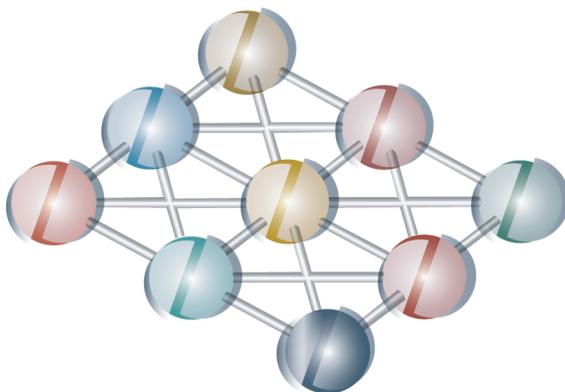




H+H Zentrum für Rechnerkommunikation GmbH

# H+H NetMan<sup>®</sup> XP





**H+H NetMan® XP**

---

**Apple Computer Inc.**

Power Macintosh and MacOS are registered trademarks of Apple Computer Inc.

**Citrix Inc.**

Citrix, ICA and WinFrame are registered trademarks and MetaFrame a trademark of Citrix Systems Inc.

**H+H-Software GmbH, Goettingen, Germany**

Virtual CD, CD Manager, HAN, ProGuard and NetMan are registered trademarks of H+H Software GmbH, Goettingen, Germany.

**IBM Inc.**

OS/2 is a registered trademark of IBM Inc.

**iXOS Software AG**

iXOS is a registered trademark of iXOS Software AG

**Microsoft Inc.**

Microsoft, Windows and NT are registered trademarks or trademarks of Microsoft Inc.

**Novell Inc.**

NetWare is a registered trademark and IntranetWare and NDS trademarks of Novell Inc.

**SAP Software AG**

SAP and R/3 are registered trademarks of SAP AG

**Unix System Laboratories**

Unix is a registered trademark of Unix System Laboratories

---

All rights reserved. Reproduction of all or part of this manual by written permission only. The company names, brand names and product names mentioned in this manual in addition to those listed above are in most cases protected by copyright, patent or trademark laws.

## Preface

Before you immerse yourself in the technical details concerning <sup>H+H</sup>NetMan and its functions, we would like to start you off with a few introductory remarks about the product.

Back when the very first version of NetMan was developed in 1990, nobody would have guessed that the product you are about to put into operation would be used around the world. Working from our headquarters in Germany, we have directed installations in renowned firms and institutions in England, France, Italy, the Netherlands, Canada, the USA, Japan, Korea and many other countries.

Today, NetMan is still unique the world over for meeting the challenges of off-line and on-line application distribution. Continuous adaptation to changes in the IT landscape and constant integration of all the processes required for optimized operation in networks that have numerous application programs—especially CD networks—are what secure the investments that have been made by NetMan customers since 1990. We look forward to writing the next chapter of our success story with you.

### *Notes on Working with This Manual*

This manual is divided into several parts, each of which describes a separate module in the NetMan Enterprise software suite. The first part describes NetMan's Base Module.



#### Note

*We strongly recommend reading the first four chapters of the Base Module manual before you install the software, or at least during the installation procedure.*

The subsequent sections describe the optional modules:

- Installer Module
- Terminal Server Module
- Language Module
- HTML View Module

In fact, each of these sections can be regarded as a separate manual, and all follow the same outline:

- The software module and all its performance features are presented in the "Introduction."
- The next section describes the basic configuration of the software.
- The first steps for putting the module into operation are then explained in detail after that.
- Practical examples and suggestions for use are given in the final sections.

The *NetMan Glossary*, included as an appendix to the overall manual, provides NetMan-specific definitions of the most important terms used here.

An *on-line version* of the NetMan manual is available on CD-ROM in the form of a PDF file. With this version, you can use the “Search” function in the Adobe Acrobat-Reader to find specific terms.

The NetMan manual is written with first-time users in mind, and provides an introduction to the basic concepts and operating design of NetMan. A complete list of NetMan commands and detailed descriptions of the program functions can be found in the on-line Help.

NetMan has special functions specifically for use in conjunction with Virtual CD, another well-known product from H+H. The NetMan user’s manual describes only the NetMan interface to Virtual CD, not the Virtual CD program itself.

The NetMan HAN (Hidden Automatic Navigator) and ProGuard modules are also available as independent products, which come with their own user’s manuals. Of these two products, the NetMan manual describes only the NetMan interface to HAN.

### ***Notes on the Enterprise Edition for Test Users***

NetMan runs in demo mode until you register your NetMan Enterprise software. In the demo mode, NetMan offers only the functionality of the *NetMan Workgroup Edition*, which is very limited in comparison to the Enterprise Edition. Furthermore, every time you start the program it reminds you that it is running in demo mode.

If you wish to test the NetMan Enterprise Edition before purchasing the software suite, please contact your NetMan software vendor to obtain a *temporary* full license.

