

NWMENU - an Application Menu for Windows on Netware

A Netware aware program for allowing users to start different programs depending on their membership of Netware security groups, and to allow automatic attaching to file servers and mapping of network drives.

Version 2.20
April 1995
Copyright © 1994–1995
Nicholas J Payne
All rights reserved

DISCLAIMER OF WARRANTY

THIS SOFTWARE AND DOCUMENTATION ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH THIS PROGRAM MAY BE PUT, NO WARRANTY OF FITNESS FOR PARTICULAR PURPOSE IS OFFERED. GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

Netware is a registered trademark of Novell, Inc
Windows is a trademark of Microsoft Corporation

Introduction

NWMenu gives you the capability, from the one icon, to show different users different applications that they can start depending on the Netware security groups of which they are a member.

It also has the capability, if you are using applications on other servers or CD-ROMs mounted as Netware volumes, to attach the user to the server; map the volume to a drive letter; and then start the application. When the application is closed the user can be automatically disconnected from the server, to minimize the number of connections in use.

If your application server or CD server has limited connections, you have two methods of automatically disconnecting a user from the server once they have finished using an application. You can either set up NWMenu to wait until the application terminates and then drop the connection if there are no other drives mapped to that server, or run a second Windows program (DISCONN.EXE), which runs continuously in the background and periodically tests whether the user still has files open on the server. When the number of open files on the server falls to zero, the user's connection to that server is terminated.

If a user is logged in as a Netware 4.x NDS user, NWMenu makes NDS calls to obtain information about the user and their groups and to attach to other servers. For bindery users NWMenu makes bindery calls. You do not need to run bindery emulation on the client to support NWMenu if you have a network which has no Netware 3.x or 2.x servers.

Prompts, menu text, and dialog boxes can appear in the language of your choice.

Why use NWMenu

One of the problems with trying to maintain a networked Windows configuration in an organisation of any size is that, no matter how much you try to standardize, different users need access to different applications. Over time, this becomes a maintenance nightmare when changes are needed. With NWMenu, you can give all users one (or more) NWMenu icons, each driven by a single INI file on a shared network drive, and depending on the Netware groups of which they are a member, different users will see different applications that they can start.

You can also configure an NWMenu icon to automatically start one of the applications listed in the INI file without any user interaction. This includes automatically attaching to a server and

mapping a drive if needed, and, if wanted, automatically detaching from the server when the application terminates.

For example, you may have a range of networked CD-ROM applications, each of which requires that the user have a connection to the CD server and a mapping to the CD volume, and for ease of maintenance you have the icons for these in a central group file that is shared by all users. All users see all icons, regardless of which applications they can actually access. With NWMenu, each user can have a "CD-ROM Applications" icon, and when they open it, each user will see only the applications to which they have access, their connection and drive mapping is automatically established for them when they choose an application, and their connection is automatically terminated when they close the application:

System requirements

Microsoft Windows 3.1 or later with support for Netware installed; 100Kb of disk space needed for NWMenu; NWCALLS.DLL, NWLOCALE.DLL, and NWNET.DLL installed in your WINDOWS\SYSTEM directory (WINDOWS directory for a shared network installation).

Note: Some of the Netware functions that NWMenu calls when attaching to a file server, mapping a volume, and automatically logging out on application termination do not work reliably if you are using old versions of some of the Netware DLLs. In particular, the DLLs in WINUP9.EXE do *not* properly support some of the Netware calls. NWMenu has been tested on Netware 3.11, 3.12, 4.02, and 4.1, using either NETX.EXE 3.32 or VLM.EXE 1.20 on the client, with the following versions of the DLLs. All the Netware functions called by NWMenu come from one of these three DLLs.

NWCALLS.DLL	146,616	10-20-94	11:56a
NWLOCALE.DLL	43,088	09-20-94	12:16p
NWNET.DLL	225,264	10-18-94	4:35p

These files are from the Netware 4.1 DOSWIN client kit, and can be downloaded from Netware (GO NOVFILES). If you are using older files than these, and you experience problems with NWMenu, try updating the DLLs and see if the problem disappears. They should be installed in the WINDOWS\SYSTEM directory, *not* the WINDOWS directory. Some installation programs, such as early versions of the Novell DOS/Windows client install program, incorrectly install these files into the WINDOWS directory. Ensure that you do not have one version of a DLL in one directory and another version in another directory, as this will nearly always cause problems.

