NORMAN®

Norman Virus Control

Administrator’s Guide

Version 4.70

Norman ASA
Mailing address: P.O. Box 43, N-1324 Lysaker, Norway   Physical address: Strandveien 37, Lysaker
Tel. +47 10 97 00 Fax. +47 67 58 99 40 E-mail: norman@norman.no

Norman Data Defense Systems Inc
9302 Lee Highway Suite 950a, Fairfax, VA 22031, USA
Tel. +1703 267 6109 Fax. +1703 934 6367 E-mail: norman@norman.com

Norman Data Defense Systems GmbH
Kieler Str. 15, D-42697 Solingen, Germany
Tel. +49 212/26718 0 Fax. +49 212/26718 15 E-mail: norman@norman.de

Norman/SHARK BV
Mailing address: P.O. Box 159, NL-2130 AD Hoofddorp, The Netherlands
Tel. +31 23 563 3960 Fax. +31 23 561 3165 E-mail: sales@shark.nl

Norman Data Defense Systems AG
Postfach, CH-4015 Basel, Switzerland
Tel. +41 61 487 25 00 Fax. +41 61 487 25 01 E-mail: norman@norman.ch

Norman Data Defense Systems Pty. Ltd.
6 Sarton Road, Clayton, Victoria, 3168 Australia
Tel. +61 3 9562-7655 Fax. +61 3 9562-9663 E-mail: norman@norman.com.au

Limited warranty

Norman guarantees that the enclosed diskette/CD-ROM and documentation do not have production flaws. If you report a flaw within 30 days of purchase, Norman will replace the defective diskette/CD-ROM and/or documentation at no charge. Proof of purchase must be enclosed with any claim.

This warranty is limited to replacement of the product. Norman is not liable for any other form of loss or damage arising from use of the software or documentation or from errors or deficiencies therein, including but not limited to loss of earnings.

With regard to defects or flaws in the diskette/CD-ROM or documentation, or this licensing agreement, this warranty supersedes any other warranties, expressed or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

In particular, and without the limitations imposed by the licensing agreement with regard to any special use or purpose, Norman will in no event be liable for loss of profits or other commercial damage including but not limited to incidental or consequential damages.

This warranty expires 30 days after purchase.

The information in this document as well as the functionality of the software is subject to change without notice. The software may be used in accordance with the terms of the license agreement. The purchaser may make one copy of the software for backup purposes. No part of this documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or information storage and retrieval systems, for any purpose other than the purchaser’s personal use, without the explicit written permission of Norman.

The Norman logo is a registered trademark of Norman ASA.

Names of products mentioned in this documentation are either trademarks or registered trademarks of their respective owners. They are mentioned for identification purposes only.

March 1999

Table of Contents

Introduction

About This Manual

This manual provides advanced information about Norman Virus Control (NVC) for system administrators.

NVC is available on the following platforms:

- DOS/Windows 3.1x
- Windows 95
- Windows NT
- OS/2
- NetWare

and Norman also has SNMP extensions for messaging in TCP/IP environments.

Norman's workstation anti-virus products (NVC for DOS/ Windows 3.1x, Windows 95, Windows NT, and OS/2) have two types of documentation: "Installing and Getting Started", which is intended to be used for quick starts and "User's Guide", which is intended for the more curious users.

Note: NVC for NetWare is a server-based product, and therefore there is only one manual (the "User's Guide") which is intended solely for system administrators.

The User's Guides for the workstation anti-virus products don not delve too deeply into technical matters. Instead, we have saved these topics for this manual in addition to chapters that concern NVC and network communications.

Pre-requisites

To take full advantage of all the functions in Norman Virus Control, you should have a good understanding of the different modules in NVC and how they work together. Please refer to the appropriate “Installing and Getting Started” and the “User’s Guide” for more details.

In addition, you should have detailed knowledge about the operating system(s) as well as the network installation in your organization.

Conventions

Throughout this manual, we use several typeface conventions.

Examples of commands that should be typed or messages that appear on the screen look like this:

```
format a: /s /u [Enter]
```

If certain keys should be used, they will appear with square brackets around the name of the key, as in:

[Ctrl]

When we describe a series of menu choices for you to choose, we will use the following:

Start|Run

Important notes appear as:

Note: This is important...

And particularly important text appears in **bold**.

General Information

Unless otherwise stated, the information in this chapter applies to all platforms.

New Utility for Central Installations and Updates

We still provide our script-based distribution program N_DIST for central installations. In addition, we're now introducing a GUI-based version of N_DIST for visual central installation, Norman Package Distribution. See "Norman Package Distribution" on page 40.

The Scanning Engine and the Host Program

The scanning engine used to be integrated in the host program. As of NVC v4.50 the scanning engine was split from the host program. In previous versions an entirely new version of NVC was required when viruses that necessitated code changes emerged. With the present solution we are able to ship updated definition files with a new DLL (Windows NT and OS/2) or VxD (Windows 3.1x and Windows 95).

Note: Changes to the format in the definition files may render the command line scanner useless. If such a situation should occur, we will supply an updated version of `nvc32.exe/nvc32X.exe`.

For Windows 3.1x and Windows 95 users:
When you recieve updated definition files and a new VxD, you need to reboot your machine after installation.

Scanning Large Hard Drives

PC hardware and software today are very different from the time the first personal computers were introduced in the early eighties. However, some of the original features are still affecting PC users today, like DOS and BIOS, for example.

There is a BIOS limitation in the way a hard drive is read that prevents the scanner to 'see' beyond the first 8 GB. Teoretically, *system boot sector viruses* on hard drives bigger than 8 GB may elude detection. That is, if you are scanning a 10 GB hard drive, we cannot 'see' the last 2 GB.

However, it is not very likely that bootable partitions are located at this area of the hard drive.

Note: The scanner will detect master boot sector viruses regardless of drive size, as these viruses are found in the first sector.

Specifying UNC Paths

NVC now allows the use of UNC (Universal Naming Convention) names and environment variables wherever file and directory names can be specified:

- Style name / areas
- Report to file
- Move infected files

Access to files on a network server is normally done via mapping network resource names to drive letters. For example, on your workstation the drive letter K: is mapped to the share name DATA on SERVER1. Other workstations may map the same network resource as P:

Differences in drive mapping can cause problems when you are creating NVCxx styles on one workstation to be used on other workstations. UNC allows you to specify a complete path to a resource on a network server, without dependening on specific drive mappings:

```
\\ServerName\PathName\Filename.typ
```

Environment Variables

Environment variables are used by the operating system to inform the programs about the environment they execute in. Some of these variables are defined in CONFIG.SYS as for example:

```
SET AUTOSTART=TASKLIST,FOLDERS,WARPCENTER
    |        |
    |        +- Environment contents
    +----------- Environment name
```

Other variables are set when you log into the network, as SET USER=hka. These environment variables can be utilized whenever specifying a file or path name in NVCxx:

[x] Report to file
```
[\\SERVER\LOGFILES\NRM_%user%.RPT]
```

The text %user% will be replaced by the contents of the environment variable USER when the scanning is performed, resulting in the file name

```
\\SERVER\LOGFILES\NRM_HKA.RPT
```

The file name will be expanded to

```
\\SERVER\LOGFILES\NRM_.RPT
```

if the environment variable USER is undefined or empty.

NVCxx allows using UNC names and environment variables when specifying paths as in all other places where filenames and directories are specified:

Report File Structure

We have structured the report file to be cross-platform consistent. The NVC report file consists of:

- A file header, stating the program name and version.
- A scan report section, containing information about directories and files scanned, and possible viruses, if encountered.
- A summary section.

File header

The file header states the program name and version:

```
>>>
Norman Virus Control for xx v4.60
```

Any user defined start-up or bundling messages will also appear here.

Scan report section

The scan report section describes areas scanned, all viruses found, and optionally, all directories and/or files scanned. The following line headers appear in this section:

< Section start. Except for this symbol, the line is empty.

> Section end. Except for this symbol, the line is empty.

- Informational message. See below.

* Error message. See below.

*** Indicates a possible virus infection.

Scanning *removable drives* will be logged with volume labels as:

```
- Scanning drive A: (Volume: DISK1, serial
number A6FB-3814)
- Scanning system areas of drive A:
- Searching for files on drive A:
```

Scanning *fixed drives* will be logged as:

```
- Scanning drive D:
- Scanning system areas of drive D:
- Searching for files on drive D:
```

Scanning *network drives* will be logged with UNC names as:

```
- Scanning drive N: (\\HANSOS2\LIBRARY)
- Searching for files on drive N:
```

Scanning a *directory* (not entire drives) will be logged as:

```
- Scanning files in the directory: E:\Viruses\
```

Scanning for *files* matching a given pattern will be logged as:

```
- Scanning files matching: E:\Viruses\*.exe
```

A *virus infection* will be logged as shown below, with the full path and filename as well as the virus name.

```
*** Possible virus infection found ***
*** D:\VirusSamples\aids2.COM -> HLLC.Aids.8064
```

Automatic deletion of infected files will be logged as:

```
- File D:\VirusSamples\aids2.COM deleted.
```

Automatic moving of infected files will be logged as:

```
- File D:\VirusSamples\aids2.COM moved to
C:\NORMAN\INFECTED.
```

All directories scanned will optionally be logged as:

```
  C:\
  C:\Work area\
  C:\Work area\Templates\
```

The two lead-in spaces on each line are, for clarity, symbolized as ’.’.

All files scanned will optionally be logged as:

```
   E:\NORMAN\
      NVC.EXE
      NVCPM.EXE
```

The five lead-in spaces in front of each filename are, for clarity, symbolized as ’.’.

Infected files within an archive file will be logged as follows:

```
- Unpacking archive: Virus.ZIP
*** F:\VirusSamples\Virus.ZIP : aids2.COM ->
HLLC.Aids.8064
*** F:\VirusSamples\Virus.ZIP : aidsload.EXE ->
HLLO.Number_1 related
```

The directory set up as target for infected files will normally not be scanned. This will be logged as:

```
- Files in C:\NORMAN\INFECTED\ NOT scanned.
```

See also error messages in report files and the report file example.

Add Your Own Help File

Some organizations have established routines for handling virus infections. If you want to incorporate these routines into NVC, you can generate your own help file, name it `USERDEF.HLP`, and place it in the `X:\NORMAN\WIN32` directory. When a virus is detected, NVCNT/NVC95 will look for `USERDEF.HLP` when the help function is invoked. If `USERDEF.HLP` is not found, the default `NVCNT.HLP` or `NVC95.HLP` is launched.

Norman Internet Update

Note: Norman Internet Update is only available on Windows NT and Windows 9x.

We assume that most network administrators will prefer to be in charge of all software running on the clients, including identical versions of NVC. Therefore, we do not recommend that you distribute this feature to the workstations. During an administrator installation, the executable `nupgrade.exe` is copied to the norman\win32 directory, but by default it's not included in the N_DIST script. In the N_DIST script, Norman Internet Update will appear like this:

```
set $Inst_InternetUpdate = 'No'
```

Replace 'No' with 'Yes' if you wish to distribute Norman Internet Update to the workstations.

Fetching NSE upgrades is a task that ought to be carried out from one central point, and the process is quite simple. When you have entered the authentication key, NVC will check the time stamp on the files in your NSE directory with the set of files on the Norman file server (definition files, DLL/VxD). When an update is available and you choose download, a self-extracting zip file called NSEUPDAT.ZIP is downloaded to your `x:\nvcadmin\nse\update` directory.

A separate section in `nvc32.cfg` describes the name and destination path for the updated files. This section will appear like this:

```
[NseUpdateInfo]
Package file name = 'NSEUPDAT.ZIP'
Package destination path = 'D:\NORMAN\NSE\UPDATE'
```

When a fresh zip file is downloaded to the `\update` directory, the files in the current `\NSE` directory are copied to the `\backup` directory. The zip file is then copied from the `\update` directory to the `\NSE` directory, where they are extracted.

Administering Security Options on the Workstations

About NCFTE

NCFTE is a command line utility made for creating and editing the file `nvc32.cfg`. NCFTE is available on the following platforms: Windows 9x, Windows NT, and OS/2. It's installed in the norman\win32 directory.

Note: For NVC NT Service, special measures are called for, as described in "Updating the NVC NT Service" on page 12.

How to use NCFTE

If you run the program without any parameters, you will see the contents of the `nvc32.cfg` file in the current directory. Other files (of the same format) can be displayed by specifying the file name on the command line:

```
NCFTE /F:NVC32ADM.CFG
```

NCFTE options

To display a list of available commands, type

```
ncfte /?
```

and the following list of options appears:

`/S`	Edit security options.
`/U`	Edit common update section (URLs etc).
`/NSE`	Edit NSE update section.
`/F:[filename]`	Specify file. Default file is NVC32.CFG.
`/H`	Show help.
`/?`	Show help.

Editing the security options

The "Security Options" section is used by the NVC applications (v4.6, v4.7) to allow or deny the user to change certain options within the programs.

Note: ‘CR’ is short for carriage return.

```
NCFTE /S
```

```
[NORMAN]
Norman Configuration Tool Version 1.00.00
(c) Norman Data Defense Systems 1993-1999

Configuration file: NVC32.CFG
Update section [SecurityOptions]:
Allow user to change:
Scanning Options (No, CR=Yes)?
```

Press Enter to keep the current value, Y to allow the user to change the “Scanning options”, or N to disallow.

Continue to answer Enter/Y/N to all the questions.

When you have edited all options, you will be asked whether you want to keep the changes. The data is saved only if you confirm.

Editing the common update section

The “Common Update Section” is used by the NUPGRADE and NSE (Norman Scanner Engine) component of the NVC applications to locate updates of itself:

```
NCFTE /U
```

```
[NORMAN]
Norman Configuration Tool Version 1.10.00
(c) Norman Data Defense Systems 1993-1999

Configuration file: NVC32.CFG
Update section [AuthenticationInfo]:
Authentication key = ’rurre’
Input new value (CR=keep current)
```

Press Enter to leave this field unchanged. Type the customer validation code/password.

The format of this field will probably vary from country to country.

```
Authentication server[0] = ’www.norman.no’
Input new value (CR=keep current)
```

Change the server name(s) in accordance with a valid address for your country.

Editing the NSE Update Section

The “NSE Update Section” is used by the NSE components of the NVC applications to locate updates of itself:

```
NCFTE /NSE
```

```
[NORMAN]
Norman Configuration Tool Version 1.10.00
(c) NORMAN Data Defense Systems 1993-1999
```

```
Configuration file: NVC32.CFG
Update section [NseUpdateInfo]:
Package file name = ’NSEUPDAT.ZIP’
Input new value (CR=keep current):
```

The format of this field will probably vary from country to country. It is normally set when installing NVC.

```
Package destination path = ’D:\NORMAN\NSE\UPDATE’
Input new value (Enter=keep current):
```

Press Enter to leave this field unchanged. It is normally set when installing NVC.

Updating the NVC NT Service

In most environments, using the settings in the `NVC32.CFG` file is an effective way to perform automatic updates of the NVC scanning engine. This means that the ordinary NVC user will always use the latest version of the virus definition files for scanning. To accomplish this, share out the directory `[drive]:\nvcadmin\nse\update` as NVCUPD where everybody has read rights and the administrator all privileges.