### ***Prerequisites for Working with This Manual***

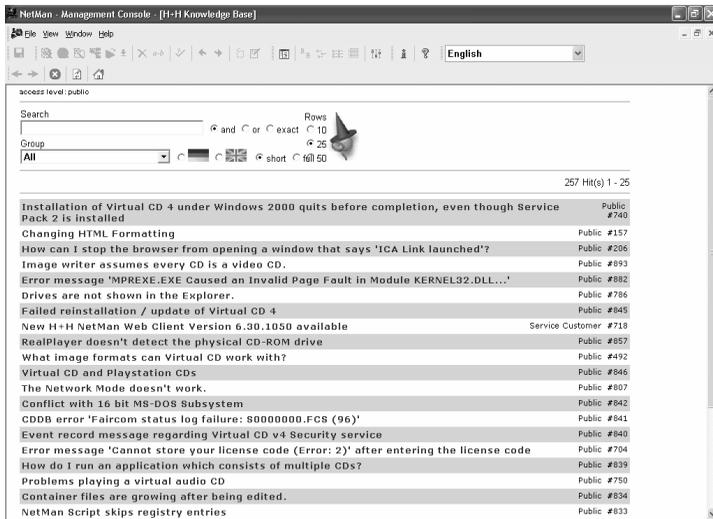
Before you can make the best use of this manual, you need to know:

- How to install personal computers and connect them in your network
- How to manage your PC operating system
- How the network operating system is administered
- How to install and operate the applications you want to use with NetMan
- How browsers process documents received from the Internet or distributed in your intranet (if you wish to manage this type of Internet resource with NetMan)

The NetMan manual does not provide instructions on these operations.

## Support

Visit the Support site on our NetMan Web server ([www.hh-netman.com](http://www.hh-netman.com)) at any time to obtain information about patches and download them as needed. Our comprehensive knowledge base also offers a wealth of information about NetMan, including suggestions for use as well as tips and tricks for working with NetMan.



If you have questions regarding support, please contact your software vendor.

You can send questions about NetMan software to the following e-mail address:

[supportnm@hh-zfrk.com](mailto:supportnm@hh-zfrk.com)

### *Before you ask...*

Before you contact your software vendor, please read the relevant sections of the manual and refer to the on-line Help in the NetMan program; if you are not sure where to look, check the Help index.

If you still have not found an answer, please provide the following information when you send us your question, or have it on hand when you call your software vendor:

- NetMan modules and version number
- NetMan serial number
- Network operating system and version number
- Text of any error messages and any relevant NetMan event log entries
- The steps required to reproduce the problem

## ***Ideas and Suggestions***

We are always happy to hear your ideas, comments, or suggestions for improvement. Please send them to:

H+H Software GmbH  
Attn: „NetMan“ Product Manager  
Maschmuehlenweg 8-10  
37073 Goettingen  
Germany

Tel: +49 (0)551 / 5 22 08-0  
Fax: +49 (0)551 / 5 22 08-25

Or send e-mail to:

supportnm@hh-zfrk.com; subject line: “NetMan”.

## ***Ask Bob!***

Bob is a Wizard, here to assist you in completing standard NetMan tasks. He also pops up now and again in this manual and in the on-line Help, to offer handy hints and tips and answer those “frequently asked questions” you are bound to have. You should make the acquaintance of this friendly little guy right at the outset.

We would also like to introduce the various manifestations that Bob takes on in this manual:

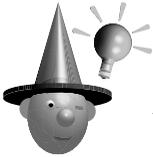


## *The Many Faces of Bob*



### ***The Note***

'Notes' tell you things you should know about NetMan, or about using the manual.



### ***The Tip***

'Tips' show you how to simplify tasks or how to avoid problems before they occur.



### ***The Discussion***

'Discussions' go into details about general topics or specific problems.



# H+H NetMan® XP Base Module





# Table of Contents

---

1. Introduction .....	1
Contents of This Manual .....	1
What is NetMan? .....	1
CrossMedia Approach .....	2
Application Framework .....	4
HTML Framework (HTML-based Information Sources) .....	6
System Requirements .....	7
Network Server .....	7
Client PCs .....	7
NetMan Licensing and Registration .....	7
2. Installing NetMan .....	9
Installing NetMan on a File Server .....	9
Destination for NetMan Installation – the NetMan Root Directory .....	10
NetMan Registration Data .....	10
Language Setting .....	11
Selecting NetMan Modules .....	11
Server Name for the NetMan Service .....	12
Installing the NetMan Service .....	13
Registering NetMan .....	17
Subsequent Installation / Registering Additional NetMan Modules .....	18
Updating from NetMan Version 1.5 .....	18
3. Configuring NetMan Clients .....	19
Client Installation .....	19
Opening the NetMan Desktop .....	20
NetMan Command Line Call .....	21
4. Global NetMan Configurations .....	23
The NetMan Client .....	23
NetMan Components .....	27
NetMan Directories and Network Rights .....	27
Defining NetMan Administrators .....	29
Configuring Drives and Shares .....	29
Application Drive .....	31
Commonly Used Network Resources .....	32



NetMan Startup and Shutdown .....	33
NetMan Settings.....	35
Global Settings Dialog Page .....	35
Language Dialog Page .....	37
Network Resources and Dynamic Connection Dialog Pages .....	38
information View Dialog Page .....	38
User ID/Station ID Dialog Page .....	38
Runtime Recording Dialog Page .....	39
LDAP Dialog Page .....	40
Advanced Dialog Page .....	40
5. Integrating Applications and Hyperlinks .....	41
The Two Types of NetMan Configuration .....	42
First Steps with the Management Console .....	46
The Public Desktop .....	48
Program Actions .....	52
Additional Program Properties .....	53
Additional Properties of NetMan Hyperlinks .....	57
Creating and Deleting Desktop Entries .....	57
Your First Application .....	63
Permission to Access Configurations and Actions .....	66
Creating Additional Desktops .....	71
Startup and Shutdown Configurations .....	73
Integrating HAN Accounts .....	74
The Trace Monitor and the Internal Architecture of NetMan .....	77
CD-ROM-based Applications .....	81
NetMan Actions .....	90
Controlling an Action Sequence .....	92
Simple Examples of the Most Frequently Used Actions .....	95
Complex Actions .....	100
The Windows Script Action .....	102
Passing Arguments to Scripts .....	103
Output to the Trace Monitor .....	104
Access to the NetMan Environment .....	105
6. Users, Stations, Groups and Profiles .....	107
NetMan Users .....	108
NetMan Stations .....	110