Installation

This archive contains:

NWMENU.WRI	this file
NWMENU.EXE	the menuing program
NETCDROM.INI	A sample INI file for using networked CD-ROM drives and applications on other servers
PASSENCR.EXE	a program for taking a password and encrypting it for storage in an INI file (see below)
DISCONN.EXE	a background program for automatically logging a user out of an application server when they finish using it (see below)
DISCONN.INI	A sample INI file for the DISCONN program
CTL3DV2.DLL	Microsoft DLL for 3-D dialog boxes

1. Copy the NWMENU.* files to any directory you choose. A shared application directory on the server disk is the most appropriate location. As NWMenu makes calls to Netware, it will not run on a workstation that is not logged into a Netware file server and that does not have Windows Netware support configured. Copy DISCONN.EXE to the same location. Copy PASSENCR.EXE to a location where it is not available to normal users, as it should only be used by the LAN administrator.
2. Copy CTL3DV2.DLL to your WINDOWS\SYSTEM directory (shared WINDOWS directory for a network installation). If you already have this file and your existing copy is a later version than the file in this archive, omit this step. This file *must* go in the Windows System directory. It will not work if called from anywhere else. You should *not* put a copy of CTL3DV2.DLL in the same directory as NWMenu. If you copy this file into the same directory as NWMenu, when you run NWMenu you will get an error message from CTL3DV2. This is because the copy in the NWMenu directory has been found

before the copy in the WINDOWS\SYSTEM directory.

3. Ensure that all your users have up-to-date versions of NWCALLS.DLL, NWLOCALE.DLL, and NWNEDT.DLL installed in their WINDOWS\SYSTEM directory. If you are using VLM.EXE on the workstation you should already have these files installed. Check that they are the most current version. See the *System Requirements* section above for details on DLL versions.
4. For each menuing icon that you want to set up, you need to create an INI file and an icon on each user's desktop. The INI file format is described in the section *Setting up INI files* below. For each icon you set up on the users desktop, give it the following properties:

Description: Whatever you like
Command Line: <path>\NWMENU.EXE <path>\<name>.INI
Working Directory: Not needed

Important note: Unless the INI file is in the users Windows directory, you *must* specify the full path to it on the command line. If you want all users to share the one INI file, then you will need to specify the path to it, as it won't be in any user's Windows directory.

5. To setup an icon which automatically runs one of the programs in the INI file without user intervention, create the icon with the following properties:

Description: Whatever you like
Command Line: <path>\NWMENU.EXE <path>\<name>.INI <program desc>
Working Directory: Not needed

Where <program desc> is the description of the program in the INI file (explained below), with any ampersands removed and spaces replaced with underscores. For example, if the sample INI file below was named SAMPLE.INI and was in the directory W:\UTILS, then an icon to automatically run the Developer Network CD would have the properties:

Description: Whatever you like
Command Line: w:\utils\nwmenu.exe w:\utils\sample.ini Developer_Network_CD
Working Directory: Not needed

This icon would automatically attach to the CD server, map the CD volume, run the application, and logout from the CD server when the user closed the application.

