listed, or "" for the top of the tree.
(s) class	A NetWare class which identifies the type of objects to be listed or "" for all objects. (see below)
(s) mask	A wildcard which identifies the objects to be listed such as "Serv*" or "" for all names.

Returns:

(i)	Returns a tab-delimited list of object names.

"Class" can be one of the following NetWare classes or a null string, "", for all objects.

"AFP Server"	"Organization"
"Alias"	"Organizational Person"
"Bindery Object"	"Organizational Role"
"Bindery Queue"	"Organizational Unit"
"Computer"	"Partition"
"Country"	"Person"
"Device"	"Print Server"
"Directory Map"	"Printer"
"External Entity"	"Profile"
"Group"	"Queue"
"List"	"Resource"
"Locality"	"Server"
"Message Routing Group"	"Top"
"Messaging Routing Group"	"Unknown"
"Messaging Server"	"User"
"NCP Server"	"Volume"

Note: Standard NetWare servers have a class of "NCP Server".

Example:

```
objects = n4ObjectList("", "", "Queue", "")
askitemlist("Queues in current context", objects, @TAB, @sorted,@single)

;Get a list of groups on a Novell Netware 4 server
grpobjects = n4ObjectList("", "Group", "", "")
askitemlist( "List of groups",grpobjects, @tab, @sorted, @single)
```

See Also:

n4ObjectInfo, n4ObjectProps, n4NameConvert

n4ObjectInfo(context, object, request)

Returns information on the specified object.

Syntax:

n4ObjectInfo(context, object, request#)

Parameters:

(s) context	a NetWare context, or "" for the current context.
(s) object	the name of a NetWare object.
(i) request#	**0** - Server where the object is stored (eg, "\\SERVER"). This is not necessarily the server on which the object is physically located. **1** - Object class (see "n4ObjectList" for a list of classes).

Returns:

(i)	information on the specified object.

Example:

```
server = n4ObjectInfo("Admin", "Print01", 0)
Message("Print01 is located on", server)
```

See Also:

n4ObjectList, n4ObjectProps, n4NameConvert

n4ObjectProps(context, object, attribute)

Returns properties of the specified object.

Syntax:

n4ObjectInfo(context, object, attribute)

Parameters:

(s) context	a NetWare context, or "" for the current context.
(s) object	the name of a NetWare object.
(s) attribute	a valid attribute of "object" or "" for a tab-delimited list of all attributes.

Returns:

(i)	a tab-delimited list of values for an attribute.

If "attribute" is a blank string (""), this function will return a tab-delimited list of all attributes for "object".

If "attribute" specifies a valid attribute of "object", this function will return a tab-delimited list of values for that attribute.

Note: Only string values will be returned correctly.

Examples:

```
host = n4ObjectProps("", "Print01", "Host Device")
Message("Host Device", host)

; list all attributes and values for an object
attribs = n4ObjectProps("", "Print01", "")
count = ItemCount(attribs, @TAB)
For i = 1 To count
  attrib = ItemExtract(i, attribs, @TAB)
  values = n4ObjectProps("", "Print01", attrib)
  ItemSelect(attrib, values, @TAB)
Next
```

See Also:

n4ObjectList, n4ObjectInfo, n4NameConvert

n4ServerInfo(server-name, request #)

Returns information on the specified Netware server.

Syntax:

n4ServerInfo(server-name, request #)

Parameters:

(s) server-name	name of a network file server or empty string.
(i) request #	see below.

Returns:

(i)	the requested server information.

Request	**Meaning**
1	major NetWare version number
2	minor NetWare version number
3	revision number of the NetWare OS on NetWare server
4	maximum number of connections the server will support
5	highest number of connections simultaneously in use
6	number of connections the server currently has in use
7	maximum number of volumes the server will support
8	SFT level the server supports
9	TTS Level of NetWare server operating system

Example:

```
AddExtender("wwn4x32i.dll")
;Assuming user is attached to and logged into server
;
 a=n4ServerInfo("\\CHICAGO",7)
 Message("The server will support at most:","%a% volumes")
```

See Also:

n4GetUser, n4DrivePath, n4DriveStatus, n4ServerList, Environment *(found in main WIL documentation)*

n4ServerList(request#)

Returns name of connected server.

Syntax:

n4ServerList(request #)

Parameters:

(i) request #	0 - all connected servers (tab-delimited list) 1 - default server 2 - primary server 3 - preferred server or tree 4 - all known servers (TAB-delimited list)

Returns:

(s)	the name of the connected server.

Example:

```
AddExtender("wwn4x32i.dll")
server = n4ServerList(1)
Message("Default server", server)
```

See Also:

n4GetUser, n4DrivePath, n4DriveStatus, Environment *(found in main WIL documentation)*

n4SetContext(context, tree)

Changes the current user's default context and/or tree.

Syntax:

n4SetContext(context, tree)

Parameters:

(s) context	Directory Services context to be set. If this parameter is a blank string (""), the context will not be changed.
(s) tree	Directory Services tree to be set. If this parameter is a blank string (""), the tree will not be changed.

Returns:

(i)	always 1.

Note: the "tree" parameter is available only in the 32-bit version, and is ignored in the 16-bit version.

This function changes the NetWare context and/or tree that is used by subsequent function calls from this extender (for those functions which do not take an explict "context" or "tree" parameter).