NetMan User Groups .....	112
NetMan Station Groups .....	113
NetMan User and Station Profiles .....	114
User Profiles .....	115
Station Profiles .....	117
<b>7. Statistical Analysis of Log Files .....</b>	<b>119</b>
Statistical Analysis with the NetMan Statistics Program .....	120
The Main Table .....	121
Table of Concurrent Use .....	124
Analyzing Data with the NetMan Statistics Program .....	124
<b>8. HTML Documents as User Interfaces .....</b>	<b>137</b>
Introduction .....	137
Method 1: Presenting Complete NetMan Desktops in Web Pages .....	139
Method 2: Inserting NetMan Configurations in Web Pages .....	142
Starting a NetMan Configuration from a Web Page .....	145
Using the Citrix Client .....	145
Using NetMan Start Files .....	148
Practical Examples .....	149
Converting a NetMan Desktop to HTML .....	149
Integrating a NetMan Desktop in a Web Page .....	156
Inserting NetMan Configurations in Web Pages .....	160
HTML Format Templates .....	165
Designing Your Own Layout .....	165
Formatting Embedded Browser Objects .....	166
Formatting the Left Margin Indentation for Desktop Entries .....	167
Using Style Sheets .....	167
<b>9. NetMan Explorer .....</b>	<b>169</b>
Examples in the Public Desktop .....	170
NetMan Explorer Settings .....	173
Using Wildcards .....	174
"Global" Dialog Page .....	175
"Included URLs" Dialog Page .....	176
"Excluded URLs" Dialog Page .....	177
Passing a URL as a Command Line Argument .....	178
Notes on the NetMan Explorer Settings .....	178



Testing the NetMan Explorer Configuration.....	180
Notes on Using the NetMan Explorer in Protected Environments .....	181
<b>10. NetMan Helper Programs .....</b>	<b>183</b>
The Application Library .....	183
Helper Programs for the 'Execute' Action .....	185
HHCOPY – Copying Files and Directories .....	186
HHMKDir – Creating Directories .....	189
HHDelete – Deleting Files and Directories .....	190
HHCMD.EXE – Hiding Command Execution .....	191
HHDummy.exe – “Do Nothing” or “Wait” .....	192
HHSetAtr.exe – Setting File Attributes .....	193
NMLogin.exe – Executing a Server Login .....	194
NMNCon32.exe – Connecting a Drive .....	195

# 1. Introduction

## *Contents of This Manual*

**Chapter 1, “Introduction”,** provides an overview of the NetMan software, including performance features, system requirements and basic components

**Chapter 2, “Installing NetMan”,** describes the steps required to install NetMan on a network server

**Chapter 3, “Configuring NetMan Clients”,** shows you how to call the “NetMan Client” interface from a workstation. If you work through this chapter while installing NetMan, that installation will be completed and you can start NetMan when you get to the end of the chapter.

**Chapter 4, “Global NetMan Configurations”,** introduces the NetMan Client and NetMan’s system programs and describes how to adjust the settings for your network configuration. The basic configuration options are explained to help you find the optimum NetMan setup for your network requirements and personal preferences.

**Chapter 5, “Integrating Applications and Hyperlinks,”** tells you how you can use NetMan to make applications available to your network clients and describes the properties you can assign using NetMan, including licensing, event logging, access rights, and much more.

**Chapter 6, “Users, Stations, Groups and Profiles,”** explains NetMan’s internal definitions of groups and profiles and describes how you can use these concepts to configure group-specific and station-specific presentations of centrally defined applications.

**Chapter 7, “Statistical Analysis of Log Files,”** acquaints you with NetMan’s basic statistical evaluation functions.

**Chapter 8, “HTML Documents as User Interfaces,”** explains the use of the HTML Wizard, which is included in the Base Module and is basically a “lite” version of the NetMan HTML View Module.

**Chapter 9, “NetMan Explorer,”** tells you how you can make the presentation of HTML documents configurable using the NetMan Explorer.

**Chapter 10, “NetMan Helper Programs,”** describes the use of NetMan’s auxiliary or “helper” programs.

## *What is NetMan?*

NetMan is thoroughly up-to-date Windows software, with 32-bit architecture throughout and the kind of easy-to-use operating elements and context-sensitive Help that you are already familiar with from your operating system.

NetMan is easy to install. At the client end, in fact, there is no installation required; all you have to do is create a shortcut on to a NetMan Desktop each client PC. You also have the option of presenting your NetMan interface in a browser.

NetMan provides a built-in interface to our NetMan Web server (at [www.hh-NetMan.com](http://www.hh-NetMan.com)) which gives you direct access to updates as they become available, as well as to a comprehensive, up-to-date knowledge base.