Setting up INI files

For the shareware version, you have to create the INI files manually. Registered users will receive a separate program which makes it extremely easy to maintain the INI files and ensure that they are in a consistent state and that the application buttons appear in the order you desire. Here is a sample INI file which I will discuss:

```
[NWMenu]
NWMenu=General Utilities
ButtonWidth=25
FirstNetworkDrive=F
AllowMultipleInstances=1
AllowbinderyAndNDS=1
GenericLoginId=NETWARECD
GenericPassword=]V[boeVVTMi`k/VPgWVZ^bXQ

[Applications]
App1=Phone &Book
App2=&Developer Network CD
App3=C&haracter Map
App4=Manage Pass&words
App5=&Virus Scan
App6=Change &Password
App7=&Archive Documents
App8=&Map Drive
App9=Con&trol Panel
App10=Change P&rint Queue
App11=&Send Message
```

```

App12=&Flex Sheet

[&Archive Documents]
CommandLine=k:\utils\winlha.exe
WorkingDirectory=l:\
NovellGroup=EVERYONE

[Change &Password]
CommandLine=k:\netware\nwtools.exe /setpass
WorkingDirectory=k:\netware
NovellGroup=EVERYONE

[Con&trol Panel]
CommandLine=w:\control.exe
WorkingDirectory=
NovellGroup=SUPPORT;HELPDESK

[&Developer Network CD]
CommandLine=MVIEWER2.EXE msdncd7.mvb
WorkingDirectory=E:\
NetwareVolume=MSDNCD
DriveLetter=E
FileServer=EUROKA
NovellGroup=PCAPPS
PromptOnDriveRemap=0
GenericLoginId=MSDNCD
GenericPassword=WTT^eac[Sebc][Pff`lVKdZ/`]QY
PromptIfLoggedIn=0
DisconnectOnTermination=1

[C&haracter Map]
CommandLine=w:\charmap.exe
WorkingDirectory=
NovellGroup=EVERYONE

[&Map Drive]
CommandLine=k:\netware\nwtools.exe /map
WorkingDirectory=k:\netware
NovellGroup=SUPPORT

[Phone &Book]
CommandLine=v:\etd\etd.exe etd.etd
WorkingDirectory=v:\etd
NovellGroup=EVERYONE

[&Send Message]
CommandLine=k:\netware\nwtools.exe /send
WorkingDirectory=k:\netware
NovellGroup=EVERYONE

[&Virus Scan]
CommandLine=k:\utils\vscan.exe
WorkingDirectory=
NovellGroup=EVERYONE

[Manage Pass&words]
CommandLine=k:\utils\wsetpass.exe
WorkingDirectory=i:\%user%
NovellGroup=SUPPORT

[Change P&rint Queue]
CommandLine=k:\netware\nwtools.exe /printers
WorkingDirectory=k:\netware
NovellGroup=EVERYONE

[&Flex Sheet]
CommandLine=h:\flex\%user%.xls
WorkingDirectory=h:\flex
NovellGroup=OFF_SYS

```

NWMenu section

This section is optional. It allows you to specify the title bar for the application (the NWMenu= entry), the width of the pushbuttons, the first network drive, a generic login name and (encrypted) password for transparently attaching the user to a file server, and a few other

options for specifying how NWMenu behaves at runtime. The entries which can be set in the NWMenu section are as follows:

NWMenu=<string-of-characters>

Default: NWMenu

Purpose: This is the text that NWMenu displays in its title bar.

ButtonWidth=<number>

Default: 30

Purpose: This is the width of the buttons that NWMenu creates in its window. The number indicates the approximate number of characters that can be displayed on the button face.

FirstNetworkDrive=<letter>

Default: F

Purpose: The first drive on the network. Used for determining what format to request from Netware when ascertaining drive mappings.

GenericLoginId=<string-of-characters>

Default: None

Purpose: For bindery users and applications which are on different servers, a generic login id to be used for attaching to the server. For NDS users, NWMenu will attempt to authenticate them to the server on which the application resides.

GenericPassword=<string-of-characters>

Default: None

Purpose: The password for the generic login id. The password *must* be encrypted. See the section below on attaching to servers for more details on storing passwords.

CenterOnScreen=<0-or-1>

Default: 0

Purpose: This controls whether the NWMenu window is centered in the middle of the screen or on the position of the mouse pointer. The default is that the window is centered on the mouse pointer.

AllowMultipleInstances=<0-or-1>

Default: 1

Purpose: This controls whether you can have more than one copy of NWMenu active at once. If this is set to 0, then you can only have one copy of NWMenu active at a time. If you try to start a second copy it merely activates the existing instance and brings it to the front. The default setting allows you to start multiple copies of NWMenu. If you want to use the automatic logout capabilities of NWMenu to detach from a file server when an application terminates, leave this at the default.

AllowbinderyAndNDS=<0-or-1>

Default: 1

Purpose: This determines whether a Netware 4 NDS user can attempt to attach to a file server as a bindery user if they are unable to authenticate to it. For example, your users may be using Netware 4 for your normal applications but using CD-View for networked CDs. CD-View only emulates a bindery server, so your NDS users need to be able to attach to it as bindery users. The default setting is that an NDS user can attempt a bindery attach if authentication fails. Set this entry to 0 if you want an error if authentication for an NDS user fails.