Example:

```
AddExtender("wwn4x32i.dll")
MyServer="\\DEPT07"
UserID="FRED"
Context="ACCOUNTING"
Tree="WORLDWIDEINC"
n4SetContext(context,tree)
n4Login(UserID, Pswd, Context, Tree)
n4Attach(MyServer)
n4Map("\\DEPT07\SYS\Excel", "Q:")
OrigDir=DirGet()
DirChange("Q:\")
RunWait("EXCEL.EXE","/E")
DirChange(OrigDir)
n4MapDelete("Q:")
if n4GetMapped(MyServer)=="" then n4Detach(MyServer)
n4Logout()
```

See Also:

n4login

n4UserGroupEx(server-name, user-name, context)

Returns a list of groups to which the specified user belongs, in the specified context.

Syntax:

n4UserGroupEx(server-name, user-name, context)

Parameters:

(s) servername	name of a network file server.
(s) user-name	name of the current user.
(s) context	a Directory Services context, or "" for the default context.

Returns:

(s)	a tab-delimited list of groups.

If the specified server is a Directory Services connection, the group names will contain distinguishing path information. (i.e. "Server1.Users")

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming user is attached to and logged into server
;
list=N4UserGroupEx("\\CHICAGO","JOE","")
AskItemList("User belongs to these groups",list,@tab,@sorted,@single)
```

See Also:

n4GetUser, n4DrivePath, n4DriveStatus, n4ServerList, Environment *(found in main WIL documentation)*

n4UserGroups(server-name, user-name)

Returns a list of groups to which the specified user belongs.

Syntax:

n4UserGroups(server-name, user-name)

Parameters:

(s) servername	name of a network file server.
(s) user-name	name of the current user.

Returns:

(s)	a tab-delimited list of groups.

If the specified server is a Directory Services connection, the group names will contain distinguishing path information. (i.e. "Server1.Users")

Example:

```
AddExtender("wwn4x32i.dll")
;
;Assuming user is attached to and logged into server
;
list=N4UserGroups("\\CHICAGO","JOE")
AskItemList("User belongs to these groups",list,@tab,@sorted,@single)
```

See Also:

n4GetUser, n4DrivePath, n4DriveStatus, n4ServerList, Environment *(found in main WIL documentation)*

n4Version( )

Returns the version of this Extender DLL.

Syntax:

n4Version( )

Parameters:

none

Returns:

(i) the version of number of this extender Dll.

This function is used to check the version number of this Dll in cases where older DLL's exist and alternate processing is desirable. Version numbers of newer versions will be larger than that of older versions.

Example:

```
AddExtender("wwn4x32i.dll")
a=n4Version()
Message("Dll Version",a)
```

Netware 4x Error Appendix

- Troubleshooting

"120: Not attached to specified server"
"121: Unknown Error"
"122: Bad user name"
"123: Unrecognized function"
"124: Drivers not loaded"
"125: Invalid Connection"
"126: No Server Error"
"127: Unknown File Server"
"128: Bad Password"
"129: Use n4Login for Directory Services login"
"130: Use n4Logout for Directory Services logout"
"131: Bad local drive letter"
"132: Resource not recognised"
"133: Volume not recognised"
"134: Bad Directory Handle (Internal Error?)"
"135: Invalid Path"
"136: Invalid Drive Number (Internal Error?)"
"137: No such property (Internal Error?)"
"138: No such object (Internal Error?)"
"139: No object delete privilege"
"140: User lookup failed"
"141: Bad UNC server name. Use \\SRVNAME"
"142: Searchdrive slot number out of range"
"143: Localdrive already mapped"
"144: Already logged into server"
"145: Map Attempted. Cannot access net resource"
"200: Directory Services Login Failed"
"201: Directory Services Logout Failed"
"202: Authentication on attach failed"
"210: Out of memory"
"211: Invalid group name"
"212: Insufficient security"
"213: Invalid request number"
"214: Connect list overflow"
"215: Invalid file name"
"216: Invalid file attributes"
"217: Invalid change mode"
"218: Specified NetWare path is invalid or inaccessible"
"219: Specified file is invalid or inaccessible"
"220: Error changing file attributes"
"221: Wildcards not allowed in filename"
"222: Invalid user name or context"
"223: Password has expired"
"1000: WIL NetWare 4 Extender"

WIL Netware Extender Help File 

Troubleshooting

If you're new to using the WIL Network Extenders, you may encounter a few easily fixed problems early on. Here are some hints and tips for getting up and running smoothly. These are also the most common mistakes. So, If you've already received an error, definitely go through the list.

- Technical Support

1. WIL commands use the Unified Naming Convention. In the server name replace all forward slashes (/) and colons (:) with backslashes (\ ).

2. The extenders must be where the WIL interpreter can find them. Are they in the same directory as the WIL script?

3. Additional dll's may be needed for full functionality. Do a search for "nwcalls.dll", a netware dll. If this dll is missing or corrupt, the networking commands will fail. Nwcalls.dll must be dated at least 2-28-94 with a size of at least 146KB. Make sure there is a copy in the Windows directory or in the current directory with the WIL script.

4. Above all, your system needs to be configured so that attachments can be made from within Windows, (not from the DOS prompt.) If you cannot connect from the Windows File Manager or the Windows Explorer manually, stop everything and configure your system to do so.