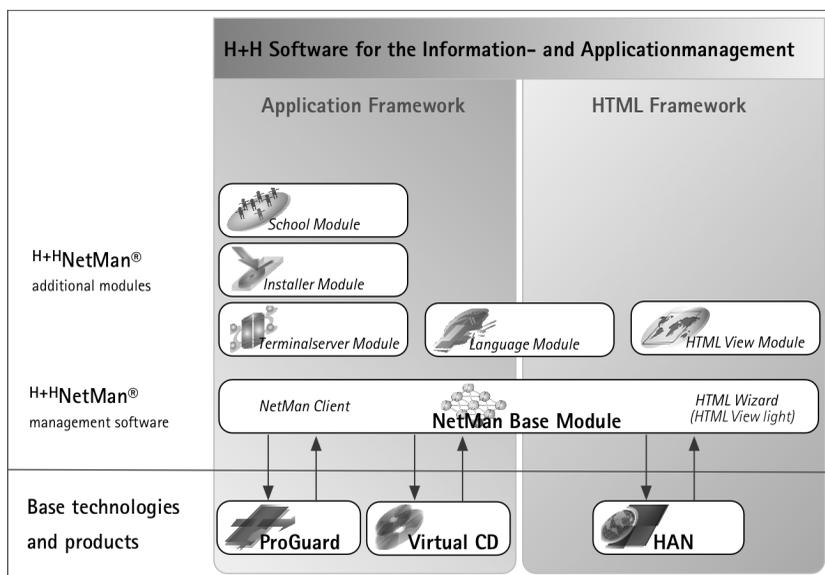
### **CrossMedia Approach**

NetMan is a software suite for application management. Whether you work with a local area network, an intranet or the Internet, NetMan enables you to manage your applications efficiently and present them attractively to users.

Your network resources may include Windows-based applications and programs, documents and/or HTML-based information sources. NetMan integrates all these different types of resources in a unified user interface, whether you work with Windows or HTML-based access.

The integration of different types of resources in a single system is implemented through

- the **Application Framework** (application management), and
- the **HTML Framework** (management of HTML-based information sources in your intranet or the Internet)



These include the following performance features:

- Central definition of program properties in the NetMan database
- Central assignment of access privileges to NetMan users and user groups, NetMan stations and station groups, specified ranges of IP addresses, host names and network groups.
- Intuitive user interfaces (Explorer or browser-based) that let you group resources in logical structures
- Automatic event logging when users and stations access the system
- Helper programs for analyzing runtime problems in your applications.

NetMan presents centrally managed resources on all your workstations. Configuration and updating of the resources on each workstation—including desktops, Start menus and browser favorites—can be implemented from a **central** location.

You can provide a variety of resources in a **unified** format, because NetMan lets you configure access through

- URLs, or
- Windows program calls.

These options are illustrated in the following two examples:

1. Your users access central information services through their browsers: You can integrate any number of Windows applications in these services, because when you define a NetMan application call, it is automatically assigned a URL by which it can be called.
2. Your users work in a Windows environment: You can provide customized shortcuts on the Windows desktop to any intranet or Internet resources.

Because NetMan is a modular suite, you have the choice of configuring it for **exclusive** use as either:

- application management software, or
- an Internet resource manager.

You can integrate additional NetMan modules at any time to expand the functionality of your NetMan software. This scalability secures your investment and leaves all options open for future adjustments and additions.

The *Language* Module lets you offer different language versions for end users.

You can assign users and stations to NetMan profiles, which let you define the user language (with the *Language* Module) as well as other options for information access.

User privileges for access to information sources can be made dependent on any of a

variety of conditions, such as the presence of specified entries in INI files or the Windows Registry on the client side, client host name or IP address, client operating system, and more.

A centralized (NT-based) NetMan Service informs you within seconds about license availability and station activity.

### ***Application Framework***

NetMan expands your programs by adding new properties. Read on for an explanation of just what this means for your network:

The program properties that your operating system offers for a Windows shortcut are:

- program call
- working directory
- icon
- program window state on start-up (maximized, minimized, normal).

With NetMan, not only can you define these properties *centrally*; you can also define *additional properties* that greatly expand your range of possibilities:

- Define a maximum number of parallel users (licenses) for each application.
- Create detailed records of application use, sorted by user and station; this data can also form the basis for easy-to-read tables and graphs generated with NetMan's powerful statistical evaluation tools.
- Assign 'execute' privileges for NetMan users (independent of complex file and directory privileges configured at the network level).
- Assign 'execute' privileges for NetWare NDS groups, as well as local and global NT and LDAP groups.
- Configure licensed applications to close automatically—and release the license—if they are left unused for a certain period of time.
- Provide information in HTML format about individual applications.
- Define whether multiple parallel instances of an application are allowed on a client machine.
- Define mutually blocked applications (i.e., application "B" cannot be started on a given client machine while application "A" is running on that station).
- Activate or deactivate applications or groups of applications, with a customized message to the user (such as, "This application is undergoing an update at the moment. Please try again later").

These functions are implemented as expanded *program* properties that you can assign. In addition to all these features, NetMan provides you, as administrator, with deeper layers of functionality for each application, involving far more than just a program call:

NetMan shifts the focus from the *program call* to *application management*, which significantly increases your control in the network.

With NetMan, each program call—e.g., “LexiROM.exe”—is a single *action* within an elaborate application definition (“configuration”). To call the LexiROM program under NetMan, you would define a NetMan *configuration* (called “LexiROM,” for example) which can include any number of other actions in addition to the program call. This sequence of actions is then executed when a user activates the LexiROM shortcut. Here is an example of the functions that a NetMan configuration can perform:

- Map a network drive for the application
- Make the required resources available (for example, by mapping the LexiROM CD)
- Call the program
- When the program is ended: Undo drive connections that were mapped by the preceding action(s).