Now the system administrator can put the `nseupdat.zip` file that contains the latest scanning engine and definition files in the `[drive]:\nvcadmin\nse\update` directory or the `\\machinename\nvcupd` share. All 32-bit NVC components on the workstations will now look for and use the updates in the share.

This is a solution that works well for NVC GUI components like NVC95 or NVCNT. The same is the case with the right-click scanner and the command line scanner (NVC32.EXE).

However, the NVC NT Service is different. If an NVC NT Service needs to be updated in the manner descibed above, the following additional actions are called for:

1. When configuring the `nvc32.cfg` file the administrator must use UNC path rather than the mapped network drive
2. The NT machine where the update package (.zip-file) resides, needs to be prepared to be accessed by an NVC Service on another machine.

About 1):

The reason you must use UNC path is that the NVC NT Service is usually started automatically when the NT machine is turned on. At this point no user is logged on and no mapped network drive is available. Even if the NVC Service is started at at later point it will not run in the context of the logged on user. Thus, to make automatic updates work with the NVC NT Service, the administrator should always use UNC path when configuring the path for the update package in the `nvc32.cfg` file. Instead of using `[drive]:\nvcadmin\nse\update`, rather use `\\machinename\nvcupd`, where ‘machinename’ is the name of the machine and ‘nvcupd’ is the share name of the `[drive]:\nvcadmin\nse\update` directory that contains the updates.

Note: This is a solution that will also work with the other NVC components.

About 2):

The general security design of Windows NT does usually not allow services on other machines access to local shares. Even if an administrator has made an `\update` share available to every user in the network, this does not automatically include services on other machines.

The solution to this is to give access to the NT machine that contains the upload package by using the following procedure:

1. Get a copy of the NVC NT Service configuration program `NCFG.EXE` (v 4.70 or newer) and copy it to the machine where the uploads reside.
2. Note that name of the common share name of the directory that contains the uploads. Using the sample above, this would be `nvcupd`.
3. Now issue the command:
   ```
   ncfg –updateshare:nvcupd
   ```
4. The preparation is complete. The updates are available also to the NVC NT Service

Note that the procedure described above is a one-time operation. All the administrator needs to do is to place new updates here as soon as desired. The operation needs to be repeated only if the common share name is changed .

If you at a later stage want to deny services access to this share, then issue the command:

```
ncfg -deleteupdateshare:nvcupd
```

Case study

Designing a network for automatic updates:

1. Sample site: 100 machines: 50 Windows 9x and 50 Windows NT.

2. The administrator decides that the nvc updates should reside on the NT machine NTCOMMON.
3. The administrator creates a directory on this machine where he wants to put the updated scanning engine.
4. This directory is shared out with the share name `nvcupd` and can be accessed by everyone in read-only mode.
5. Using `NCFTE.EXE` now design a `nvc32.cfg` that uses `\\ntcommon\nvcupd` as the update path and `nvcupd.zip` as the name of the package containing the updated scanning engine.
6. Now issue the command `ncfg –updateshare:nvcupd` on the same machine.
7. At this point the server is configured and the common `nvc32.cfg` is created. As a one-time operation this new `nvc32.cfg` now needs to be copied out to every machine, using N_DIST, for example.

Technical Details on Workstation NVC Products

There are several technical details about our workstation anti-virus products that are not included in either "Installing and Getting Started" or the "User's Guide". You will find these details in this chapter.

Canary

Canary is Norman's one-time non-resident bait for file viruses. Please see the NVC "User's Guides" for DOS/ Windows 3.1x, Windows 95, and OS/2 for more information.

Where to Install Canary

Canary should **not** be installed in a directory which can be accessed by several different users. If you wish to install on a server, you must make a separate directory for each person and install Canary in each directory. Canary should only be accessed by one person per directory.

There are three reasons for this:

1. Since Canary is bait for computer viruses, its program files must be freely accessible by all users. If access is restricted — if some users do not have write access, for example — the virus will not be able to infect the Canary files, and you will not be warned about the virus attack. But...
2. If you install Canary with full access rights on a network drive, this could be a very dangerous source of contamination. A virus on one workstation will infect

the Canary program, and when users on other workstations try to run the program, their workstations will become infected as well.

3. More viruses are being written with “stealth” capabilities, in hopes of being difficult to detect. To detect these viruses, Canary must communicate with DOS in a certain way. Some networks do not support this communication.

Canary and NetWare

Because of the way NetWare communicates with DOS, Canary v1.30+ will work correctly if you load it from a shared directory on a NetWare server only if you are running `VLM`s. Canary will, however, work correctly in a NetWare environment if you install it in a directory on a local drive. See the section below for information on running Canary on diskless workstations.

If you must load Canary from a network drive, you can use Canary version 1.20. This version does not, however, offer the same protection as later versions, particularly with regard to “stealth” viruses.

Canary and Diskless Workstations

If you need to use Canary on a diskless workstation, we recommend that you create a RAM disk and set up the system so that Canary is loaded to the RAM disk at startup. This requires RAM memory of 640 KB or more. You can then run Canary from the RAM disk.

In the following examples, we assume that DOS is located in `C:\DOS`, and we install the RAM disk in extended RAM.

1. Create a RAM disk with `CONFIG.SYS`. The RAM disk must have room for `CANARY.COM` and `CANARY.EXE` and some room to spare.

The program necessary for creating a RAM disk is included with DOS. The two most common programs are:

- VDISK.SYS (included with IBM's PC-DOS and MS-DOS from Compaq Computer Corp.)
- RAMDRIVE.SYS (included with most MS-DOS and Microsoft Windows).

To create a RAM disk using VDISK.SYS, type:

```
device=c:\dos32 512 16 /E
[Enter]
```

To create a RAM disk using RAMDRIVE.SYS, type:

```
device=c:\dos\ramdrive.sys 32 512 16 /E
[Enter]
```

The system gives the RAM disk the first available drive letter.

Refer to your DOS manual for more information about VDISK or RAMDrive.

2. When you have created the RAM disk, you must edit AUTOEXEC.BAT so that it copies CANARY.COM and CANARY.EXE to the RAM disk. In the following example, we assume that the RAM disk has been given the letter “G:”.

```
copy f:\utils\canary.* g:
[Enter]
```

Now Canary is ready to use. Remember that if the RAM disk is not included in the path, the RAM disk's drive letter must be specified when running the program.

NVC.SYS

NVC.SYS is Norman’s smart behavior blocker (a DOS device driver). It provides protection from known and unknown virus infections. Please see the “Norman Virus Control for DOS/Windows 3.1x User’s Guide” for more information.

NVC.SYS and QEMM

On some computers, when NVC.SYS is loaded before QEMM386.SYS, NVC.SYS warns that:

```
QEMM386.SYS "Attempts To Trace".
```

The "Attempts To Trace" message is normally generated by advanced viruses that single-step through memory in search of the original ROM BIOS entry point. If this information is found, the virus can then communicate directly with the ROM BIOS and bypass all virus monitoring programs.

QEMM uses the same technique in order to locate the same information, although for entirely different purposes.

Note: To avoid the possibility of getting this warning from NVC.SYS, load NVC.SYS **after** `QEMM386.SYS`.

Starting in version 7, QEMM has a feature which pushes almost all of the DOS operating system over the 640 KB memory region. In order to do this, QEMM’s file, `DOSDATA.SYS`, insists on being loaded as the first device driver in `CONFIG.SYS`. This results in `NVC.SYS` being loaded after `DOS-UP.SYS` and `QEMM386.SYS`. At this point, parts of DOS (or the ROM BIOS) are already hidden somewhere in memory. Upon loading, `NVC.SYS` checks memory for boot viruses starting at base memory (below 640 KB) and ending wherever the ROM BIOS is located (even above the 1 MB memory region), regardless of whether or not QEMM is running in `ST:` mode.

Since QEMM does not anticipate that any program will reach the "hidden" DOS or ROM region directly, QEMM will react in two ways:

1. The computer might hang. When this occurs, only the `NVC.SYS` startup box and copyright messages appear.
2. QEMM may print out messages about memory violation errors which in fact never occurred.

The same error happens with security or access control programs that load from the Master Boot Sector.

Note: When running `NVC.SYS` and pushing DOS high with QEMM, we recommend loading `NVC.SYS` with the `/A` parameter **after** QEMM. Note well that this parameter results in `NVC.SYS` not detecting boot viruses in memory.

If there is a boot virus active in memory, pushing DOS high with `DOSDATA.SYS` and `DOSUP.SYS` will also cause the computer to hang unexpectedly, regardless of whether or not `NVC.SYS` is running.

On most computers, it is possible to run `NVC.SYS` with QEMM without specifying `/A` — if DOS is not pushed high with `DOSDATA.SYS` and `DOS-UP.SYS`. However, conflicts will appear again if the user runs `OPTIMIZE`.

To solve **this** problem, either temporarily disable `NVC.SYS` or run `NVC.SYS` with the `/A` parameter.

NVC.SYS and Turbo IDE Cards

Most of the new Turbo IDE cards – mainly for Vesa Local Bus computers – provide three options for booting the computer: [T]urbo, [F]ast, or [N]ormal. For both the [T]urbo and [F]ast modes, the BIOS automatically decreases base RAM by 2 KB, copies a special part of the ROM which handles disk I/O into that area, and then revectors INT 63h to point to the usual ROM routine.

This process, however, looks very virus-like, and `NVC.SYS` will warn:

```
"Boot Virus Detected In Memory!"
```

The Turbo IDE card manufacturers have provided a way to disable the [T]urbo and [F]ast modes through hardware jumpers.

These two modes also conflict with some SCSI devices and network redirectors. In addition, they conflict with Windows when run in 32 Bit Access mode.

Note: The solution is to use the hardware jumpers or to use the `/A` parameter on `NVC.SYS`. Note well that using this parameter results in `NVC.SYS` not detecting boot viruses in memory.

NVC.SYS and PC-NFS

If you are running `NVC.SYS` in a PC-NFS environment, it is necessary that you run `NVC.SYS` with the `/T` parameter. If this is not done, `NVC.SYS` will not co-exist with PC-NFS, and the machine may hang. Inclusion of the `/T` parameter will not impair the ability of `NVC.SYS` to detect known viruses. Instead, `NVC.SYS` will detect the virus as it tries to infect and not when it goes resident in memory.

Prevent Scanning of Remote Drives

Note: By default, NVC scanners that are installed on the workstations in your organization are able to scan remote drives. If you wish to prevent a workstation from scanning remote drives, then create a 0 byte file called `NVCNET.CFG`.

If you wish to do this for more than one workstation in your organization, you should use Norman Package Distribution or N_DIST in order to distribute the `NVCNET.CFG` file. Please see the chapter “Central Installation and Updates” on page 23.

Suppress Warning

When NVC scanners find that the virus definition files (`nvcmacro.def` and `nvcbin.def`) are over 6 months old, they will display a message stating that fact. The virus definition files contain information about known viruses, and the files are updated on a regular basis. If you have failed to update the files for 6 months, and you wish to suppress the warning, use the `/NW` command line parameter.

Central Installation and Updates

If you are a system administrator, we know that you do not want to walk around to each individual workstation in your organization and install NVC.

Therefore, you have three choices for installing NVC centrally from a shared location:

1. You can use N_DIST, Norman's multi-platform distribution program
2. You can use Norman Package Distribution, the GUI-based distribution program
3. You can run the setup processes from the shared location, just as you would from the diskette or CD-ROM

Many users are familiar with the script based distribution tool, N_DIST. Customized scripts that need a minimum of editing are generated during an Administrator setup. However, for those who prefer a graphical user interface we are now offering a GUI version of N_DIST: Norman Package Distribution.

Norman Package Distribution is a new distribution program for the NVC products for the Windows platforms. It's based on the Thunderbyte Network Distribution (TBND) facility. Thunderbyte AntiVirus is now being integrated with the Norman products, and a part of this process involves integrating the visual network distribution of the Thunderbyte product.

N_DIST

By running N_DIST from a login script, you can automatically install or update NVC on individual workstations from a central location on the network.

N_DIST is a powerful scripting language that is able to utilize scripts for installing/updating all types of software (not just NVC). You may either use N_DIST with the default NVC distribution scripts as they are provided by Norman or with scripts that you create on your own.

Note: Although N_DIST can read any ASCII script file, we use the extension “`.NXD`” as a convenient way to designate script files that are intended to be used with N_DIST for NVC installations.

The remainder of this chapter is dedicated to a discussion of how to prepare for and implement a central install/update of NVC.

Preparing N_DIST for Customizable Central Installs

As discussed above, N_DIST is designed to be used from a shared location in order to install software onto multiple platforms of workstations (DOS/Windows 3.1x, Windows 95, Windows NT, OS/2...). N_DIST uses a distribution script which is entirely customizable, so you may edit the script in order to meet the needs of your organization.

To use N_DIST for central NVC installs, you must first get the NVC files and N_DIST over to a shared directory by installing the Norman Virus Control Administrator (see below).

If you wish, you may edit the distribution script that is generated during the NVC Administrator install.

Then run N_DIST from a location such as a login script in order to automatically install/update NVC on the workstations in your network as they login.

Norman Virus Control Administrator

Note: If you receive NVC on CD-ROM, you can install directly from this media by choosing **Install on**
[ ] **Network**.

The first step in preparing for a **customizable** central install/update of NVC is to install the Norman Virus Control Administrator. The Administrator is not a single executable but rather a set of files that results after copying over all the Norman programs to a directory of your choice in a central location and generating the distribution script `NVCxx.NXD`, which incorporates your selections for certain options.

The processes for NVC for DOS/Windows 3.1x, Windows 95, and Windows NT are performed in the same manner, but it is a bit different for NVC for OS/2.

Note: Installing the Norman Virus Control Administrator is only possible if you have a **corporate license** for NVC.

Administrator for DOS/Windows 3.1x, Windows 9x, and Windows NT

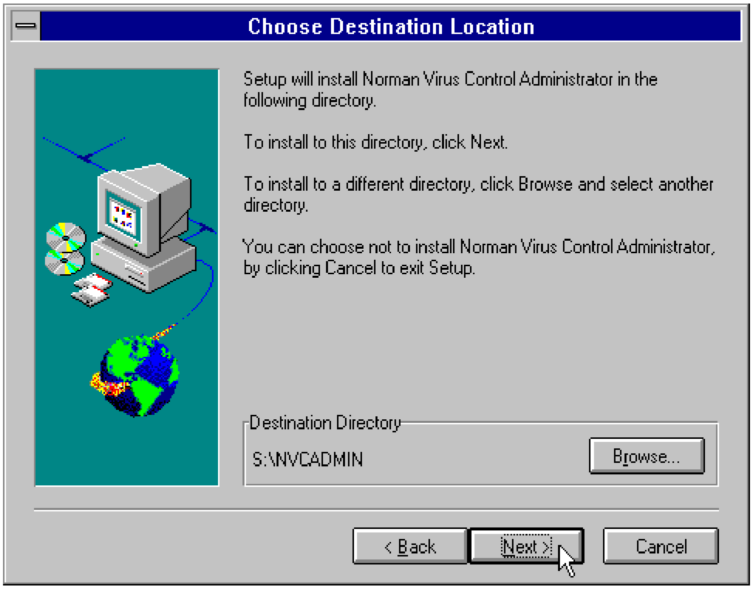
Note: If you receive NVC on a CD-ROM, you must generate the diskettes described below by choosing the option **Make diskettes**.
The CD booklet also provides information on installation as well as an overview on available programs and documentation.