Internationalizing NWMenu

Most of the NWMenu prompts and menus can be displayed in the language of your choice, provided that it supports the roman character set (i.e. a–z and A–Z). You can add the following entries to the [NWMenu] section to change the default menus and prompts:

HelpMenuItem=<string-of-characters>

Default: Help...

Purpose: To change the text that NWMenu shows for the "Help.." item that it adds to the system menu. For example, if you users want their prompts to appear in French, you would add HelpMenuItem=Aide... to the [NWMenu] section of the INI file.

AboutMenuItem=<string-of-characters>

Default: About...
Purpose: To change the text that NWMenu shows for the "About..." item that it adds to the system menu.

HelpText=<string-of-characters>

Default: Select Help... from the System menu to view the default value.
Purpose: To change the text that NWMenu displays in the message box that it shows when "Help..." is selected from the system menu. It is not possible to duplicate the layout of the default English text, as the INI file format does not allow you to enter the control characters needed to specify line breaks and tabs. Whatever text you use here will appear as a single paragraph. You are limited to a maximum of 511 characters.

OkButtonText=<string-of-characters>

Default: OK
Purpose: If you use a language other than English, or you want something other than OK to appear on the button, specify the text you want to appear on the face of the OK button in the Login dialog. If you have set up autologin for your users they will not see this dialog.

CancelButtonText=<string-of-characters>

Default: Cancel
Purpose: To change the text you want to appear on the face of the Cancel button.

LoginDialogText=<string-of-characters>

Default: Enter userid and password for attaching to %s to use application '%s'.
Purpose: The prompt text that appears in the login dialog to inform the user of the purpose of the dialog. If you change this from the default, you *must* include the two %s fields in the replacement string. The first %s is replaced with the name of the server containing the application or file, the second %s is replaced with the application name. If you omit these fields, NWMenu will GPF when it tries to substitute the values.

LoginNameText=<string-of-characters>

Default: &Login Name:
Purpose: The text that appears in the login dialog above the field where the user specifies their login name.

PasswordText=<string-of-characters>

Default: &Password:
Purpose: The text that appears in the login dialog above the field where the user specifies their password.

Applications section

This lists all the applications that NWMenu can display. They are merely entered as App1=..., App2=... up to App30. NWMenu limits you to thirty buttons, but realistically, from the user's point of view, anything more than about a dozen buttons starts to get confusing. The name that you specify here is what will appear on the buttons in the NWMenu window.

The applications will be displayed in application order: the App1= application first, the App2= application second, and so on. Each application name that you list here must have a corresponding section for the application.

If you want users to be able to use the keyboard as well as the mouse to select an application, put an ampersand (&) in front of the mnemonic character to be used. This character will appear underlined on the button face and the user can use Alt+character to select the application.

The section for a particular application

The section name must be identical (including any ampersand, if used) to the name as given in the [Applications] section. For a normal application, there are two required entries in each section:

CommandLine=<string-of-characters>

Default: None
Purpose: This entry is required, and is identical to the command line you would use if you were setting up an icon to run the application. You can use DOS environment

variables on the command line if you wish, or set up a document with an association (as with the flex spreadsheet above, which will start a copy of Excel if needed or load the sheet into an existing copy of Excel if it is already running. If no value is entered, NWMenu will report an error when it starts.

NovellGroup=<string-of-characters>[;<string-of-characters>]

Default: None

Purpose: This entry is required, and is the group or groups of which the user must be a member in order to see a button for starting this application. If you want everyone to see the button, make the group EVERYONE (but note, if you are using Netware 4.x, that group EVERYONE does not necessarily exist on a new Netware 4 installation, and that new users are not, by default, made members of group EVERYONE). If you specify multiple groups here, they must be separated by semicolons, and do not add spaces between the groups, as the spaces will be interpreted as part of the group name.

The other entries for the section for an application are optional, except that if you are setting up an application that attaches the user to another server, you must specify the server name, the volume name, and the drive letter:

WorkingDirectory=<string-of-characters>

Default: None

Purpose: This is identical to the working directory you would use if you were setting up a normal icon. If the application needs no working directory, you can leave this entry blank. You can use DOS environment variables for the working directory if you wish.