This example includes only a few of the many NetMan actions at your disposal. Other actions cover broad range of functions, from password prompts to running other programs before or after the activated program. And the execution of any given action can be made dependent on any of a variety of conditions, defined in the form of ‘execute’ privileges.

Furthermore, you can define action return values which are stored in variables; for example, to integrate user input in the processing of a NetMan *configuration*. NetMan’s own interface to the Windows Scripting Host lets you combine the many options available in NetMan with scripts you write yourself.

The following modules are also available for use in application management:

The *NetMan Installer* monitors the local workstation during application setup; changes made by the Setup program at the file level or in the Windows Registry are documented and can be written up in the form of scripts. You can insert these Installer scripts in NetMan configurations to distribute application components where and when you need them in the network.

The *Terminal Server Module* enables access through terminal servers, such as Citrix MetaFrame or Microsoft Terminal Server, and lets you defined authorized host names or IP addresses for working with anonymous users. With this module and a MetaFrame server you can allow platform-independent access (for instance, from Unix, Macintosh or OS/2 terminals) to your Windows applications.

The *ProGuard Module* lets you centrally configure permitted programs and processes on the client workstations. It comes with a separate user’s manual, and thus is not described in detail here. For an example of how this module works, see “Notes on Using the NetMan Explorer in Protected Environments” in chapter 9.

The *NetMan for Schools* Module includes special programs and pre-defined NetMan configurations for use in educational contexts.

## ***HTML Framework (HTML-based Information Sources)***

The NetMan Explorer enables access to intranet and Internet resources. You can control this access by restricting navigation options for end users. For example, you can deactivate operating controls in the browser, and define permissions for specific hypertext links. Users can access only those HTML-based resources that you permit.

In conjunction with the Terminal Server Module, you can allow access to Windows applications through the intranet or the Internet to be started in a browser page.

The *HTML Wizard* is included in the NetMan Base Module. This program lets you present NetMan desktops in the form of HTML pages, or add NetMan configurations to existing HTML pages. This simplified version of HTML View creates only static HTML documents, which means you cannot define user-specific or station-specific presentations of NetMan configurations.

The *HTML View* Module, on the other hand, lets you create “dynamic” HTML pages. HTML View analyzes the privileges granted to users and stations, and presents only the permitted resources as defined for the particular client. It also implements licensing controls for the requested resources and can react to inquiries based on a client’s browser type, operating system, host name or IP address.

The *HAN* Module lets you provide fully automatic access to Internet resources. “Fully automatic” in this context means users do not need to logon to the desired Web site for a certain resource, because authentication is handled using enterprise-internal databases. This means access can be permitted even when the client’s IP address that does not fall within the IP address range of the institution at which he or she works.

## System Requirements

### Network Server

You can run NetMan on a Microsoft NT server (version 4.0 or later), a Novell server (version 3.12 or later) or other PC-compatible server.



#### Note

*The NetMan Service requires a computer running Microsoft NT (server or workstation; version 4.0 or later).*

### Client PCs

In conjunction with the Terminal Server Module and a MetaFrame server, you can enable platform-independent access.

NetMan programs require a Windows operating system (Windows 98, ME, NT 4.0 (Service Pack 6a), 2000 or XP). The Microsoft Internet Explorer, version 5.0 (or later) must also be installed. On administrative stations, we recommend generous proportions for both RAM (256 MB) and monitor (19 inches).

### NetMan Licensing and Registration

When you first install NetMan, it runs in demo mode. The demo version has only limited performance features, and reminds you that it is in demo mode every time you start the program.

The NetMan software must be registered before the full version will run. For details on this procedure, see chapter 2, “Installing NetMan.” If you wish to test a full version—including the modules you need—before purchasing NetMan, contact your software vendor for details on obtaining a temporary license.

NetMan offers two different schemes for client licensing:

- With the *Concurrent User* scheme, user rights are assigned for simultaneous parallel use of the NetMan Client.
- With the *Named License* scheme, licenses are assigned by workstation and are valid for up to 40 days.



## 2. Installing NetMan

This chapter describes how to install NetMan, the NetMan Service and other NetMan modules, as well as how to register the software so you can run the full licensed version.

### *Installing NetMan on a File Server*

Depending on the modules you install, NetMan requires up to 100 MB capacity on the hard disk. Keep in mind that the NetMan databases will be filling up with data while you work with the NetMan system, and be sure to reserve enough space on the hard disk for this as well.

You can install NetMan from any workstation. The Setup program copies files only to the directory you name as the destination, and creates Start menu entries on the workstation used for installation.

NetMan is usually installed on a file server.



#### Tip

*If you want to use NetMan in a “pure” terminal server environment, we recommend installing NetMan on the terminal server. A “pure” terminal server environment in this case means that NetMan application calls are launched only in terminal server sessions, not directly from the LAN. If you have several terminal servers with load-balancing functions (i.e., a “server farm”), select a fast file server for your central NetMan installation.*

Call *setup.exe* from the H+H NetMan CD-ROM and enter the information requested.

## Destination for NetMan Installation – the NetMan Root Directory

Following the “Welcome” window and the licensing agreement, you are prompted to specify the destination for your NetMan installation.

The destination is usually a path on a file server in your network. You can enter either the drive letter and pathname of the desired directory or the UNC designation (\\<server>\<share>\).