To install the Norman Virus Control Administrator, follow these steps:

1. Start the appropriate operating system for the platform of NVC that you wish to install.
2. From the first NVC distribution diskette, run:

   setup /a [Enter]

The setup program will look for the first available network drive and suggest the `\NVCADMIN` directory on that drive as the directory which will **store** N_DIST and the files that are to be ultimately installed on the workstations:



Choose Destination Location

Setup will install Norman Virus Control Administrator in the following directory.

To install to this directory, click Next.

To install to a different directory, click Browse and select another directory.

You can choose not to install Norman Virus Control Administrator, by clicking Cancel to exit Setup.

Destination Directory

S:\NVCADMIN

Click on the **Browse** button if you wish to change the drive and/or directory.

Confirm your selections in the next screen, and Setup will copy all available NVC modules to the specified directory and the following subdirectories:

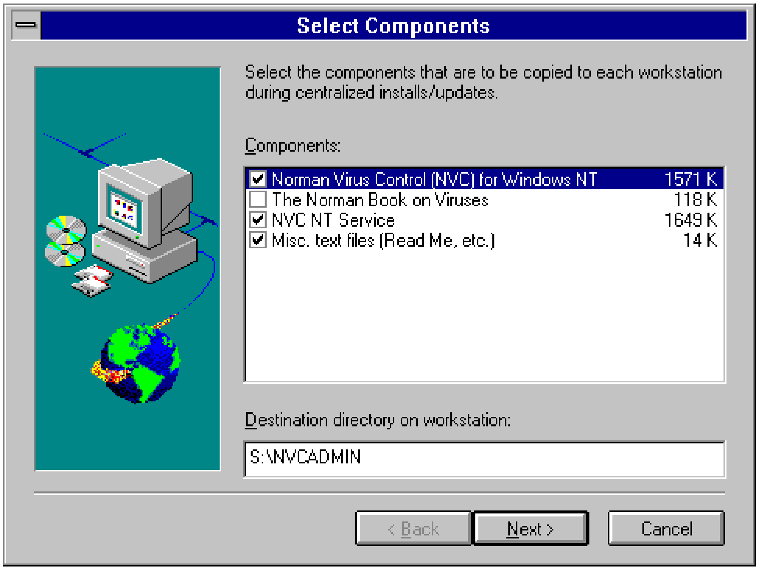
\DOS

\WIN16

\WIN32

\NDIST

The next step is to define which NVC modules you wish to be installed on the **workstations** and in which directory you wish them to be installed.



Select the components that are to be copied to each workstation during centralized installs/updates.

Components:

☑	Norman Virus Control (NVC) for Windows NT	1571 K
☐	The Norman Book on Viruses	118 K
☑	NVC NT Service	1649 K
☑	Misc. text files (Read Me, etc.)	14 K

Destination directory on workstation:

S:\NVCADMIN

Deselect modules by unchecking the relevant checkboxes. Your choices here will be reflected in the distribution script, which is automatically generated during Setup. Each platform of NVC has its own distribution script called `NVCxx.NXD`, where “`xx`” denotes the platform.

The script filenames for the different platforms are as follows:

Filename	Platform
`NVCW.NXD`	DOS/Windows 3.1x
`NVC95.NXD`	Windows 95/98
`NVCNT.NXD`	Windows NT

Note: If you wish to use N_DIST to install more than one NVC platform, you must install the Norman Virus Control Administrator for each platform desired.

By default, `NVCxx.NXD` is placed in the directory `\NVCADMIN\NDIST`, together with these files:

MSCOMSTF.DLL

MSDETSTF.DLL

MSINSSTF.DLL

MSSHLSTF.DLL

MSUILSTF.DLL

NRMWINST.EXE

NRMWLNK.EXE

N_DIST.EXE

N_DIST2.EXE

N_DIST16.EXE

N_DIST32.EXE

3. Ensure that you have a login script or batch file from which N_DIST and its accompanying distribution script, `NVCxx.NXD`, can be run.
4. Ensure that users have read access to the server directories in which the Norman programs now reside.
5. Edit the distribution script as you see fit. The Administrator install only adds critical information to the `.NXD` file. You will see that the `.NXD` file is very powerful, and you may set it up so that it best suits your needs. In addition, you may consider having several copies of the `.NXD` file to accommodate different types of workstations onto which NVC will be installed. In this case, you will have to set up your login scripts or batch files to allow for different `.NXD` files being used by different users.

6. We recommend that you try running N_DIST on a test user or on yourself before rolling NVC out to your entire organization. Remember to make a backup of your login script before beginning.

Please see the sections “Running N_DIST” and “Interpreting and Editing the Distribution Script” on pages 31 and 55.

Updating the NVC NT Service

We have improved the ability to support updating of the NVC NT Service. Many companies want to distribute uniform settings of the NVC NT Service throughout the organization. The NVC NT Service now supports automatic updating via a NT registry (`.reg`) file:

1. Start NCFGW and specify the desired settings on one particuar computer.
2. Start `regedit.exe` and open and highlight the key: HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/NvcSrv/Parameters
3. Within `regedit.exe` now select Registry|Export registry file and save the file `nvcsrv.reg` (mandatory name).
4. Copy this `nvcsrv.reg` file to the `\norman\win32` path on each NT machine.
5. The next time the NVC Service is started, it will use the settings from this `.reg` file.

Note: This will only work with the NVC NT Service, not with the GUI component (NVCNT.EXE).

Administrator for OS/2

This is performed differently in OS/2 than in the Windows-based operating systems:

1. From the first NVC for OS/2 diskette, run `INSTALL.EXE` and choose a shared directory as the destination location instead of `C:\NORMAN`.
2. From the last NVC for OS/2, manually copy `N_DIST2.EXE` and `NVCOS2.NXD` to the shared directory.
3. Edit `NVCOS2.NXD` to suit your needs. The only lines that should be edited are:

```
// Set source directory on the file server
set $Source='J:\NvcFiles'


// Set target directory on the workstation
set $Target='F:\NORMAN'


// Set path to the OS2 install directory on
// the target workstation:
set $Os2Path='C:\OS2'


// Select components to transfer
set $Inst_Os2PM= 'Yes'
set $Inst_Os2VirBook = 'Yes'
set $Inst_Os2CmdLine = 'Yes'
set $Inst_ReadMe     = 'Yes'
```

4. Make a backup of the `.CMD` file you are using for logging users into the network.
5. Run the command `N_DIST2.EXE NVCOS2.NXD` from the `.CMD` file that you are using for logging users into the network.

Refer to “Running N_DIST” on page 31 and “Interpreting and Editing the Distribution Script” on page 55 for more details.

Running N_DIST

As mentioned above, N_DIST is a scripting language, and therefore, it can be used for functions other than installing/ updating NVC.

The syntax for using N_DIST in DOS/Windows 3.1x, Windows 95/98, and Windows NT environments is:

```
n_dist <script filename> </option(s)>
[Enter]
```

You are not limited to using the default distribution scripts that are provided in the Norman Virus Control Administrator, nor are you limited to saving your scripts with the .NXD extension.

In order to use N_DIST in DOS/Windows 3.1x, some supporting files must reside in the same directory as N_DIST.EXE:

```
N_DIST16.EXE

NRMWINST.EXE

NRMWLNK.EXE
```

In order to use N_DIST in Windows 95/98 and Windows NT, some supporting files must reside in the same directory as N_DIST.EXE:

```
N_DIST32.EXE

MSCOMSTF.DLL

MSDETSTF.DLL

MSINSSTF.DLL

MSSHLSTF.DLL

MSUILSTF.DLL
```

Note: N_DIST and its accompanying support files should all reside in the same directory.

The syntax for using N_DIST in an OS/2 environment is:

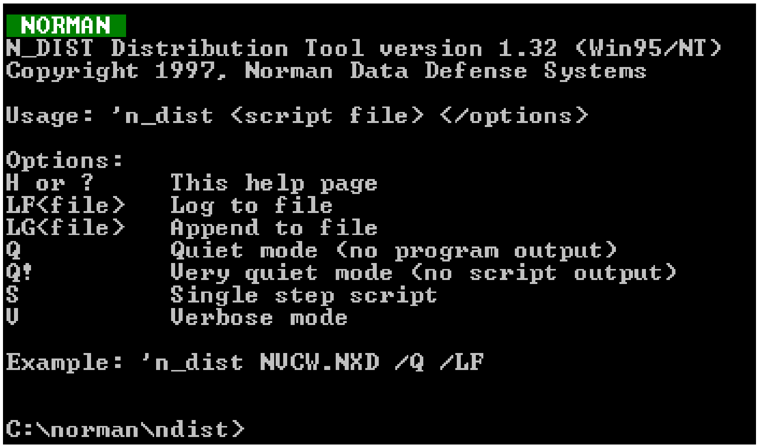
```
n_dist2 <script filename> </option(s)>
[Enter]
```

`N_DIST2.EXE` does not require any additional supporting modules other than the script file.

There are a handful of options that are available from the command line. For an overview, type:

```
n_dist [Enter]
```

and you will see:



```
 NORMAN
N_DIST Distribution Tool version 1.32 (Win95/NT)
Copyright 1997, Norman Data Defense Systems

Usage: 'n_dist <script file> </options>

Options:
H or ?      This help page
LF<file>    Log to file
LG<file>    Append to file
Q           Quiet mode (no program output)
Q!          Very quiet mode (no script output)
S           Single step script
V           Verbose mode

Example: 'n_dist NVCW.NXD /Q /LF


C:\norman\ndist>
```

Note: The functionality performed by all these options is available from within the distribution script as well. See “Options Used in the Distribution Script” on page 71 for more information.

The Service Module N_Serv

Normally, N_Dist is invoked by running n`_dist.exe` or `n_dist2.exe`. A special case exists when N_Dist is running on a NT workstation. Unless the user has administrator rights, some of the commands in the script may not be executed in a flawless manner.

This is especially the case when updating device drivers or manipulating certain parts of the registry. The N_Serv utility is a remedy for such problems. It is an installable service that will invoke N_Dist with its own rights. If the N_Serv service is installed with administrator rights, any N_Dist script may also be run with these rights.

To install the service, run the following command:

```
N_Serv -install
```

You will be prompted for an administrator password. When you have typed in the password, the service is installed and will be started the next time the workstation is rebooted. The service may also be started immediately by typing:

```
N_Serv -start
```

provided that the logged on user has sufficient privileges.

To remove the service, type:

```
N_Serv -remove
```

`N_Serv -?` will display some basic help.

When the service is running, a N_Dist script may be run by typing:

```
N_Serv {[drive]:\nvcadmin\ndist\} script.nxd
```

The usual N_Dist options are available, simply replace ’N_Dist’ with ’N_Serv’ in the login script. N_Serv will then invoke N_Dist32.exe with the given script name and options.

To ease the pain when installing N_Serv with administrator rights in large networks, an extra password feature has been implemented. This feature allows the installation of the product without physically typing in the administrator password over and over again for each user.

The option used is ’-p:’. When installing N_Serv, the command line is hence:

```
N_Serv -install -p:<password>
```

The password is an encrypted, readable version of the administrator password. It is safe to distribute the password through login scripts and the like. To generate the encrypted password to be used for direct installations, use the ’-p:’ option without any other options.

Example:

The common administrator password in a network is ’g7Kp’. To generate a distributable password, run:

```
N_Serv -p:g7Kp
```

Note the case sensitivity. N_Serv will reply with the password ’bfcbgibl’. Use this new password to install the N_Serv service across the network using the login script or other utilities. The command line to enter on each NT workstationis:

```
N_Serv -install -p:bfcbgibl
```

IMPORTANT: Using N_Serv will create an opportunity for a user to write a N_Dist script and execute it with administrator rights. Network administrators should evaluate any security risks before using N_Serv.

By default, the N_Serv service will try to log on to the ’Administrator’ account . It is possible to specify another account, either by using the Services applet in the control panel, or by using the '-a:' option together with '-install'. This is relevant when, for example, the default administrator account is unable to log on to the network. To solve this problem, create a user on the network and the workstations that also has administrator rights, and use this user name and password when you install N_Serv.

The N_Serv executable file (N_Serv.exe) must reside on the system where the service is to be installed.

Example:

How to use N_Serv to distribute NVC for Windows NT in a network environment:

1. Install the n_serv service on all workstations. You can do this by running the following commands from a loginscript:

```
copy x:\nvcadmin\ndist\n_serv.exe c:\norman\win32
c:\norman\win32\n_serv -install -p:****** -a:****
c:\norman\win32\n_serv -start
```

where ’-p:******’ is the encrypted password and ’-a:****’ is the user ID for a user with local admin rights and read access to the x:\nvcadmin directory on the server.

2. Run n_serv to install NVC for Windows NT. This may be done by running the following command from a login script:

```
x:\nvcadmin\ndist\n_serv x:\nvcadmin\ndist\nvcnt.nxd
```

The user who is installing N_Serv, either directly or through a login script, does not need administrator privileges to do the service installation, as long as the credentials given by the account name and password has the privileges 'Act as part of the operating system', ’Log in as service', and 'Bypass traverse checking' set.

'Act as part of the operating system' and 'Bypass traverse checking' are powerful privileges which should be disabled for all users immediately after N_Serv has been installed in a network. It is regarded security malpractice to let a process log on to a workstation using a new set of credentials the way it is done when installing N_Serv.

Summary of User Rights for N_Serv

The user that is logging on to a workstation and thus running 'N_Serv -install' to install the N_Dist service does not need administrator rights. However, the user needs the following rights:

1. 'Act as part of the operating system'
2. 'Log on as service'
3. 'Bypass traverse checking'

The first right is off by default, and must be switched back off after installing N_Serv. The other user right is usually on. Remember that ’Bypass traverse checking’ needs to be "on" for administrator too.

The user that N_Serv is logging on as, does not need any special rights. However, N_Serv, and hence N_Dist, will execute with the rights of this user. Since the purpose of the N_Dist service is to allow N_Dist to run with administrator rights, a user with such rights would be desirable.

Sample Installation

1. Create User "NSERV" on network with minimal rights (needs only ’read’ rights in [drive]:\NVCADMIN’).
2. Create User "NSERV" on workstations with ’USER’ rights and add ’Act as part of the operating system’, ’Logg on as service’ and ’Bypass traverse checking’ rights.
3. Copy N_SERV.EXE to the workstations (C:\NORMAN\WIN32)
4. Run once on workstations:
   C:\N_SERV.EXE -install -p:<encrypted password> -a:NSERV
5. Remove the ’Act as part of the operating system’ and ’Bypass traverse checking’ rights on user "NSERV" on workstations.
6. Add N_serv commandline in loginscript to run NXD script. Example:

   ```
   c:\norman\win32\n_serv.exe
   [drive]:\nvcadmin\ndist\nvcnt.nxd
   ```

Note: The user *must* exist on the local PC, on the Domain and on the local server in order to have network access. Otherwise N_SERV won’t be able to access the network drive to execute the NXD script.

N_DIST and NVC Platforms

During the installation of Norman Virus Control Administrator (see page 25), default distribution scripts for use in central NVC installs/updates are generated. You will also notice that in the shared directory, there are several executables that bear the N_DIST name: `N_DIST.EXE`, `N_DIST16.EXE`, `N_DIST32.EXE`, and `N_DIST2.EXE`.