For an application on another server, there are three entries that must all be specified:

FileServer=<string-of-characters>

Default: None

Purpose: This is the name of the Netware server on which the volume is mounted.

NetwareVolume=<string-of-characters>

Default: None

Purpose: This is the name of the Netware volume.

DriveLetter=<string-of-characters>

Default: None

Purpose: This is the drive letter to which the volume will be mapped before attempting to execute the application. If you want to root map a directory on the volume, see the *RootOfDrive* entry below.

There are also several optional entries:

RootOfDrive=<string-of-characters>

Default: None

Purpose: If you want to root map a directory on the volume, specify it here. Specify it in the format DIRECTORY[/SUBDIR]. If you omit this field or leave it blank, the root of the volume will be mapped to the drive letter.

GenericLoginId=<string-of-characters>

Default: None

Purpose: For bindery users, if a login id is specified here, it is used for this application instead of any login id that is specified in the [NWMenu] section of the INI file. For NDS users, NWMenu will attempt to authenticate them to the server.

GenericPassword=<string-of-characters>

Default: None

Purpose: This is the (encrypted) password for the application-specific login id.

PromptOnDriveRemap=<0-or-1>

Default: 0

Purpose: This determines whether the user will be prompted if the CD volume is being mapped to a drive letter that is already in use. If it is set to 1, they are queried whether they wish to change the drive mapping.

PromptIfLoggedIn=<0-or-1>

Default: 0

Purpose: This determines whether the user will not be prompted if they are already logged into the file server and the application specifies a different login id to the id under which they are already logged in. If it is set to 1, they are queried whether they wish to login using the new id, or remain logged in using the old id, or cancel the operation. If it is set to 0, they are automatically logged in using the new login id.

DisconnectOnTermination=<0-or-1>

Default: 0

Purpose: This entry determines what action NWMenu takes when it starts an application. If it is 0, when the application is started NWMenu will terminate. If it is set to 1, when the application starts NWMenu will hide itself but remain running. Every 30 seconds it will test to see if the application is still running. When the application terminates, and if there are no other drives mapped to the file server to which it attached for this application, the user will be disconnected from the server. (See also the section on DISCONN below for another method of automatically disconnecting from a server).

Note: if you set this entry to 1 for any application, you should also ensure that AllowMultipleInstances is not set to 0 in the [NWMenu] section of the INI file, or the user will not be able to start another copy of NWMenu while a previous version is waiting for an application to finish.

If you look at the sample INI file above, there are three applications (*Control Panel*, *Map Drive*, and *Manage Passwords*) which a user will not see unless they are a member of the SUPPORT group, and one CD application (*Developer Network CD*) which will only be seen by members of the PCAPPS group, and which uses its own login id and password. All other entries are available to all users.

If you put an APPn =... in the [Applications] section, and there is not a correspondingly named section for the application, NWMenu will report an error at startup.

NWMenu and applications on other servers

The way that NWMenu deals with these applications is as follows:

1. It checks to see whether the user is already attached to the file server specified for the application.
2. If the user is not logged in and is an NDS user, NWMenu will attempt to authenticate the user to the server. If the user cannot be authenticated, NWMenu will attempt to connect as a bindery user provided that AllowbinderyAndNDS is not set to 0 in the [NWMenu] section of the INI file.

For bindery users, if the user is not logged in, and there is a generic login id and password specified in the section for that particular application in the INI file, NWMenu decrypts the password and attempts to attach to the server. In the absence of an application-specific generic login id, NWMenu will use the GenericLoginId and GenericPassword entries in the [NWMenu] section if they exist. If there is no login id in either location, NWMenu shows the user a dialog box in which to enter a login id and password for attaching to the server.

If the user is already logged in as a bindery user, the action taken depends on what ids and passwords are specified. If there is no application-specific login id, the user's existing connection is used. If there is an application-specific login id which is different from the existing login id, and PromptIfLoggedIn=0, the user is automatically logged in with the new id. If PromptIfLoggedIn=1, the user is queried whether they wish to login again using the new id.

3. Once a connection to the server has been established, NWMenu checks to see whether the drive letter specified is already in use. If it is, and PromptOnDriveRemap is set to 1, the user is asked whether to change the drive mapping.
4. Once the drive mapping has been made, the application is started.