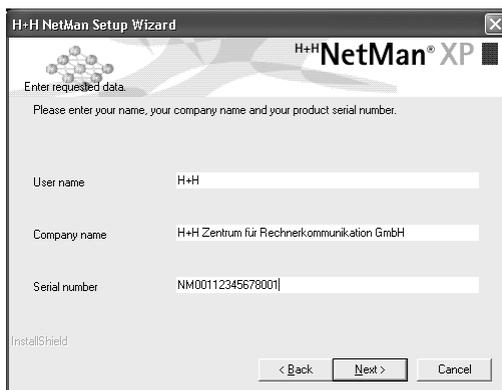


### Note

*The installation path is referred to in this manual as the NetMan root directory.*

## NetMan Registration Data

The next dialog prompts your registration data. You will find the serial number printed on adhesive labels included with the CD; you will need this number again later, when you call in to receive your registration key.



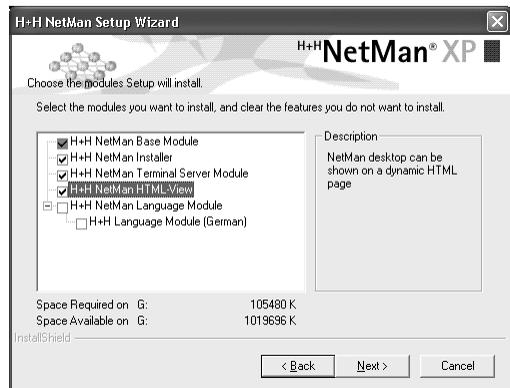
## Language Setting

NetMan's administrative programs are installed in English and German. If you would like to have a choice of languages available for your users, you need the NetMan Language Module. If you do not have the Language Module, the end-user language is defined here:



## Selecting NetMan Modules

This dialog offers you the option of selecting additional modules for installation.



- **H+H NetMan:** Installs the NetMan Base Module.
- **H+H NetMan Installer:** Monitors the Windows systems on the workstation during application setup. You can save changes to the system in a script for distributing the application to other clients in the network.
- **H+H NetMan Language Module:** Makes NetMan multilingual by installing other languages in addition to the NetMan base language.
- **H+H NetMan Terminal Server Module:** Allows you to run NetMan on

terminal servers, control access based on client IP address or host name, and take advantage of expanded security features.

*H+H NetMan HTML View:* Shows NetMan desktops and configurations on dynamic HTML pages. Which applications and/or hyperlinks are included in a given HTML page depends on the privileges granted to the NetMan user or workstation requesting that page.

For more information on these modules, see “What is NetMan?” in the introduction to this manual.



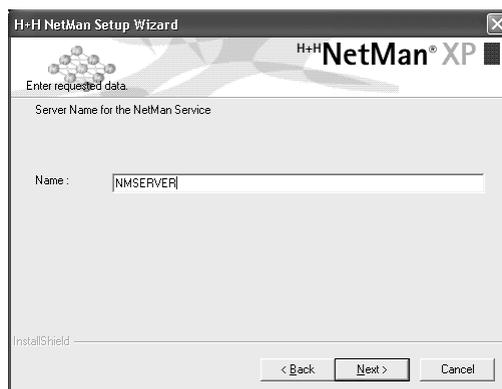
### Note

*The HAN and ProGuard Modules come with their own manuals and setup programs.*

## Server Name for the NetMan Service

The NetMan Service monitors your workstations for licensing and data logging functions. This is an NT service and must be installed on an NT station (see also Section 2.2, “Installing the NetMan Service”).

In this dialog, enter the name of the server on which the NetMan Service is to be installed. Please enter the host name or the IP address of the server. If you are not sure which server you should specify here, refer to the section below entitled “Installing the NetMan Service” for detailed information:



The last dialog lets you specify a program directory for the NetMan shortcut.

The program now shows a summary of the information you entered during setup.

Check the information and click on **BACK** if you need to make any changes. Click on **NEXT** to execute the installation of NetMan. Check the 'Readme' file for the latest information on NetMan.

## Installing the NetMan Service

The NetMan Service is an NT service and must be installed on the console of the NT station designated in the NetMan setup program.



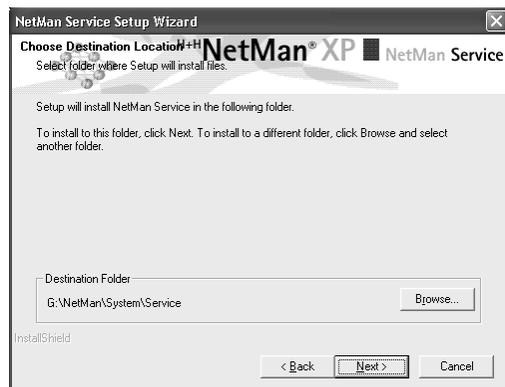
### Note

*The Setup program requires that the Microsoft Installer, version 2 or later, be installed on this computer as well.*

There are two slightly different procedures for installing the NetMan Service, depending on whether or not it will be installed on the same server as the one NetMan is installed on.

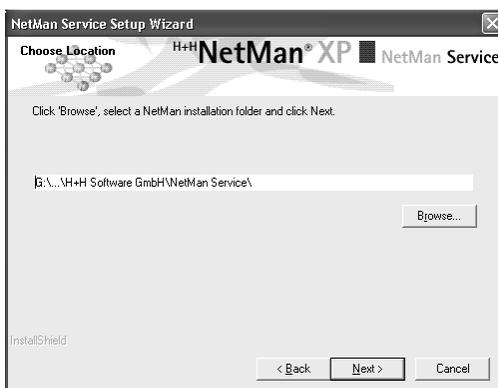
To install the NetMan Service, call `%NMHome%\System\Service\Setup\setup.exe` from the NetMan root directory.