`N_DIST.EXE` will spawn `N_DIST16.EXE` or `N_DIST32.EXE`, depending on which operating system is found. Therefore, in DOS/Windows 3.1x, Windows 95/98, and Windows NT environments, it is only necessary to run `N_DIST.EXE` along with the appropriate script.

However, in OS/2, it is necessary to run `N_DIST2.EXE` along with the appropriate script.

Following is an overview of how the N_DIST executables should be run on different platforms for central installation/ update of NVC:

Platform	**Syntax**
DOS/Windows 3.1x	N_DIST NVCW.NXD
Windows 9x	N_DIST NVC95.NXD
Windows NT	N_DIST NVCNT.NXD
OS/2	N_DIST2 NVCOS2.NXD

Note: Although it is necessary to run `N_DIST2.EXE` in OS/2 instead of `N_DIST.EXE`, you may design a script so that both `N_DIST.EXE` and `N_DIST2.EXE` can use it in multi-platform environments.

Using N_DIST in NetWare

In NetWare, we suggest that the system login script run N_DIST. We also recommend that you run the N_DIST script with the option `/Q` to suppress messages. See “Options Used in the Distribution Script” on page 71 for an overview on available options.

Example:

```
If "%LOGIN_NAME" <> "SUPERVISOR"
```

Note: The following example must be entered on *one* line.

```
#S:\NVCADMIN\NDIST\N_DIST.EXE
S:\NVCADMIN\NDIST\NVCW.NXD /Q

...
```

In this example all users except for the Supervisor will have NVC installed when they login.

Using N_DIST in LAN Server

In LAN Server you can start N_DIST from the `.CMD` file you are using for logging users into the network. We also recommend that you run the N_DIST script with the option `/Q` to suppress messages.

Example:

```
;login first...
```

Note: The following example must be entered on *one* line.

```
S:\NVCADMIN\NDIST\N_DIST2.EXE
S:\NVCADMIN\NDIST\NVCOS2.NXD /Q

...
```

Using N_DIST in NT Server

In NT Server, you can add a command similar to this to the logon scripts. We also recommend that you run the N_DIST script with the option `/Q` to suppress messages.

Example:

```
;login first...
```

Note: The following example must be entered on *one* line.

```
S:\NVCADMIN\NDIST\N_DIST.EXE
S:\NVCADMIN\NDIST\NVCNT.NXD /Q
...
```

Alternatively, you can insert a line like the above in the Startup group.

Running N_DIST in a Mixed Environment

If your organization is like most, all workstations on your network do not necessarily run the same operating system.

In this case, you must run an administrator install for each platform, and three separate distribution scripts will be generated: `nvcw.nxd`, `nvc95.nxd`, and `nvcnt.nxd`.

One of the lines in the distribution script created during the NVC Administrator process tests for operating system. If the operating system found does not match that of the distribution script, all commands are skipped and the next login script command is run.

In order to ensure that N_DIST is run for all available operating systems, you may add the following lines to your login script.

```
N_DIST [path] NVCW.NXD

N_DIST [path] NVCNT.NXD

N_DIST [path] NVC95.NXD

N_DIST2 [path] NVCOS2.NXD
```

Note: Although it is necessary to run `N_DIST2.EXE` in OS/2 instead of `N_DIST.EXE`, you may design a script so that both `N_DIST.EXE` and `N_DIST2.EXE` can use it in multi-platform environments.

Resulting Directory Structure on the Workstations

During a custom installation from the NVC distribution diskettes/CD-ROM or via N_DIST, it is possible to select a workstation installation directory other than the default directory `C:\NORMAN`.

However, when NVC is installed on the workstation (either from the NVC distribution diskettes or via N_DIST), a certain subdirectory structure underneath `C:\NORMAN` (or whatever you have chosen) is created.

Common files are installed in the `C:\NORMAN` directory (or whichever directory you specified):

`NVC32.CFG`

`README.TXT`

Files for the scanning engine (definition files and DLLs/ VxDs) and certified macros are installed in the `C:\NORMAN\NSE` directory.

Platform-dependent modules are installed in subdirectories of `C:\NORMAN` (or whichever directory you specified) as follows:

`\DOS`

`\WIN16`

`\WIN95`

`\WIN32`

`\OS2`

Note: This subdirectory structure cannot be changed.

Norman Package Distribution

Users of Thunderbyte AntiVirus are probably familiar with this tool (previously called Thunderbyte Network Distribution). However, certain changes have been implemented. As an aid for these users, significant changes to the previous version will be specified like this whenever they occur:

Note for TBND users: The corresponding function in TBND is...

About Norman Package Distribution

Like N_DIST, Norman Package Distribution (NPD) is a system that facilitates automatic installation of Norman software on workstations across the network the first time as well as installing subsequent updates. It relies on:

- Features for silent installation of the setup programs of the Norman software. By "silent" we mean installations without user prompts.
- The login-script mechanism of the network file server ensures that each workstation is updated with the most recent virus protection at login time, and with as little manual effort as possible.

Prerequisites

You need a good understanding of the configuration of the file servers, especially with respect to login-script features and network access rights.

Technical Description

Norman Package Distribution handles the distribution of first-time installs as well as updates to the Norman workstation software by relying on:

1. The login script features of the workstation’s file server.
2. The silent setup feature of the NVC installation procedure, present from version 4.60.

This is the basic outline:

1. Add packages of the latest version of NVC to a public directory on the file server which is accessible to each network user.

2. The login script is extended with a call to the NRNDDET, a detection utility integrated in Norman Package Distribution, that detects the OS version of the workstation and checks if Norman software is already installed. If Norman software is not installed or if the installed version is older than the current version, the SETUP program is started in silent mode.

System Requirements

1. NPD will not work with NVC versions prior to v4.60.
2. Only networks that allow logon from Windows and that support some form of login script are supported. This includes Novell NetWare and Microsoft Windows Network if configured properly.
3. Windows 3.x workstations where the login script is executed from DOS (NETX) are also supported.

Installing Norman Package Distribution

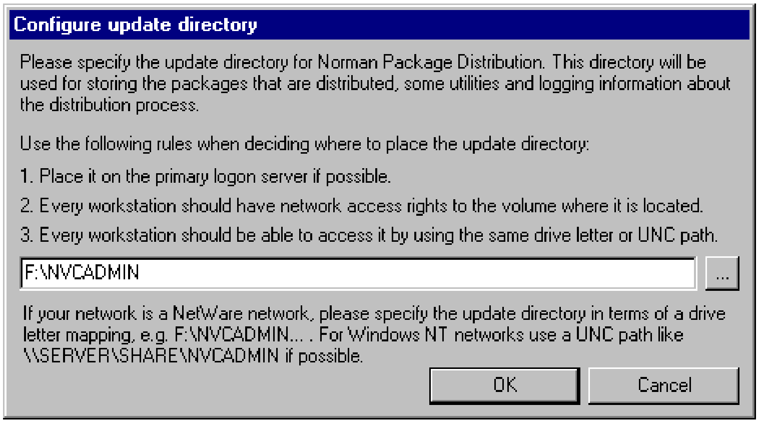
1. Make sure that you are logged in as Administrator.
2. Run setup from the NVC floppies/CD-ROM to install the NPD GUI on a Windows 9x or Windows NT machine. By default, the product will be installed in the directory `c:\Program Files\Norman Package Distribution`. Note that this step will not make any attempt to configure your file server.
3. During the installation, you’ll be prompted to allow a change in the login script. Click on **OK** to accept an entry in the login script allowing automatic installation of NVC to the workstations.

   For NetWare4.x and 5 you should use the NWADMIN or NWNET utilitities.
4. The setup program will also create a shortcut for starting the NPD user interface on the Windows desktop (or Program Manager for Windows NT 3.51).

Running Norman Package Distribution

Use the NPD user interface (its icon is installed in the Start menu) to configure, monitor, and manage a centralized installation of the NVC products. Click on the icon to start NPD.

The first time it is started, you will be prompted to specify an Update directory. The default location is `F:\NVCADMIN`.



Configure update directory

Please specify the update directory for Norman Package Distribution. This directory will be used for storing the packages that are distributed, some utilities and logging information about the distribution process.

Use the following rules when deciding where to place the update directory:

1. Place it on the primary logon server if possible.
2. Every workstation should have network access rights to the volume where it is located.
3. Every workstation should be able to access it by using the same drive letter or UNC path.

F:\NVCADMIN

If your network is a NetWare network, please specify the update directory in terms of a drive letter mapping, e.g. F:\NVCADMIN... . For Windows NT networks use a UNC path like \\SERVER\SHARE\NVCADMIN if possible.

Note for TBND users: The default directory for TBND is `F:\TBUPDATE`.

1. Make sure that you observe the rules for access rights and special network considerations specified in this dialog.
2. By default, you are offered F:\NVCADMIN, which should be fit Novell NetWare networks where F: is mapped to a volume on the file server. The actual position of this directory does not matter as long as any network user is able to access it by using the same specification.

   The following subdirectories to NVCADMIN are also created:

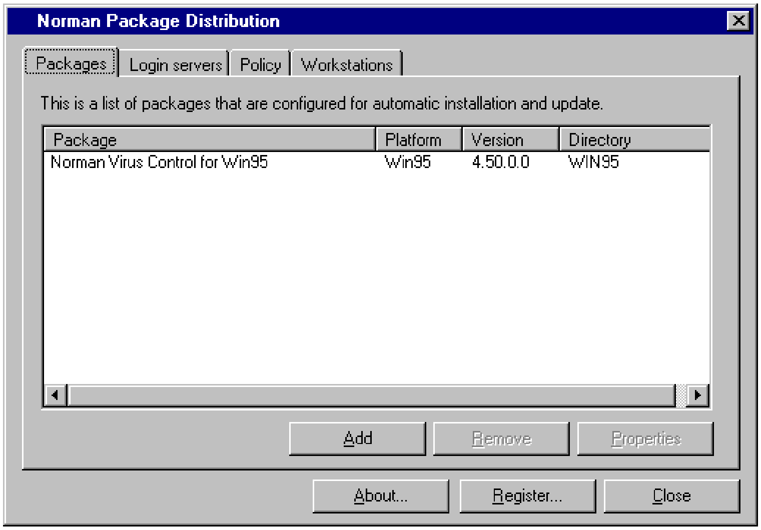
LOGS: The directory where Norman Package Distribution logs information about its activity for each workstation.

SERVERS: The directory where Norman Package Distribution stores information about logon servers on the network.

3. Make sure that every network user has ***read*** rights to the NVCADMIN directory and ***read***, ***write*** and ***create*** rights to the F:\NVCADMIN\LOGS directory.
4. Select **OK** to create the directories and copy the necessary files.

The Main Window

The Norman Package Distribution's main window is made up of four tabbed dialog boxes:



This is a list of packages that are configured for automatic installation and update.

Package	Platform	Version	Directory
Norman Virus Control for Win95	Win95	4.50.0.0	WIN95

From here, you can add NVC versions for distribution to the workstations, select servers, administer and monitor the package distribution on the network.

Packages

A package is a complete version of NVC for a specific platform. There are three available packages for NPD:

- NVC for DOS/Windows 3.1x
- NVC for Windows NT
- NVC for Windows 95

To install a package:

1. Click the **Add** button.
2. From the ’Add new package’ dialog, select package by clicking the ’...’ button and choose NVC version from the CD-ROM or floppy. The program will look for .ASU files. If you want NVC for Windows 95, for example, then select `nvc95.asu`.
3. When you have selected a package, click on **OK** to confirm.
4. During the installation of the packages, you can choose which modules of NVC you want to install. Refer to the appropriate *User’s Guide* for detailed information on the modules. Click on the **Next** button when you have made your choices.
5. The package will now appear in the ’Packages’ dialog.
6. Repeat step 1 - 4 for each package you want to install.

Each package will be installed to a new subdirectory to NVCADMIN. NSE packages will not appear in the list box, because this is an integrated part of the package. NSE holds all files used by the scanning engine, like virus definition files, DLLs and VxDs.

To update the network package, follow the procedure described above and select the .ASU package file with the new version.

From the tabbed dialog ’Packages’ you can delete packages by clicking on **Remove**. Note that removal of a package does not include removal of the directory

F:\NVCADMIN\NSE, where the scanning engine is located.

Click on the **Properties** button to configure the installation settings for a highlighted package.

Note for TBND users: The **Properties** button will not pop-up a properties window for a NVC package.

Login Servers

The login script that is used network wide when each user logs on needs to be updated to initiate the detection utility, NRNDDET.EXE. This tool detects which platform the workstation is running on (Windows 3.1x, Windows 9x, or Windows NT) and subsequently starts NRND16.EXE or NRND32.EXE.

Note for TBND users: The corresponding file names for TBND were: TBNDDET.EXE, TBND16.EXE, and TBND32.EXE.

The user interface provides help for the following items in the login script:

- Modifies the login script of a NetWare 3.x server.
- Creates/modifies the login script of a Windows NT server.

Modification/creation of the login script on NetWare 4.x servers and higher is not done automatically. The program describes the process how to update the script necessary for the update procedure.

For Windows NT servers a login script will be created in the Repl\Import directory of the server, but it will not be activated for the users. The admin has to go to each network user in the UserManager and turn the login script on.

You can modify the login script by clicking **Add** from the tabbed dialog 'Login servers'. The 'Add new login server' window shows a list of server types.

Note: Depending on your network protocol stack configuration this list may sometimes not show all servers present on the network. If this is the case, select the proper server type and type in its name.

Policy

Use the tabbed dialog 'Policy' to establish the location of update directories, workstation log files, and security options.

1. The default option for the update directory for network distribution is the directory you specified in the dialog 'Configure update directory', for example F:\NVCADMIN. Click on **Configure** to change location.
2. When NVC is distributed to the workstations in a network, a log file is created for each workstation. Please refer to the next section for information on log file content. The default location of the log files is identical to what you specified in the 'Configure update directory', for example F:\NVCADMIN\LOGS. You can change the location by entering a different location in the list box.
3. If a first time NVC install or a NVC update fails on a workstation, the default action is to block the workstation from the network. You can uncheck the option **[x] Deny logon to network** if you don't want this action.
4. Finally, you can assign administrator's privileges on workstations for certain users by adding username(s) in the list box at the bottom of this display. Identify users by entering their user names, separated by ';' (semicolons).

Workstations

The tabbed dialog ’Workstations’ is made up of two list boxes: workstation log files and locked workstations.

1. You can highlight any log file in the list and click on the **View** button. Note that workstations without read-write and create rights to \NVCADMIN\LOGS, are unable to generate log files to this location. The log files contain information of the network distribution activity. The name is created from the name of the machine, and the file extension is `.log`. Examples of log file information is the time when a package will be installed, setup status, NVC version and .ASU file installed, if reboot is required, and when the machine was rebooted. Log information can be very useful for troubleshooting.
2. After an unsuccessful NVC installation/update, the machine will be locked off from the network by NPD by means of a lock file created in the Logs directory. The name of this file is built up from the username, and the file extension is `.lck`.
   If you selected the option **[x] Deny logon to network**, the names of the workstations which failed to install NVC will appear in the ’Workstations that are locked’ list box. You can examine the log files for these workstations, and manually move a workstation online by highlighting the workstation name and click on the **Allow to log on** button.

When the User Logs In

Installation and updating is almost invisible to the workstation user. A NPD dialog will appear for a few seconds. The information from the distribution activity is stored in the log file for each user.