Setting up access to other servers

These suggestions can apply equally to networked CD-ROMs or to any application where users need to attach to another server to run the application.

I suggest you set up security for these applications in the following way. You could give each user their own login to the server, but it is simpler to setup and maintain if either a single login

is used for all applications, or, if you want more control, if each application has its own login id.

1. On the users normal login server, set up a Netware security group for every volume or application which will be accessed, except those which are available to every user. Put the users of the applications in these groups, and use these groups for the entries in the NWMenu INI file, so that users will only see the applications to which they should have access.
2. On the CD or application server, set up the generic login id(s) for access. You only need to set up as many login ids as you have GenericLoginId entries in the INI file. These will be the login ids for all users who attach to the server to use an application. Under Account Restrictions in SYSCON, set the maximum number of connections to whatever limit you wish to set, and do not allow the password to be changed by the user.
If you have only a single login id, you can either make this user a member of group EVERYONE, and make EVERYONE a trustee of the volumes (if you are mapping CD volumes), or create a special group to be a trustee of the volumes, and make the generic userid a member of this group (and only a member of this group). The second method will limit the user's rights to only the CD volumes.
For CD applications, if you have a different id for each application, you can give each of these ids their own group and make that group the trustee for the volume. This restricts use of that CD to only those users who login using the id for that CD. However, if you set things up this way you have a potential problem:
 - a) A user selects a CD application and is logged into the server using the login id for that application.
 - b) The user minimises that application, but leaves it running, and starts another CD application which uses a different id.
 - c) If the user logs in with the second id their first application gets trashed, and if they don't log in under the second id, the attempt to map the second CD volume will fail because the id under which they are logged in does not have trustee rights to that volume.
3. Enter the generic login id in either the [NWMenu] or application section of the INI file, as:
`GenericLoginId=<loginname>`
4. Run PASSENCR.EXE and enter the password for the generic userid. Click the Encrypt button and the encrypted password will be copied to the clipboard. Paste it into the INI file as:
`GenericPassword=<encrypted password>`
5. If your first network drive for your users is something other than F:, enter the drive letter in the [NWMenu] section as:
`FirstNetworkDrive=<driveletter>`
6. Save the INI file and create an icon which has it on the command line, *including* the full path to the INI file. e.g.
`NWMENU.EXE W:\NETCDROM.INI`

Setting up automatic logout from the attached server

You have two methods for automatically logging users out of a server when they finish using an application.

1. Add the **DisconnectOnTermination=1** entry to that application's section in the INI file. This will sever the connection to the server when the application terminates, provided that no other drive is mapped to that server. This is the easiest method to use. It does mean, though, if the user has started several such applications and still has them running, that several instances of NWMenu are still active in the background, waiting for the applications to close.
2. Use the separate program DISCONN.EXE to terminate the user's connection to the servers entered in the DISCONN.INI file if they are no longer using any applications on those servers. DISCONN does this by periodically checking whether the user has a connection to each server (every 30 seconds), and if there is a connection it then checks to find what files the user has open. If the user has no files open their connection to that server is terminated. There are, however, a couple of potential problems with this second approach:

- a) For a user to be able to find out what files they have open, they need console operator privileges. This is not *quite* as bad as it sounds, as unless the user knows either the console lock or Supervisor password, RConsole will not let them in, and if they don't have R F rights to SYS:SYSTEM then they can't see RConsole at all. If you are using the generic signon approach then the users don't even know the password for the userid they are using to attach to the CD server.
- b) It's possible for a CD application to be running but not actually have any files on the CD open. For example, your user may have Coreldraw installed on a local drive but want access to the clipart on a CD which is mounted as a Netware volume. Unless they actually have some clipart open, they will have no files open on the CD. For the more normal business-type CD applications that I tested, if the application was running then it had files open on the CD, so this may not be a problem, but test your applications first.
- c) You have to setup DISCONN separately, and add it to the Run= line of every user's WIN.INI file.

How to setup DISCONN

Note: If a user does not have console operator privileges, there is no point setting them up to run DISCONN. If they don't have console operator rights on a server, DISCONN will never terminate their connection to the server because it cannot determine whether they have any files open on the server.