Make sure you use the local drive letter when you name the NetMan root directory. If the Setup program is called from a local path, the Setup program suggests a path within the NetMan directory structure:

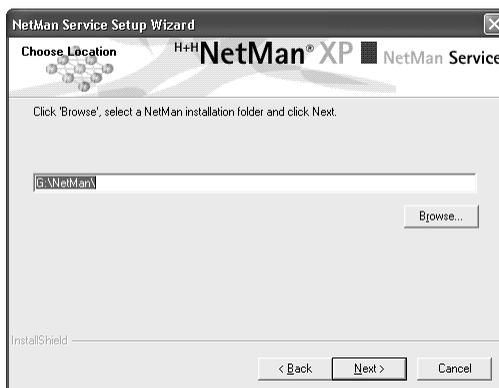


Confirm the suggested path, unless it is absolutely necessary in your case to use a different path.

If you call the Setup program for the NetMan Service from a network path, the suggested installation directory is the Windows program directory:



If the NetMan program is installed on the same computer, you need to enter the local path to the NetMan root directory:



Once it has been installed, the NetMan Service runs under a system account. Make sure this account has sufficient privileges in the NetMan root directory (see also “NetMan Directories and Network Rights” below).



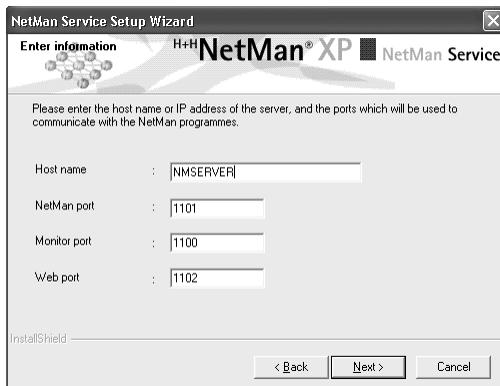
#### Note

*In most cases, access privileges have to be modified if the NetMan root directory is on a different computer; you may also find it necessary to modify privileges if NetMan is on the same computer but not installed in the computer's system directory.*

If NetMan and the NetMan Service are on different computers, enter the path to the NetMan root directory in UNC syntax (\\<Server-name> \<Share> \<NetMan>):



Next, you can configure the ports used by the NetMan Service for communication with other NetMan programs.



The following information is required in these input fields:

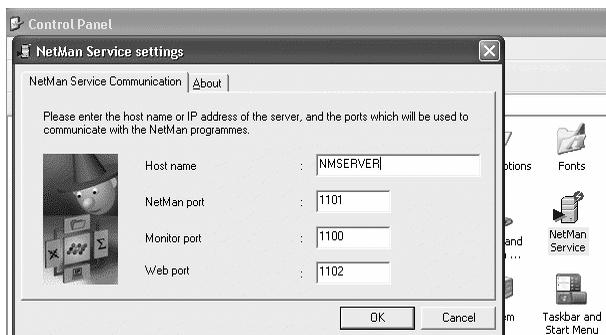
- **HOST NAME:**  
Host name or IP address of the computer on which the NetMan Service runs.
- **NETMAN PORT:**  
For communication between the NetMan Service and the NetMan Client.
- **MONITOR PORT:**  
For communication between the NetMan Service and the station and license monitors.
- **WEB PORT:**  
For communication between the NetMan Service and HTML View.



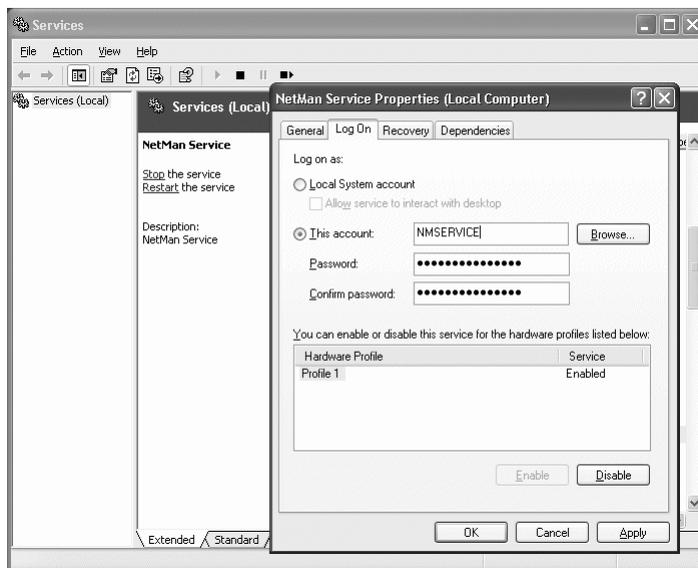
#### Note

*In most cases you can accept the defaults offered here. Make sure the ports you name here are accessible over any routers in your data path.*

You can change the ports later in the Windows Control Panel, if necessary.



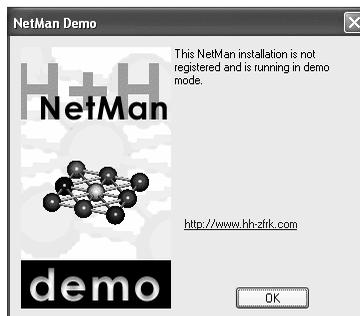
If the system account you entered does not have privileges in the NetMan root directory, you need to change configurations in the Control Panel (under Administration / Services):



In addition to standard NetMan user permissions, the account named must also have 'write' privileges in the `prot` directory of the NetMan installation. For details, see also "NetMan Directories and Network Rights" below.