Note for TBND users: The 32bits version of TBND agent used to run in a console window (DOS-like). NPD runs in a GUI window.

Platforms Not Supported

This version of NPD does not support DOS and OS/2. Please refer to the N_DIST section on the following page for distribution on these platforms.

Distributing Cat's Claw

If you have certified macros for use in your organization, or configuration settings to be used by all users, then there are some topics that should be considered if Cat's Claw should be updated automatically at login time.

When you configure Cat's Claw, the following files are generated:

1. `nvcmacro.crt`, if you certify macros.
2. All other configuration settings in `claw31.ini` or `claw95.reg`, for Windows 3.1x and Windows 95/98, respectively.

Make sure that these files resides on the server in their respective folders (default directories):

- `nvcmacro.crt` in \`nvcadmin` where the definition files reside
- `claw31.ini` in `\nvcadmin\win16`
- `claw95.reg` in `\nvcadmin\win95`

If you're using default N_DIST distribution, Cat's Claw will be updated with new files whenever changes are done to the files on the server.

Note: If your configuration involves checking of uncertified macros, the file `nvcmacro.crt` must exist. If not, Cat's Claw will not start and an error message appears.

If you’re using another distribution vehicle than N_DIST and want to update automatically at login, after new files are copied to the workstations, run this command:

`claw31.exe /install`, or

`claw95.exe /install`

Running the Installer from the Network

Note: If you receive NVC on a CD-ROM, you must generate the diskettes described below by choosing the option **Make diskettes**.
The CD booklet also provides information on installation as well as an overview on available programs and documentation.

When Norman ships NVC for DOS/Windows 3.1x, Windows 95, and Windows NT diskettes, you are instructed to run `SETUP.EXE` (a Windows-based program) from the first disk in order to install NVC on your workstation.

Similarly, you run `INSTALL.EXE` (an PM-based program) from the distribution diskettes in order to install NVC for OS/2 on your workstation.

If you wish, you may also copy the original NVC distribution files to a shared directory and run `SETUP.EXE`/`INSTALL.EXE` from there in order to install NVC on a workstation without requiring the use of diskettes.

The following two sections discuss these methods in detail.

See the section “Interpreting and Editing the Distribution Script” on page 55 if you wish to use an alternative method.

SETUP.EXE for DOS/Win 3.1x, Windows 95, and Windows NT

In order to run `SETUP.EXE` from a shared directory, follow these steps:

1. For each NVC distribution diskette, make a corresponding directory on the network with the names "`DISK1`", "`DISK2`", and so on.

   For example, let's assume that you have a directory called `\NVCFILES`, you wish to install NVC for DOS/ Windows 3.1x, and it comes on 3 diskettes.Then you should make a directory structure as follows:

   ```
   \NVCFILES
       \DISK1
       \DISK2
       \DISK3
   ```

2. Manually copy all files from each NVC diskette into the appropriate `\DISKx` directory. That is, from the NVC diskette 1, copy all files to the `\NVCFILES\DISK1` directory; then from the NVC diskette 2, copy all files to the `\NVCFILES\DISK2` directory, etc.
3. From the workstation on which you wish to install NVC, start the appropriate operating system.
4. Run `SETUP.EXE` from the `\DISK1` subdirectory.

   This will result in exactly the same type of installation as if you had run `SETUP.EXE` from the actual NVC distribution diskettes. That is, the installation process will require user input as it goes along, and it will assume certain default settings.

If you wish to customize an installation for more than one workstation on the network, then we recommend using N_DIST.

INSTALL.EXE for OS/2

In order to run `INSTALL.EXE` from a shared directory, do the following:

1. Make a directory on the network and manually copy all files from the NVC for OS/2 diskettes into it.

   For example:

   ```
   MKDIR J:\NVCOS2
   XCOPY A:*.* J:\NVCOS2
   ```

2. From the workstation on which you wish to install NVC, start OS/2.
3. Run `INSTALL.EXE` from the `J:\NVCOS2` directory.

   This will result in exactly the same type of installation as if you had run `INSTALL.EXE` from the actual NVC distribution diskettes. That is, the installation process will require user input as it goes along, and it will assume certain default settings.

   Please see the following sections for information on performing unattended installs, updates, and uninstalls using `INSTALL.EXE`.

If you wish to customize an installation for more than one workstation on the network, then we recommend using N_DIST. See "Interpreting and Editing the Distribution Script" on page 55 for more information.

Performing an Unattended INSTALL.EXE with Default Settings

The following will result in an unattended installation process that installs NVC for OS/2 into the `C:\NORMAN` directory.

1. Make a directory on the network and manually copy all files from the NVC for OS/2 diskettes into it.

   For example:

```
MKDIR J:\NVCOS2

XCOPY A:*.* J:\NVCOS2
```

2. From the workstation on which you wish to install NVC, start OS/2.
3. Run the following as a command on **one** long line:

   ```
   INSTALL /a:i /X /R:J:\Nvc-
   Files\NVCOS2.RSP /l1:X:\LOG-
   PATH\ERRORS.LOG /
   l2:X:\LOGPATH\HISTORY.LOG
   ```

   Note that the path for `ERRORS.LOG` and `HISTORY.LOG` will be created if they don't exist. Inspect the `ERRORS.LOG` file for errors and `HISTORY.LOG` for information about files transferred, etc.

   For additional command line parameters, run `INSTALL /?`.

Performing an Unattended INSTALL.EXE with Custom Settings

The following will result in an unattended installation process that installs NVC for OS/2 into a directory of your choice. You may also use this method in order to update or delete only a few components of NVC for OS/2.

1. Make a directory on the network and manually copy all files from the NVC for OS/2 diskettes into it.

   For example:

   ```
   MKDIR J:\NVCOS2

   XCOPY A:*.* J:\NVCOS2
   ```

2. Edit the file `NVCOS2.RSP` by finding the line with the keyword "`FILE`" and changing "`C:\NORMAN`" to the desired installation target directory. See the comments in `NVCOS2.RSP` for more information.

3. Deactivate the components that you do not want INSTALL.EXE to handle. Do this by putting an asterisk in the first column of the COMP statements in question.
4. From the workstation on which you wish to install NVC, start OS/2.
5. Run the following as a command on **one** long line:

   ```
   INSTALL /a:i /X /R:J:\Nvc-
   Files\NVCOS2.RSP /l1:X:\LOG-
   PATH\ERRORS.LOG /
   l2:X:\LOGPATH\HISTORY.LOG
   ```

   Note that the path for `ERRORS.LOG` and `HISTORY.LOG` will be created if they don’t exist. Inspect the `ERRORS.LOG` file for errors and `HISTORY.LOG` for information about files transferred, etc.

   For additional command line parameters, run `INSTALL /?`.

Performing an Unattended Update or Delete

To update an existing NVC for OS/2 installation, follow these steps:

1. Make a directory on the network and manually copy all files from the NVC for OS/2 diskettes into it.

   For example:

   ```
   MKDIR J:\NVCOS2
   ```

   ```
   XCOPY A:*.* J:\NVCOS2
   ```

2. From the workstation on which you wish to install NVC, start OS/2.
3. Run the following as a command on **one** long line:

   ```
   INSTALL /a:u /X /R:J:\Nvc-
   Files\NVCOS2.RSP /l1:X:\LOG-
   PATH\ERRORS.LOG /
   ```

```
12:X:\LOGPATH\HISTORY.LOG
```

Note that the path for `ERRORS.LOG` and `HISTORY.LOG` will be created if they don't exist. Inspect the `ERRORS.LOG` file for errors and `HISTORY.LOG` for information about files transferred, etc.

To delete NVC for OS/2 files, use the parameter `/a:d` instead of `/a:u`.

For additional command line parameters, run `INSTALL /?`.

Interpreting and Editing the Distribution Script

You need not create distribution scripts for NVC yourself, for these are generated automatically during the NVC Administrator install. However, if you choose to edit a default script or create one of your own, you should be aware that N_DIST is an extremely powerful tool, and you should therefore test all your scripts thoroughly before deploying them.

If you choose to edit a distribution script, you should have a basic understanding of how batch files work.

Commands Used in the Distribution Script

There are some rules to follow and other items to note about N_DIST distribution scripts:

1. The script itself must be an ASCII file.
2. Only the lines between the statements "`NXD BEGIN`" and "`NXD END`" will be interpreted.

Note: The statements "`NXD BEGIN`" and "`NXD END`" must be in uppercase letters.

3. Any line that begins with "`;`", "`//`" or "`/*`" is interpreted as a comment.

4. All string arguments **must** be enclosed in single quotes.
5. When directory names are given as arguments, it is not necessary to enclose them in single quotes unless they contain spaces.
6. Commands, variable names, etc. are case insensitive unless otherwise specified.
7. A variable's name can be up to 255 characters. If you include more than one variable, this means that the total number of characters in these variables cannot exceed 255.
8. Extra spaces between arguments are not interpreted.
9. When a "`=`" is used in the command, there are no requirements on spaces before or after the "`=`" sign.
10. When a "`:`" is used in the command, there are no requirements on spaces before or after the "`:`".
11. N_DIST will not override original attribute settings on files or directories, and it will not reset or change access rights.
12. In some instances, it is necessary to use a section name (i.e., `[COMMON]` within `CONFIG.SYS`) as an argument. In this case, the section name **must** be enclosed in square brackets like these `[ ]`.
13. The position of the arguments following N_DIST commands is significant.

Note: In the standard `.NXD` files that Norman provides, there is a provision for making backups of `CONFIG.SYS`, `AUTOEXEC.BAT`, and `WIN.INI` if these files are to be modified. These backups will have the extension `.NVC`, and if the file is to be modified again, while a `.NVC` file already exists, the existing `.NVC` file will be overwritten.

In our descriptions of all N_DIST commands below, we will use the following conventions:

- Each command is listed in bold and is followed by its arguments.

- Mandatory arguments are not displayed with any characters surrounding them, except for single quotes `‘ ’` where necessary.
- If there is a choice between two or more arguments that must be used, the choices will be displayed within brackets like these `< >`, and the choices themselves will be separated with the “`|`” character.
- If there is an optional argument, it will be displayed within curly brackets like these `{ }`.
- If an argument is presented as `{<argument 1 | argument 2>}`, this means that neither are required, but if you wish to choose one, then only one can be used.

We will discuss the commands alphabetically, but where one command is related to another, we will group them together regardless of alphabetization.

BEEP `{<x |’file.wav’>}`

Choose between emitting a default beep, a beep number (between 1 and 5) or a `.WAV` file (only for Windows 95/98 and Windows NT). If you choose to use a `.WAV` file, then you should specify the full path.

Example:

```
BEEP 2
```

This emits the second of five optional beeps.

See also the `Nosound` option in the discussion that begins on page 71.

CLEARREGISTRY, GETREGISTRY, and SETREGISTRY

CLEARREGISTRY `’key’ ‘parameter’`

Removes a parameter from the Registry (in Windows 95/98 and Windows NT only).

Example:

```
CLEARREGISTRY ‘HKEY_CURRENT_USER\Environment’ ‘TEMP'
```

This clears the “`TEMP`” parameter from the key “`HKEY_CURRENT_USER\Environment`”.

Warning: Use this command with extreme caution. Clearing entries in the Registry may render the system inoperable.

GETREGISTRY `$variable ‘key’ ‘parameter’`

Extracts the value of a parameter from the Registry (in Windows 95/98 and Windows NT only) and returns it in a variable.

Example:

```
GETREGISTRY $Result ‘HKEY_CURRENT_USER\Environment’ ‘TEMP’
```

This would return the value of “`TEMP`” in the variable called `$Result`.

SETREGISTRY `‛key’ ‘parameter’ ‘value’`

Sets or replaces a parameter in the Registry (in Windows 95/98 and Windows NT only). Only parameters of the type “*string”* can be created or changed.

Example:

```
SETREGISTRY ‘HKEY_CURRENT_USER\Environment’ ‘TEMP’
‘%SystemDrive%\TEMP’
```

(Must be entered on **one** line.) This would set the value of “`TEMP`” to “`%SystemDrive%\TEMP`”.

Warning: Use this command with extreme caution. Invalid entries may render the system inoperable.

COPY `source target <always | update | deferred>`

Copies files from source to target, using the modes “always”, “update”, or “deferred”. Wildcards are allowed.

“Always” will perform the copy every time N_DIST is executed.

Example:

```
COPY c:\*.exe d: always
```

This command always copies the *.EXE files from the root of C: to the D: drive.

“Update” checks the date and time of the source and target files. If the target file exists, and it is older than the source file, the copy operation is performed.

“Deferred” is for the Win32 environments and is used together with the mode switches 'update' or 'always'. When 'deferred' is used, the copy operation runs as normal. However, if the target is unavailable (locked), the copy operation will fail and be deferred until the next system boot.

Example:

```
COPY ’c:\winnt\newfile.dll’
’c:\winnt\oldfile.dll’ update deferred
```

CUTWORD, INSERT, and REPLACE

CUTWORD $variable ‘keyword’

Use this command in combination with other commands to remove a word or string from the line given in $variable. The delimiter for the word or string is ‘space’. This command is useful for editing lines like “LOAD =“ in win.ini. The resulting line will be stored in `$variable`. Therefore, during the course of using this command, `$variable` will have two different values.

Example:

```
SEARCH $Result C:\AUTOEXEC.BAT ‘PATH=‘

CUTWORD $Result ‘C:\NORMAN’

INSERT C:\AUTOEXEC.BAT beginning
$Result
```

This would find the full path statement in `AUTOEXEC.BAT`, cut `C:\NORMAN` from that path and then insert `C:\NORMAN` into the beginning of the path statement in `AUTOEXEC.BAT`.

INSERT `file {[section]} <beginning | end> {‘keyword’} ‘string’ {nodup}`

Adds a string to the beginning or end of a line containing a keyword in a text file. “`NODUP`” skips a command if the string entry already exists. “`NODUP`”’s comparison requires a case insensitive match, including spaces. If a file and/or a section do not exist, they will be created.

If a section is given then the specification of “`beginning`” or “`end`” applies to the beginning or end of the section.

If no section is given, then the specification of “`beginning`” or “`end`” applies to the beginning or end of the file.

If a keyword is given, then the specification of “`beginning`” or “`end`” applies to the beginning or end of the line which contains the keyword.

Note: Whenever a file is specified, you should give the full path. In addition, the position of the arguments are significant, and we recommend that you take this into consideration when distinguishing between a keyword and a string.

Example:

```
INSERT C:\config.sys end [common] ‘device=nvc.sys’ nodup
```

The line “device=...” is inserted at the end of the “common” section of c:\config.sys. If the line already exists anywhere in c:\config.sys, the “nodup” argument instructs N_DIST not to insert the string.

Example:

```
INSERT c:\config.sys end ‘device=nvc.sys’ nodup
```

The line “device=...” is inserted at the end of c:\config.sys. If the line already exists anywhere in c:\config.sys, no insertion is done.

Example:

```
INSERT c:\config.sys beginning ‘nvc.sys’ ‘REM ‘
```

A “REM” entry is inserted in the beginning of all lines in c:\config.sys containing “nvc.sys”.

REPLACE ‘string1’ ‘string2’ file {nocase} {all}

Replaces one or all instances of “string1” with “string2” in the specified file. “Nocase” results in a case-insensitive search. “All” replaces all occurrences of “string1” in the file.

