For 16-bit Intel processors

   AddExtender("wwn3x16i.dll")
            Other required DLL's: nwcalls.dll

For 32-bit processors

   AddExtender("wwn3z32i.dll")
            Other required DLL's: nwcalls.dll, wwn3z16i.dll

Other required DLL's: NWCALLS.DLL

This extender provides standard support for Novell 3.x networks.
It may be used in conjunction with other extenders, such as the
Windows for Workgroups Multinet extender, and the Novell 4.x extender.

Table of Contents

FUNCTIONS

n3Map(net-path, local-name)
n3MapDelete(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)
n3ServerList(request)
n3Version( )

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)

Remember you can add up to 10 extender Dlls or a combined total of 100 functions.

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 Network Extender.

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.
s̲ystem
        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.

This network extender developed by Morrie Wilson.

Documentation written by Tina Browning.

Wilson WindowWare, Inc.
2701 California Ave SW   ste 212
Seattle, WA 98116

Orders: **(800) 762-8383**
Support: **(206) 937-9335**
Fax: **(206) 935-7129**

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

AddExtender(extender filename)

Remember you can add up to 10 extender Dlls or a combined total of 100 functions.

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 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.

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)*

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("wwn3x16i.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)*

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)*

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)

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)*

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 "".

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("wwn3x16i.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

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("wwn3x16i.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

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.

Note: This function uses Novell VLM's.   If this function does not work for you and you are using older Novell software try the n3DrivePath2.

Example:
```
AddExtender("wwn3x16i.dll")
netpath = n3DrivePath("Q:")
Message("Q: is mapped to", netpath)
```

See Also:
n3DriveStatus, n3GetMapped
n3DrivePath2, n3ServerList

Returns the name of a connected network resource.

Syntax:
n3DrivePath2(local-name)

Parameters:
(s) local-name   local drive name.

Returns:
(s)   name of a network resource.

The **n3DrivePath2** function returns the name of the shared resource associated with the specified redirected local drive or device.

Example:
```
AddExtender("www3a16i.dll")
;Checking what net resource a local drive is connected to
netpath = N3DrivePath2("Q:")
Message("Local drive Q: is connected to",netpath)
```

See Also:
n3Map, n3MapDelete
n3DrivePath

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("wwn3x16i.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

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("wwn3x16i.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

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
(s) attribs	one or more NetWare file 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("wwn3x16i.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

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("wwn3x16i.dll")
MappedDrives = n3GetMapped("\\SERV01")
Message("Mapped drives on \\SERV01", MappedDrives)
```

See Also:
    n3DriveStatus, n3DrivePath

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("wwn3x16i.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, n3ServerList, Environment *(found in main WIL documentation)*

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("wwn3x16i.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

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.

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("wwn3x16i.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, n3DrivePath, n3DriveStatus

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("wwn3x16i.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, n3DriveStatus, n3DrivePath

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("wwn3x16i.dll")
;
;Assuming operator is attached to and logged into server
;
n3MemberDel("\\DEPT07", "MAUI SALES", "BSMITH")
```

See Also:
   n3MemberGet, n3MemberSet

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("wwn3x16i.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

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("wwn3x16i.dll")
;
;Assuming operator is attached to and logged into server
;
n3MemberSet("\\DEPT07", "NOME SALES", "BSMITH")
```

See Also:
n3MemberDel, n3MemberGet

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("wwn3x16i.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

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("wwn3x16i.dll")
n3MsgSendAll("\\DEPT07","Blue Chevy, License 237-EKL, Lights on in back lot")
```

See Also:
n3MsgSend

Returns name of connected server

Syntax:
n3ServerList(request #)

Parameters:
(i) request #   0 - all connected servers (tab-delimited list)

                     1 -default server

                     2 - primary server

Returns:
The name of the connected server.

Example:
```
AddExtender("wwn3x16i.dll")
server = n3ServerList(1)
Message("Default server", server)
```

See Also:
n3GetUser, n3DrivePath, n3DriveStatus, Environment *(found in main WIL documentation)*

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("wwn3x16i.dll")
a=n3Version()
Message("Dll Version",a)
```

"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"