If you want to use DISCONN, this is the procedure to follow:

1. Create a DISCONN.INI file which contains the name(s) of the server(s), as below. If you don't want users to see the icon for DISCONN, add the Hidelcon=1 line. This will prevent them from being able to terminate the application.

```
[DisConn]
Server3=EUROKA
Server2=OS-TEST
Server1=ANN-A
Server4=ACT
HideIcon=1
```

You can have up to 100 server names in the file. Entries must be numbered consecutively from 1 with no gaps, though there is no requirement for the entries to appear in the file in ascending order. The server name(s) should be in upper case. If Hidelcon is set to 1 then the user cannot see the program or terminate it. If Hidelcon is 0 or not present then the DISCONN icon can be seen and the program can be closed by bringing up the task list and selecting End Task.

2. Place this INI file in the same directory as DISCONN.EXE, where it can be shared by all users. If you want an individual user to have a different setting, you can place a copy of the INI file in their Windows directory, where it will be found before the shared copy.
3. Place DISCONN.EXE on the Load= or Run= line of every user's WIN.INI. While Windows is running it will run in the background, doing nothing until the user attaches to one of the nominated servers, at which time it will start checking for open files every thirty seconds. As it is a very small program (< 15k), and consumes no resources apart from a timer, it's very unlikely that it will have any effect on any other programs. I have had no reports of any side effects.

NWMenu and Netware 4

NWMenu completely supports Netware 4.x. It makes NDS calls if run by an NDS user and bindery calls if run by a bindery user. You do not need to load bindery support (BIND.VLM and NETX.VLM) on the workstation, provided that all the servers to which your users can potentially attach are Netware 4.x servers.

For applications which are on another server, if you are not authenticated to that server, NWMenu will authenticate you to the server and then map the drive. If you cannot be authenticated, and you have not set **AllowBinderyAndNDS=0** in the [NWMenu] section of the INI file, NWMenu will attempt to attach you as a bindery user.

Registration

NWMenu is marketed as shareware. You are granted a 30 day trial period, after which you are required to register the product. The cost of a license is \$US150 for a license for a single server, allowing you to run NWMenu on any workstation attached to that server.

Discounts are available for site or multiple server licences. The pricing structure is as follows:

First five servers	full price
Servers 6–20	25% discount
Servers 21–50	50% discount
Servers 51–100	75% discount
Servers over 100	90% discount

Or you can purchase an unlimited site licence for \$US8000. This puts the break-even point at about 195 servers. If you have more servers than that, the unlimited site licence is cheaper.

To register the program for a server licence, either:

on CompuServe, GO SWREG at any ! prompt and quote number 2108

or print out the registration form included here and post to:

Nick Payne
P.O. Box 4384
Kingston ACT 2604
Australia

Registered users will be sent a disk containing the latest version of NWMenu, a separate program for maintaining the INI file used by NWMenu, a registration number to disable the splash screen that comes up when NWMenu starts an application, and will be informed of any further upgrades that become available. Registered users can get support through Compuserve mail to 100033,432.

Entering your Registration information

Once you have received your registration information, add the following entries to the [NWMenu] section of each shared INI file:

Name=<name>
Company=<company>
Key=<key>

All details must be entered exactly as you received them, including case and punctuation. If Name or Company is blank, put nothing after the equals sign for that entry.

I would be pleased to receive any comments, bug reports or suggestions for future releases. I may be contacted by E-Mail at:

CompuServe: 100033,432

or by post at:

PO Box 4384
Kingston ACT 2604
AUSTRALIA

NWMenu Shareware registration form

Please complete order details (at minimum licensee name and/or company (preferably company), address, and payment details) and either:

submit via the SWREG forum on Compuserve (quote number 2108)

or post to: Nick Payne
P.O. Box 4384
Kingston ACT 2604
Australia

Use a separate order form for each licensee.

Please give an e-Mail address if you have one so I can inform you of your registration code as soon as I receive your order, and give timely information on upgrades.

ORDER FORM

Name: _____

Company: _____

Address: _____

Phone: _____ Fax: _____

CompuServe: _____

____ server licenses at USD \$150 _____

Do you already have a copy of the program? Y/N _____

If so, what version do you have? _____
(select About from the system menu and click
on Credits to see the version number)

If not, please add shipping and handling

Shipping & handling (international post orders)

_____ USD \$10 _____

Total _____

Enclose check/P.O./money order for total (or converted equivalent):