Example:

```
REPLACE ‘REM CANARY 1’ ‘CANARY1’ C:\AUTOEXEC.BAT all
nocase
```

(Entered on **one** line.) This replaces all instances of the string “REM CANARY 1” with “CANARY1” in C:\AUTOEXEC.BAT.

Note: When specifying the file, you should always give the full path.

DELETE `path {directory} {hidden}{recurse}`

Deletes a path which can be a filename, a set of files, a directory, or a set of directories. When a directory is given, there is no need for the ”\”. N_DIST can even look for the hidden attribute and delete hidden files and directories. Wildcards can be used when specifying both files and directories. A directory must be empty in order for it to be deleted.

Example:

```
DELETE nvc.* hidden
```

This deletes all `NVC.*` files in the current directory, even if they are hidden.

Example:

```
DELETE C:\nvc directory
```

This deletes the `C:\NVC` directory, if it exists and if it is empty.

Note: When specifying a file or directory, you should specify the entire path. Otherwise, N_DIST will only check the current location.

DISPLAY *‘anything-of-your-choice’*

Displays any strings and/or variables that you choose. Several strings and variables can be displayed on one line. A “+” sign between sub-strings concatenates the strings into one continuous string.

Example:

```
DISPLAY ‘End of program. The current directory
is:’ $DirectoryDrive+$DirectoryPath
```

This would display the "`End of program...`" string followed by the values that are stored within `$DirectoryDrive` and `$DirectoryPath`. Since the two variables are connected with a "`+`", they would appear as "`D:\TEMP`", for instance.

GETDRIVE and GETPATH

GETDRIVE `$variable 'path'`

Extracts the drive letter from a complete path and returns the result in the specified variable.

Example:

```
GETDRIVE $drive 'c:\dos\chkdsk.exe'
```

will place the value `c:` in the variable `$drive`.

GETPATH `$variable 'path'`

Extracts the directory name from a complete path and places the value in the specified variable.

Example:

GETPATH $newpath 'c:\dos\chkdsk.exe'

will place the value `c:\dos` in the variable `$newpath`.

Note: If the path given in the argument does not have a filename at the end of it, then the value of the variable will be just the same as the path but without the final "`\`".

GETINI and SETINI

GETINI `$variable .INI-file [section] parameter`

Extracts a parameter value from an `.INI` file.

Example:

GETINI $Result c:\windows\win.ini [windows] 'load'

The variable `$Result` will contain the string that follows the "`=`" in the "`LOAD`" line of the section called [`windows`] within the file `C:\WINDOWS\WIN.INI`.

SETINI `ini-file [section] parameter value`

Inserts or changes a parameter in an `.INI` file. If the section and/or the file do not exist, they will be created.

Example:

```
SETINI c:\windows\win.ini [windows]
'load' 'c:\norman\nvcsys.exe'
```

(Entered on **one** line.) This would replace everything after "`LOAD=`" line in the [`windows`] section and replace it with `c:\norman\nvcsys.exe`.

Example:

```
SETINI c:\windows\win.ini [desktop]
'Iconspacing' '75'
```

(Entered on **one** line.) This would set "`Iconspacing`" in the [`desktop`] portion of `win.ini` to 75, regardless of any previous setting.

GOTO `#label`

Go to the line in the script starting with "`#label`".

Example:

```
GOTO #end
```

IF `<EXIST | !EXIST | ERROR | $variable> <= | !> 'value' {WRITE} {EXECUTE}`

The existing condition will process the next line in the script. If a condition is false, the next line will be skipped. “EXIST” checks for existence of paths and files, which may also be checked for write and execute rights.

Example:

```
IF !EXIST ‘c:\norman\nvc.exe’ execute
```

This checks to see if the file C:\NORMAN\NVC.EXE exists with execute rights. If so, then the next line in the script will be processed. If not, then the next line in the script will be ignored.

INCREMENT $variable

This command increments the integer value stored in the variable by one. Use this command to enable the script to create loops for repetitive tasks.

MAKEDIR path

Creates a directory or complete directory path.

Example:

```
MAKEDIR c:\temp\test\a
```

will create the complete path c:\temp\test\a.

Note: All non-existent directories in the path will be created at one time.

POPMESSAGE <message>

When this command is encountered in the script, N_dist32.exe will display a message box containing the ’message’, and wait for the user to push the OK button. If N_dist is running in ’very quiet’ mode (q!), the command will be ignored.

REGISTER and UNREGISTER

REGISTER `‘folder/group name’ {path ‘icon name’}{’common‘}`

Creates a folder/group and/or program icons on a workstation desktop. On the NT platform, `’common’` will register links in the common program’s area.

Example:

```
REGISTER ‘Norman’ c:\norman\win32\nvcnt.exe ‘NvcNT’
```

This will create an icon for `NVCNT.EXE` named “`NvcNT`” in the Norman folder. If the folder does not exist, it will be created.

In Windows NT and Windows 95/98, the folder “`Startup`” is automatically resolved to the correct startup folder name for the current language.

UNREGISTER `folder/group name {icon name}`

Removes a folder/group and/or program icons from a workstation’s desktop.

If no icon name is given, then the folder/group will be deleted only if no icons are present.

Example:

```
UNREGISTER ‘Norman’ ‘NvcNT’
```

Note: When processing the `REGISTER` and `UNREGISTER` commands, N_DIST will update the `WIN.INI` file and set a new date and time on the file. If you are running any access control software that monitors the date and time of this file, you will receive a warning from the access control software. Similarly, if the script contains commands to update files such as `AUTOEXEC.BAT` and `CONFIG.SYS`, your access control software will most likely warn you about these changes.

When either `REGISTER` or `UNREGISTER` are used, a copy of `WIN.INI` is stored as `NRMWINST.TMP` in the Windows directory. Changes are made to the file, and it is copied back to `WIN.INI`. Then `NRMWINST.TMP` is deleted. Please note, however, that there is an upper size limit of 1024 characters per line in `WIN.INI` in order to make these changes.

RENAME `source target`

Renames a file.

Example:

```
RENAME ‘c:\config.sys’ ‘c:\config.bak’
```

REMOVE `file ’word’ {all}`

Removes one or multiple lines which contains `’word’` from a file. `’All’` will remove all lines with the specified word. If `’all’` is not specified, only the first line containing the `’word’` is removed.

Example:

```
REMOVE ’c:\config.sys’‘buffers’all
```

This removes all lines containing the word ’buffers’ in the file `CONFIG.SYS`.

RUN `application {‘command line parameters for the application’} {’NOWAIT’}`

Runs an application and returns to the script when the job is finished. The application return code is stored in the reserved variable `$Returnvalue`.

On 32-bit platforms, `’nowait’` will cause the application to be run as a separate process. N_DIST will then immediately return.

Note: You should include the full path for the application. If the application and its path includes spaces, then you must enclose the complete path in single quotes.
In addition, you should never enclose the complete path and the command line parameters together within one set of single quotes.

Example:

```
RUN c:\norman\dos\nvc.exe ‘/lfc:\norman\report.log’
```

This runs the application `NVC.EXE` as if the user had typed this:

```
nvc.exe /lfc:\norman\report.log
```

SEARCH `$variable ENVIRONMENT ’string’`

Searches the environment for the specified value and returns the complete path (without the “=“) in the specified variable.

Example:

```
SEARCH $Environ_path1 ENVIRONMENT ‘buffers’
```

Note: When using this command, the word “ENVIRONMENT” must be used.

SEARCH `$variable file phrase`

Searches for a line within a file that contains a certain phrase. Returns the complete line in the variable. The phrase is case insensitive.

Example:

```
SEARCH $Line_for_phrase c:\config.sys ‘nvc.sys’
```

Let's assume that `c:\config.sys` contains the line "`device=c:\norman\dos\nvc.sys /t /f`". The command above instructs N_DIST to look for the line that contains the phrase "`nvc.sys`". When found, the complete line will be returned to the variable called `$Line_for_phrase`.

Note: If there are multiple instances of the phrase in the specified file, then N_DIST will only process the first instance.

SEARCH `$variable startpath filename {case}`

Searches for a file in a directory tree. You must state where the search should start (you can use wildcards here) as well as the filename. The filename is case-insensitive by default, but you can turn case-sensitivity on by using "`case`". The complete path is returned in the specified variable.

Example:

```
SEARCH $Path_of_file c:\*.* win.ini
```

This would search for the file `win.ini` in all of `c:`, and the complete path for `win.ini` would be returned in the variable `$Path_of_file`.

SET `$variable value`

Assigns a value to the specified variable. You may set one of the special variables, although we strongly recommend against doing so.

Example:

```
SET $Source_directory = M:\NVC_DIST
```

The `SET` command can be used to identify the source and target directories and folder name, for example. The following is an extract from the distribution script automatically generated during the NVC Administrator install:

```
NXD BEGIN
set $Source=’S:\NVCADMIN’
set $Target=’C:\NORMAN’
set $Folder=’Norman Virus Control’
set $Inst_Nvc_Win32=’Yes’
set $Inst_Nvc_Help=’Yes’
set $Inst_Nvc_Word6=’Yes’
set $Inst_Nvc_Word7=’No’
set $Inst_ReadMe=’Yes’
set $Icon_Nvc=’Norman Virus Control’
...
```

These commands will instruct the distribution script to pick the NVC files from the directory `s:\nvcadmin` and install them in the workstation’s `c:\norman` directory. The folder name is set to “`Norman Virus Control`”, and an icon with the same name is created.

WAIT `{x}`

Waits for a keystroke or for *x* seconds.

Example:

WAIT 3

If a number is given, the N_DIST will wait for that number of seconds. The wait time cannot be overridden.

If no number is given, then N_DIST will wait for a keystroke and store the value of that key in the pre-defined variable called `$KEY`. See page 73 for more information on special variables.

Options Used in the Distribution Script

It is possible to set options for N_DIST within the script itself. To use these options, simply insert a line in the script, which contains the command:

`option:`*`option name`*.

The script will run with the specified option(s) from the insertion point and onwards until you specify another option.

Available option	**Description**	**Notes**
Log	Log to default log file	By default, the log will be written to the current directory, and the filename will be the name of the script with a `.log` extension. You can also specify this option from the command line with the parameter `/LF(filename)`. Unlike `Options:Log`, however, you can specify the filename from the command line.
Append	Append to the log file	By default, N_DIST overwrites the log file. You can also specify this option from the command line with the parameter `/LG(filename)`.
Verbose	Verbose log file	You may also use the `/V` command line parameter.

Single	Single-step script interpretation	You may also use the /S command line parameter.
Nosingle	Turn off single-step	There is no associated command line parameter for this function.
Nosound	Turn off sound	There is no associated command line parameter for this function.
Quiet	Displays intro logo, system messages, syntax errors, and user messages only.	You may also use the /Q command line parameter.
Veryquiet	No output at all	You may also use the /Q! command line parameter.
Normal	Resets "Nosound" "Quiet", "Veryquiet", and "Verbose" options	There is no associated command line parameter for this function.

Example:

```
Option: Verbose
```

N_DIST will give verbose screen output from the "Option" statement and onwards.

In the following example, you want to run a certain part of the script in 'Single-step' mode, but the rest of the script should run in non-single step mode. Insert the "Single" option command before the section where it should apply and then the "Nosingle" option at the point where it should no longer apply:

```
...
option: single
#DeleteOld
   if exist $Target+’\DOS’
      goto #MakeDirs
...
option: nosingle
```

Special Variables

There are certain special variables that are set automatically by the system. They can be read and displayed, but we do not recommend that you set them yourself.

$Error

Is either “`TRUE`” or “`FALSE`”. The value is set to "`FALSE`" by the execution of the next instruction. The value is set to “`TRUE`” if a line in the script is executed unsuccessfully.

$Key

Holds the value of the keypress that is executed after a “`Wait`” command that does not have a time delay.

$Returnvalue

Holds the return value from a routine called by the “`Run`” command. If you wish to see the details on what happens during the “`Run`” command, use the “`Verbose`” option. The value of `$Returnvalue` will be held until the “`Run`” command is used again.

$Startpath

Contains the startup path of N_DIST.

$System

The value of this variable is "`WIN16`", "`WIN95`", "`WINNT`", "`OS2`" or "`DOS`", depending on platform/ program environment. This variable is set when you start N_DIST.

$Updated

This variable is set when you use the "`Copy`" command with "`update`". The initial value is "`NO`", but it will be set to "`YES`" if a 'copy - update' operation actually performs an update of a file.

Note: This variable must be reset manually by the "`SET`" command.

$Windir

This variable contains the Windows root directory.

Example: `C:\WINDOWS\`

$Winsysdir

This variable contains path to the Windows system directory.

Hints

When you run a distribution script you will receive messages whenever an error occurs.

Note: An error message does not necessarily indicate that something is wrong. If, for example, a condition is not met because it is irrelevant, an "error" message is generated for informational purposes.

Error messages are displayed when you run the distribution script with the “`Quiet`” option but not when you run the script with the “`Veryquiet`” option.

If you want a full report about the installation/update, we suggest that you run the distribution script with the verbose and log to file options from the command line:

```
N_DIST INSNVC.NXD /v /lf
```

If you wish to monitor the installation, you can specify the single-step option `/s`. This option requires keyboard input after each line in the script.

Custom commands

It is possible to write 32-bit DLLs containing specialized commands which can be called from within a script. Please contact Norman for more information on how this can be done.

Messaging

One of Norman Virus Control’s strengths is its messaging features. If you are a network administrator, and a virus was detected on a workstation, you want to know about it without relying on the user picking up the phone and calling you.

When NVC products are installed, you can receive virus alert messages in these ways:

NVC products on the workstation send messages via IPX communications to NVC for NetWare (“**FireBreak**”), which can then forward those messages to:

- another installation of NVC for NetWare on the WAN
- a NetWare group
- a network printer
- the system console

Note: In addition, FireBreak can forward the messages along as SNMP traps.

Additionally, the workstation NVC products themselves can send virus alerts as SNMP traps.

Please refer to the following sections called “NVC and IPX Communications” and “NVC and SNMP Traps” below for more details.

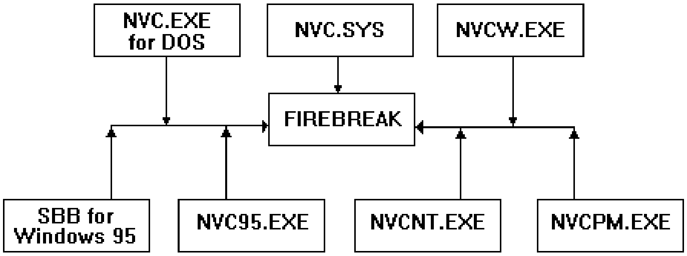
Furthermore, it is possible to customize messages that are displayed by NVC workstation products when a virus is detected. Please refer to the section called “Custom Messages on NVC Workstations” on page 92.

NVC and IPX Communications

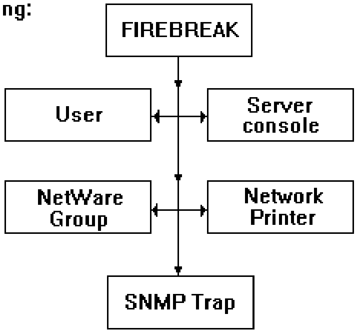
NVC for DOS/Windows 3.1x, Windows 95, Windows NT, and OS/2 (except the command line scanner for OS/2) can send virus alert messages to NVC for NetWare (“FireBreak”) through the Novell network protocol, IPX.

This intercommunication system allows for centralized monitoring of virus events on your LAN or WAN and is designed in a way that ensures proper information flow and low load on the network. The quantity of information sent is quite small, so this functionality will not negatively influence network performance.

IPX Messaging from Workstations to Server



FIREBREAK Messaging:



Requirements for Proper Communication

These are the versions of client software necessary for NVC v4.00+ workstation products (with the exception of NVC.SYS) to send IPX messages to FireBreak v3.60+.

DOS/Windows 3.1x:

- Netx
- VLMs
- NetWare Client 32 for DOS/Windows 3.1x

Note: Only VLMs support Canary on the server.

Windows 95:

- NetWare Client 32 for Windows 95

Windows NT:

- NetWare Client v4.0 for Windows NT

Note: Microsoft's Client Services for NetWare (available in Windows 95 and Windows NT) are not supported by NVC IPX communications. Therefore, in Windows 95 and Windows NT, you must be running NetWare's client software.

OS/2:

- NetWare Client v2.12 for OS/2

NVC workstation products (with the exception of NVC.SYS) will send messages to all versions of FireBreak via IPX, but only FireBreak v3.60+ will accept them.

IPX messaging from NVC workstations doesn't include messages about viruses found in memory.

FireBreak must be up and running on the NetWare server to which the workstations are attached. Although the workstation programs will send messages via IPX if FireBreak is not loaded, FireBreak must be loaded in order to accept the messages.

Multiple Server Networks

If the network contains more than server, the NVC workstation programs will pass a message to the server that has the highest priority. This feature follows the guidelines defined by Novell. As a rule, this server will be the server which the user is logged onto. Eventually the message can be sent to another server by using the "`Set Preferred Server`" command in `NET.CFG`. (See the NetWare documentation for more information).

Number of Messages Sent per Session

To minimize the load on the network, the system is designed to keep the number of messages that are being sent by the NVC workstation products to a minimum. The following table gives an overview:

Program	**Number of messages sent per “session”**	**Number of messages sent out by FireBreak**	**Other details**
`NVC.EXE` for DOS, `NVCW.EXE`, `NVC95.EXE`, `NVCNT.EXE`, and `NVCPM.EXE`	Each time a virus is found.	One only for each workstation scanning session.	All virus alert messages received by FireBreak from the workstations will be logged. In addition, information about these messages received by FireBreak will be displayed on FireBreak’s monitor screen.

`NVC.SYS` and the Smart Behavior Blocker for Windows 95	One only for each workstation boot session. That is, if you have had 5 warnings from `NVC.SYS` between the time that you turned your machine on and off, then only 1 alert will be sent to FireBreak. However, `NVC.SYS` does keep a running tally of virus warnings in `C:\NVCSYS.LOG` and the Smart Behavior Blocker for Windows 95 keeps a similar tally in `C:\NORMISA.LOG`.	One only for each workstation boot session.	All virus alert messages received by FireBreak from the workstations will be logged. In addition, information about these messages received by FireBreak will be displayed on FireBreak’s monitor screen.

Nature of the Message

The virus alert message sent from NVC workstations to FireBreak contains the following information:

- The name of the program that generated the message
- The user logon name (may be "Unknown" if the name is not available)
- The physical address of the workstation
- Date and time
- The name of the virus
- The location of the virus, for example filename, boot area, or memory)

FireBreak receives this information and assimilates it according to the manner in which it is configured. See the NVC for NetWare User’s Guide for more information.

Messages While Not Logged In

Even if an NVC workstation is not logged onto the server running FireBreak, the NVC workstation programs will still recognize the NetWare drivers in memory and send virus alert messages via IPX. When this happens, however, the user’s ID is unknown, and FireBreak reports it as “Unknown”.

NVC and SNMP Traps

SNMP stands for "Simple Network Management Protocol". It is a protocol which controls and monitors TCP/IP-based networks. An SNMP management station may poll an SNMP agent to obtain information about the agent system. The management station uses UDP (User Datagram Protocol) on port 161 to send a PDU (Protocol Data Unit) containing such a request.

A system does not have to be polled to transmit information to the management station. Another SNMP mechanism is called a “trap”. An SNMP agent may send a trap message to the management station without being polled. This is typically done when something extraordinary has occurred. UDP port 162 is used for SNMP traps.

Note: SNMP is not supported for the Windows 3.1x command line scanner (NVC32X.EXE).

NVC products support SNMP traps in the following ways:

- If our SNMP configuration file, `TCP_IP.CFG`, is present, Norman’s scanners for Windows 3.1x, Windows 95, Windows NT, and OS/2 as well as the Smart Behavior Blocker for Windows 95 and Cat’s Claw can send virus alerts as SNMP traps.
- In addition, virus alerts that are sent via IPX from NVC workstations to NVC for NetWare (FireBreak) can be forwarded from FireBreak as SNMP traps.

Note: We refer to `TCP_IP.CFG` as our SNMP extension. SNMPtrap

SNMPtrap is a utility to debug applications that are supposed to send SNMP traps when certain events occur. When started, the application uses WinSock to receive UDP/IP packets on port 162. When packets are received, they are dumped in a straight hex format on the screen. If the trap is recognized, it may also be decoded to yield the information contained within the ASN.1 PDU. This is done by clicking on the **Decode** button. If the trap is not recognized, the **Decode** button is not available.

The **Clear** button will clear the data output windows.

SNMPtrap will automatically try to do a reverse DNS lookup to determine the name of the sender of the trap.

SNMPtrap is only a tool for debugging. It is not intended to replace SNMP management software.

SNMP Extensions Diskette

Norman ships its SNMP extensions on a separate diskette from our NVC products for DOS/Windows 3.1x, Windows 95/98, Windows NT, and OS/2.

The SNMP extension diskette contains all the necessary files for PC-NFS, Lan Workplace and OS/2:

Filename	Description
README.TXT	This file contains any last-minute changes to this documentation.
SNMPTRAP.EXE	A trap debug routine. It can run on Windows 9x/ NT systems with TCP/IP installed.
SYSTEMS.TXT	An ASCII configuration file which contains the user-defined message and the names of the servers that are to be notified by the agent. This file is an example only — users are to modify this file according to their own needs.

SETUP.EXE	A configuration program that compiles the file SYSTEMS.TXT into TCP_IP.CFG, which is the configuration file used by the agents.
NORMAN.MIB	The SNMP MIB file that is common to all Norman products. Some network administration software may require this file to be able to interpret incoming NVC traps.
IP_TEST.EXE	This program sends a dummy trap to the list of management stations specified in the configuration file TCP_IP.CFG. It may be helpful when troubleshooting an installation. This is a DOS application which requires the presence of TSND.EXE. The application can not run with WINSOCK.

Preparing the SNMP Extensions

Note: Before preparing the extension, make sure that TCP/IP is up and running by issuing a "ping" command or something similar.

Copy all the files from the root directory of the SNMP extensions diskette into the directory where your Norman programs reside. Then depending on your environment, copy `TSND.EXE` from the corresponding subdirectory (`\PC_NFS`, `\LAN_WP`, or `\OS2`) into the directory where your Norman programs reside.

Now you are ready to configure the system. Edit the file called `SYSTEMS.TXT`, using DOS `EDIT`, Windows Notepad, or any other ASCII editor.

`SYSTEMS.TXT` contains a list of the machines in the network where SNMP traps are to be sent. You may also enter a message string that is to be included with the trap. An example of `SYSTEMS.TXT` is shown below:

```
; Norman Data Defense Systems TCP/IP server name file
;
; This file lists the names of the servers that are to
; be notified in case of a virus incident. Lines starting
; with ';' are ignored and can be used for comments. Up
; to 150 server names may be given in this file.
; A custom-designed message of max. 70 characters may be
; included on any separate line, starting with the character
; ’@’.
; This brief message may be used to identify the sending
; system in more detail, etc.
; You may enter a community name other than ’public’ by
; entering the name after the ’#’ character below.
; The program ’setup.exe’ will compile this file into a
; configuration file called ’TCP_IP.CFG’, to let NVC generate
; TCP/IP messages.
;
; The following is a list of target systems
norman
;
; The following is the user-definable message (not required)
@This is a custom message.
;
; The following is the community name (not required)
#public
;
```

As you can see, comments may be included on any line by starting the line with a semi-colon. A custom message is included by starting the line with the character “@”.

In this example, SNMP traps will be sent to the machine NORMAN. All specified machines **must** be available in the hosts table on the workstation or its name server so that valid IP-addresses can be resolved from the names. However, can you enter a specific IP-address manually.

If you want to use an SNMP community name other than the default of “public”, then edit the line starting with "#".

After the SYSTEMS.TXT file has been edited, you must compile it using SETUP.EXE in order to create the file TCP_IP.CFG.

Note: TCP_IP.CFG is required for any SNMP trap to be sent from an NVC workstation (i.e., it is not required for FireBreak to forward virus alert messages as SNMP traps).

Testing the SNMP Extensions

A test program called IP_TEST.EXE is included on the SNMP extension diskette. This program will send a test-trap to the systems specified in TCP_IP.CFG. IP_TEST.EXE will return a variety of messages, depending on whether or not the operation ended successfully. The MIB id for the test-trap is Internet.1007.1.7.

Receiving Traps

Normally, management software like ’HP OpenView’ is used for receiving traps. We have included a small 32-bit application called SNMPtrap on the SNMP diskette. This Windows 9x/NT application is a debug trap receiver which you can use to check that the system is transmitting traps as intended.

Troubleshooting SNMP Extensions

Following are explanations to some of the error messages that may be returned by the configuration program (SETUP.EXE) and the test program (IP_TEST.EXE).

Configuration Program Error Messages

The configuration program (SETUP.EXE) uses the data contained in the file SYSTEMS.TXT to generate a configuration file. This is a short overview of some of the error messages that may be encountered when running SETUP.EXE:

The messages below all refer to errors with the system names, custom message, or community name in SYSTEMS.TXT.

```
Error in line x: Server name is too long: <name>.
Error in line x: Illegal character in position y of name: name>.
Error in line x: Duplicate name: <name>.
Error in line x: Customized message is too long. Line ignored.
Error in line x: Community name is too long. Line ignored.
```

To correct an error, edit the file again. The error messages should tell you on which line the error occurred. Remember that:

- a system name cannot exceed 8 characters in length.
- a system may only occur once in the file.

The following messages are self-explanatory. The first two appear if there is a problem with the file SYSTEMS.TXT. Make sure that this file exists in the directory from where you run SETUP.EXE.

```
Error: Cannot find systems name file.
Error: Cannot open systems name file.
```

The next two messages appear when, for some reason, the setup program is unable to create the file TCP_IP.CFG.

```
Error: Cannot open configuration file.
Error: Could not write to configuration file!
```

Make sure that there are enough file handles available and that there is room enough on your disk.

The following message appears if there are no system names in SYSTEMS.TXT. The program will still generate a configuration file, but no traps will be generated by the applications.

```
Warning: No system names given
```

Test Program Error Messages

The test program, IP_TEST.EXE, will normally terminate with the following message:

```
Trap PDU sent, Result OK!
```

This means that a trap was successfully generated and sent to the systems specified in SYSTEMS.TXT. The program will list the systems to which it is sending traps. If the list does not match the systems that you specified in SYSTEMS.TXT, you will have to re-run SETUP.EXE to generate a new configuration file.

If you enter an illegal command line option when you run IP_TEST.EXE, the following will appear:

```
Illegal option: <option>
```

Available options are:

```
/F<text>  : Send alternate file location string

/T<number>: Send alternate enterprise trap number

/V<text>  : Send alternate virus name string
```

As it appears, it is possible to send any file location string, enterprise trap number and virus name string to the systems. Using this feature makes it easy to customize the service routines at the receiving end. Remember that the specific trap number indicates whether a virus is normal (1) or dangerous (2).

Following are brief descriptions of other error messages that may be produced by IP_TEST.EXE:

When the configuration file TCP_IP.CFG is not available in the NVC directory, you will see:

```
Cannot open configuration file.
```

The following message means that the local TCP/IP environment is not installed and/or not running.

```
The client is not installed.
```

When the configuration file `TCP_IP.CFG` is defective, you will see:

```
Configuration file checksum error!
```

To solve this, generate a new one using `SETUP.EXE`.

These messages result from a lack of free memory on the local system:

```
Unable to generate PDU!
```

```
Could not allocate buffer!
```

```
Out of memory error, xxx.
```

If there is a problem reading `TCP_IP.CFG` (most likely because of a problem in the DOS environment) you will see:

```
Could not read configuration file.
```

The following message means that there are problems with the port at the receiving end of the connection.

```
Could not open endpoint!
```

If your configuration file does not contain any systems to which to send traps, you will see:

```
Configuration file is empty!
```

The solution is to edit `SYSTEMS.TXT`.

The following message indicates that the program was unable to establish a connection with the specified system. Usually, this is because the receiving system is down. If many systems are not responding, only the last one will show in the error message.

```
Could not send to system name no. x!
```

If the specified system does not resolve to a valid IP-address, this message appears:

```
System name no. x is not valid!
```

Check that the system appears in your hosts-file or at your name server.

Installing the SNMP Extensions onto an NVC Workstation

Once you have verified that the `TCP_IP.CFG` file that you generated is valid, then you must copy it to a workstation that already has NVC installed.

You can place it in either of two places depending on which NVC products you would like to be generating the SNMP traps:

- If you wish either the DOS or OS/2 command line scanners to generate the SNMP traps, then place `TCP_IP.CFG` in the subdirectory of `C:\NORMAN` that is associated with the appropriate operating system. For example, if you wish the SNMP trap to be generated by the OS/2 command line scanner, then place `TCP_IP.CFG` in the `C:\NORMAN\OS2` directory. Remember to also place the appropriate version of `TSND.EXE` in the same directory.
- If you wish NVC for Windows 3.1x, Windows 95, Windows NT, or OS/2 (the PM version) to generate the SNMP traps directly, then place `TCP_IP.CFG` in the subdirectory of `C:\NORMAN` that is associated with the appropriate operating system. For example, if you wish the SNMP trap to be generated by the Windows NT scanner, then place `TCP_IP.CFG` in the `C:\NORMAN\WIN32` directory. In this case, there is no need for `TSND.EXE` to be copied to the workstation.

Note: Alternatively, you can place `TCP_IP.CFG` in the directory in the common Norman directory (`C:\NORMAN` by default) in either of these cases.

In order to distribute the SNMP installation throughout a network, you can distribute `TCP_IP.CFG` and TSND.EXE using N_DIST. See “Interpreting and Editing the Distribution Script” on page 55 for more information.

SNMP Technical Issues

SNMP is gaining popularity in network communities every day. It provides a well-structured common platform for network management and makes it easy to keep control in networks of any size.

Products from Norman have their own place in the SNMP MIB tree.

When a trap is sent from any of Norman's anti-virus applications, the trap number is 6, which means that the enterprise-specific trap number is set. The enterprise-specific trap from Norman anti-virus programs indicates whether or not the virus that has been discovered is dangerous. Specifically, trap number 1 indicates normal viruses, while number 2 indicates dangerous viruses.

Traps that are sent from Norman applications have the following IDs:

Program name	**ID**
`NVC.SYS and NORMISA.EXE`	.1.3.6.1.4.1.1007.1.1
`NVC.EXE for DOS`	.1.3.6.1.4.1.1007.1.2
`NVCW.EXE, NVC95.EXE, NVCNT.EXE`	.1.3.6.1.4.1.1007.1.3
`NVCPM.EXE`	.1.3.6.1.4.1.1007.1.4
`FIREBRK.NLM`	.1.3.6.1.4.1.1007.1.5
`IP_TEST.EXE`	.1.3.6.1.4.1.1007.1.7

In the future, additional applications will be added to the Norman sub-tree.

There is a variable bindings list included in Norman virus traps. It consists of four octet strings containing information about the virus attack.

Octet-string no.1: This string contains the origin of the message (user login and machine name).

Octet-string no.2: If possible, this string indicates the name of the virus that has been detected.

Octet-string no.3: This string indicates the location (filename) of the virus.

Octet-string no.4: Contains the user-defined message included in the local `TCP_IP.CFG` configuration file.

Custom Messages on NVC Workstations

Norman’s scanners have a built-in scripting language that allows you to customize the messages displayed by the scanners when a virus is found as well as standardize on certain options. When this feature is in use, it overrides some of the settings contained within each individual user’s Norman setup files.

The scanners must find a file called `NVC.MSG` either in the directory in which the scanners reside or in the common Norman directory (`C:\NORMAN` by default). This file is not provided by Norman. Rather, you must create it if you wish to use this feature.

All Norman’s GUI scanners support these custom message features, but the command line scanners only support some of the features. Therefore, unless otherwise specified, this discussion covers Norman’s GUI scanners.

Note: If you wish to use this feature with Norman’s scanners for Windows 95/98 and Windows NT, then you must do the following:
1. Start the Registry Editor.
2. Under HKEY_CURRENT_USER\Software\Norman\NVC, edit a new string value named “`NVCMSG`”. Set the value as the path where the `NVC.MSG` file resides.

Creating NVC.MSG

In order to create `NVC.MSG`, you must use certain tokens and block symbols, taking into account these rules:

- it must be a plain, ASCII file
- any line preceded by a “`;`” is considered to be a comment and therefore ignored
- any tabs and spaces preceding any line is ignored
- each token must reside on its own line
- each line must be less than 70 characters long and must end in a carriage return
- except for the conditions above, any ASCII characters may be used

To ensure that your work remains untouched by any user, you should set `NVC.MSG` to be read-only (and hidden, if you want to be extra careful).

There are five groups of what we call **Control tokens**:

1. User-defined messages. You may use block symbols in conjunction with these tokens.
2. Button
3. Report
4. Move file
5. Unattended mode

Control Tokens

When you use the control tokens, the Styles option will be grayed (and therefore unavailable). In addition, whichever of the scanning options that you override in the `NVC.MSG` file will also be grayed.

For instance, if you set the report name to always be `C:\NORMAN\VIRUS.RPT`, then the user will **not** be able to edit this filename in the scanning options dialog box.

The syntax for using tokens is as follows:

```
SYMBOL=VALUE
```

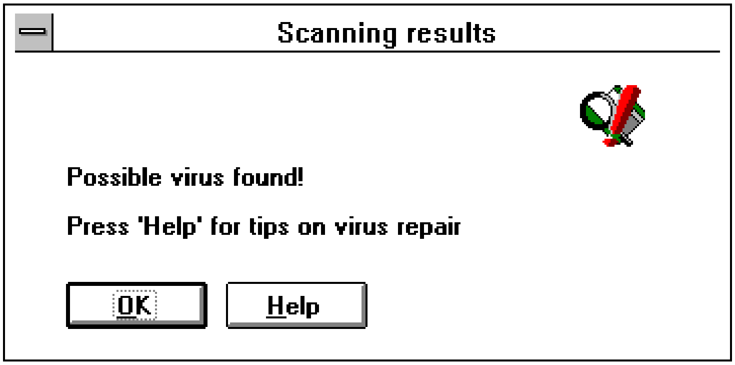
and no spaces are allowed on either side of the "=".

User-Defined Messages Tokens

Syntax:

```
@USR_MSG@=NONE
```

When this command appears in `NVC.MSG`, the GUI scanners will display their standard dialog box when a virus is found, even if any of the tokens listed below (such as `@VIR@`) are used:

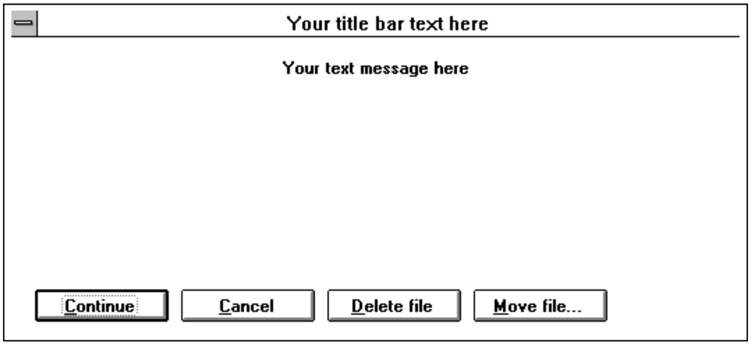


Scanning results

Possible virus found!

Press 'Help' for tips on virus repair

When this command does **not** appear in `NVC.MSG`, the GUI scanners will display any message that you construct, up to 10 lines long, when they find a virus.



Your title bar text here

Your text message here

As you can see, the user may choose to continue the scan, cancel (abort) the scan, delete the file, or move the file. All of these can be disabled by using the button tokens (see “Button Tokens” below). Your message can be static, as in “Don't panic”, or you may devise a dynamic message by using three additional tokens and/or block symbols:

Syntax:

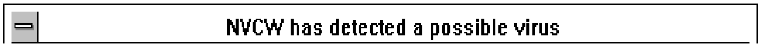
```
@HIVI@[title bar text]
```

Use this if you wish to customize the text in the title bar of the dialog box the GUI scanners display when they find a virus.

For example, the line:

```
@HIVI@NVCW has detected a possible virus
```

would produce a title bar like this when a virus is found:



NVCW has detected a possible virus

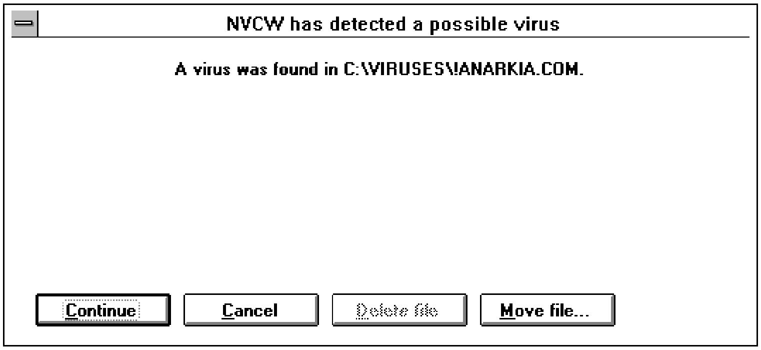
Syntax:

```
@LOC@
```

Use this if you wish your message to include the location of the virus that was found. For example, the line:

```
A virus was found in @LOC@.
```

in conjunction with the `@HIVI@` line above, would produce a screen like this when a virus is found:



A virus was found in C:\VIRUSES\ANARKIA.COM.

Note: the `@LOC@` token is supported by the command line scanners if the command is found within a block symbol structure (see below).

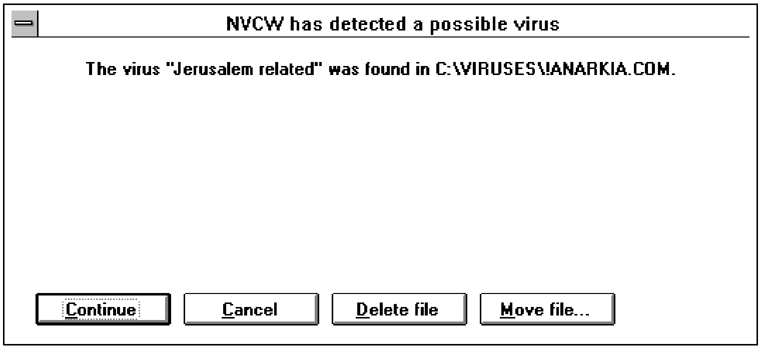
Syntax:

```
@VIR@
```

Use this if you wish your message to include the name of the virus that was found. For example, the line:

```
@VIR@ was found in @LOC@.
```

in conjunction with the `@HIVI@` line above, would produce a screen like this when a virus is found:



The virus "Jerusalem related" was found in C:\VIRUSES\ANARKIA.COM.

Note: The `@VIR@` token is supported by the command line scanners if the command is found within a block symbol structure (see below).

You may also use **block symbols** as part of your message if you wish your message to be dependent on what type of virus was found and where it was found.

The syntax for using a block symbol is as follows:

```
BLOCKSYMBOL
```

Text you wish to display when this type of virus is found or when the virus is found in this location

```
#END
```

There are five different block symbols:

Block Symbol	The text on the line following this block symbol is displayed when...
#INRAM	a non-stealth or non-destructive virus is found in RAM.
#C_INRAM	a stealth or destructive virus is found in RAM. (In this case, scanning must be aborted.)
#INBS	a virus is found in the Master Boot Sector or System Boot Sector.
#INFILE	a virus is found in a file.
#INARCH	a virus is found in an archive file.

Note: The block symbol structure is necessary in order to use this custom message feature for the command line scanners.

For example, the lines:

```
#INARCH
```

```
The virus "@VIR@" is within the archive
file  @LOC@.
```

```
#END
```

would produce the message “The virus xxxx is within the archive file xxxx” only when you have instructed the scanners to look within archive files, and it detects a virus within an archive file.

You may use any of these block symbols in conjunction with any other message you wish to display. In other words, these block symbol messages do **not** override other messages.

Note: All block symbols are supported by the command line scanners.

Button Tokens

Syntax:

```
@KEY_CNT@=NONE
```

When this command appears in `NVC.MSG`, the [`Continue`] button on the “virus found” dialog box will be grayed. This means that the scan will be aborted.

When this command does **not** appear in `NVC.MSG`, the [`Continue`] button on the “virus found” dialog box will be available. This means that the user can continue a scan when a virus is found.

Syntax:

```
@KEY_ABT@=NONE
```

When this command appears in `NVC.MSG`, the [`Cancel`] button on the “virus found” dialog box will be grayed. This means the user cannot abort a scan. In addition, the user cannot double-click the control menu in order to close the dialog box.

When this command does **not** appear in `NVC.MSG`, the [`Cancel`] button on the “virus found” dialog box will be available.

Syntax:

```
@KEY_DEL@=NONE
```

When this command appears in `NVC.MSG`, the [`Delete file`] button on the “virus found” dialog box will be grayed. This means the user cannot delete an infected file.

When this command does **not** appear in `NVC.MSG`, the [`Delete file`] button on the “virus found” dialog box will be available.

Syntax:

```
@KEY_MOV@=NONE
```

When this command appears in `NVC.MSG`, the [`Move file`] button on the “virus found” dialog box will be grayed. This means the user cannot move an infected file.

When this command does **not** appear in `NVC.MSG`, the [`Move file`] button on the “virus found” dialog box will be available.

Report Tokens

Syntax:

```
@REP_APP@=NONE
```

When this command appears in `NVC.MSG`, the user may not append the results of the scan to a previous report file.

When this command does **not** appear in `NVC.MSG`, the user may append the results of the scan to a previous report file.

Syntax:

```
@REP_OVW@=NONE
```

When this command appears in `NVC.MSG`, the user may not overwrite a previous report file.

When this command does **not** appear in `NVC.MSG`, the user may overwrite a previous report file.

Syntax:

```
@REP_NEW@=NONE
```

When this command appears in `NVC.MSG`, the user may not specify a new name for the report file.

When this command does **not** appear in `NVC.MSG`, the user may specify a new name for the report file.

Syntax:

```
@REP_NOR@=NONE
```

When this command appears in `NVC.MSG`, the user may not turn off the reporting function.

When this command does **not** appear in `NVC.MSG`, the user may turn off the reporting function.

For example, if you decide that reports should always be appended to a previous report, you would include these three lines in `NVC.MSG`:

@REP_OVW@=NONE

@REP_NEW@=NONE

@REP_NOR@=NONE

Syntax:

`@REP_FIL@=`

This token can appear two ways:

When this command appears as `@REP_FIL@=YES` in `NVC.MSG`, a report is always generated.

When this command appears as `@REP_FIL@=NONE` in `NVC.MSG`, a report is never generated.

Syntax:

`@REP_NAM@=[filename]`

This token specifies a name for the report file. For example, if you would like all report files to be written to `F:\USERS\VIRUS.RPT`, the line in `NVC.MSG` would appear as:

@REP_NAM@=F:\USERS\VIRUS.RPT

This token **must** be immediately preceded by the token `@REP_FIL@=YES`. Otherwise, this token will be ignored.

Syntax:

`@LOG_DIR@=YES`

This token specifies that all directories that are scanned will be listed in the report file.

Syntax:

`@LOG_FIL@=YES`

This token specifies that all files that are scanned will be listed in the report file. When the lines:

;@LOG_DIR@=YES

;@LOG_FIL@=YES

appear in NVC.MSG, the only information included in the report file is the names of infected files (and not all scanned directories and not all scanned files)

Move File Token

Syntax:

```
@MOV_DIR@=[directory]
```

This token instructs the GUI scanners to move all infected files to the specified directory. If a virus is found and this token is being used, the user cannot continue until the infected file is moved to the specified directory.

Unattended Mode Token

Syntax:

```
@MOD_AUT@=NONE
```

When this command appears in NVC.MSG, the [Don’t stop on virus] option in the scanning options dialog box is grayed. This means that the “virus found” dialog box will pop up after each virus that is found.

When this command does not appear in NVC.MSG, the [Don’t stop on virus] option in the scanning options dialog box is available. This means that the “virus found” dialog box pops up only once, no matter how many viruses are found.

Sample NVC.MSG File

```
;Sample NVC.MSG file which
;  1: Overrides info in NVC.INI
;  2: Defines user-defined messages to display when a
; virus is found.
; the @USR_MSG@=NONE line below (including the ";") turns
; on user-defined messages
;@USR_MSG@=NONE
; the line below sets the text of the title bar
```

```
@HIVI@NVCW has detected a possible virus
; the lines below are text that appear each time a virus
; is found
;they will appear along with the text from other tokens.
If FireBreak is running, a report of this virus detection
has been sent to the network administrator.
Call John at extension 1234 for help.
; the @MOV_DIR@ line below (incl the ";") suppresses
; mandatory moving of files
;@MOV_DIR@=g:\infected
;The line below disallows the ability to delete infected
; files
@KEY_DEL@=NONE
;the @KEY_MOV@ line below (incl the ";") allows the user
; to move a file
;@KEY_MOV@=NONE
; the line below mandates that the report be written to
; C:\VIRUS.RPT
@REP_NAM@=C:\VIRUS.RPT
;the 2 @LOG lines below (incl the ";") mean only infected
; files are logged
;@LOG_DIR@=YES
;@LOG_FIL@=YES
;the lines below set the message for when a virus is in a
; file
#INFILE
The virus "@VIR@" is in the file @LOC@.
Use NVC.EXE with the /CL parameter to clean the file.
#END
```

Index