## Registering NetMan

NetMan must be registered in accordance with the license you purchase before you can use the full version and any modules you have acquired. Before it is registered, NetMan runs in demo mode and the following message is displayed every time you start the program and any time NetMan is used to launch an application:



### Note

*The NetMan modules, licensing scheme and number of licenses are all defined when you order the software and have to be registered once the software is installed.*

NetMan comes with a program for software registration; this program can be found in the NetMan Client, under SYSTEM ADMINISTRATION / CONFIGURATION / UTILITIES / REGISTRATION directory.

Call your software dealer to obtain the registration key.

The following information is required for registration:

- The registration data entered during installation (name and company)
- NetMan serial number (see the enclosed adhesive labels)
- The identification number generated

This data is displayed in the registration dialog:

Enter the registration key. The next window shows the modules and number of NetMan client licenses purchased. The next time you start the NetMan Client, the full NetMan version will run.

Some registration keys can be loaded from other directories. For example, if the NetMan HAN Module or ProGuard Module is already installed in your network, you can import the registration codes for those programs into NetMan. To do this, click on **IMPORT LICENSE DATA** and import the *NMCfg.dat* file.



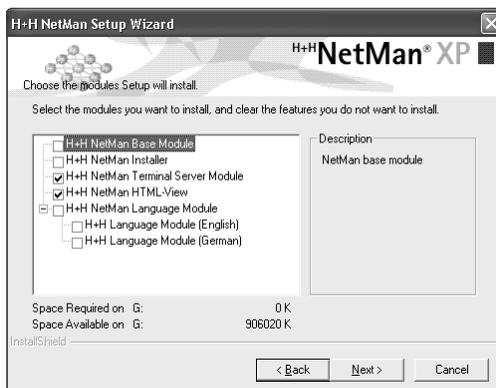
### Note

*Client licenses for NetMan and the HAN and ProGuard Modules can be used together, and registration codes imported from one program into another.*

## Subsequent Installation / Registering Additional NetMan Modules

Modules acquired subsequently to your original NetMan purchase are installed by the same program used for the initial installation: start *setup.exe* from the H+H NetMan CD-ROM.

Enter the NetMan root directory as the target directory for installation. A dialog opens for selecting the modules to be installed:



### Note

*The procedure for registering the new modules is the same as for registering the NetMan program after initial installation.*

## Updating from NetMan Version 1.5

The Setup program is also used to update earlier versions to the present NetMan version. Make sure no one is working with NetMan while you perform the update. Run *setup.exe* and enter the path to the root directory of your NetMan installation. If you have an earlier NetMan version than 1.5, you need to update that version to NetMan 1.5 before you can update to NetMan XP.

### 3. Configuring NetMan Clients

This chapter describes how to make NetMan available on your client PCs. This is the shortest chapter in the entire “Base Module” manual, because the procedure is so simple.

#### *Client Installation*

You do not need to install anything on the client PC. NetMan does not require any resources from the client. The only point that must be mentioned here is that NetMan records its own window coordinates in the client’s Registry. You can delete this data at any time.



#### Note

*NetMan registers itself automatically. If your users do not have permission to start self-registering programs, all you have to do is start NetMan one time on each workstation under an administrative account.*

The NetMan Client does not require server drive mapping. It can be started directly in the network environment (UNC: \\servername\sharename).

Users and stations are automatically detected by NetMan—you do not have to set these up, either. For more information you might want to refer to chapter 6, “Users, Stations, Groups and Profiles” and, for details on the mode of station detection, see “NetMan Settings: USER ID/STATION ID DIALOG PAGE” in chapter 4.7. For details about system security, see the sections entitled “NetMan Directories and Network Rights” and “Defining NetMan Administrators” in chapter 4.

There are two different ways to start NetMan and access NetMan applications calls:

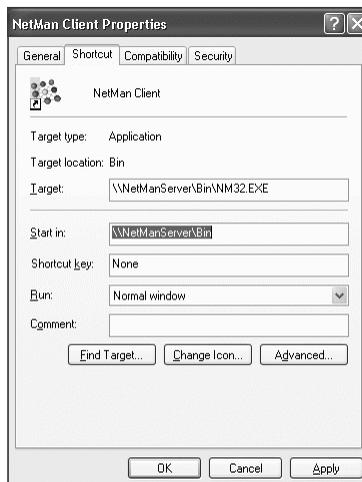
- Start the NetMan Client from the *NetMan desktop*
- Use *NetMan command line call*.

## Opening the NetMan Desktop

To open the NetMan desktop, you need to set up a shortcut from your workstation to the NetMan Client program, *NM32.exe*:

```
<NetMan root
directory>\bin\NM32.exe
```

The root directory is the installation directory.



### Note

*Make sure the NetMan working directory <root directory>\bin is entered correctly. If you create the shortcut in the Explorer using “drag & drop”, the path is automatically entered correctly; if you use the Windows desktop shortcut menu, it is not entered.*

You can also enter drive letter rather than the UNC path to start NetMan.

New users automatically inherit the *Public Desktop*.

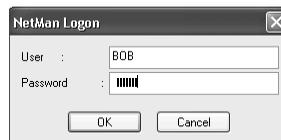


### Note

*You can change the name of the Public Desktop if desired, but it remains the default desktop for all users. You can also configure desktops for user-specific and station-specific startup configurations.*

To force a NetMan login dialog, enter the command

```
<root directory>\Bin\NM32.exe /logon
```

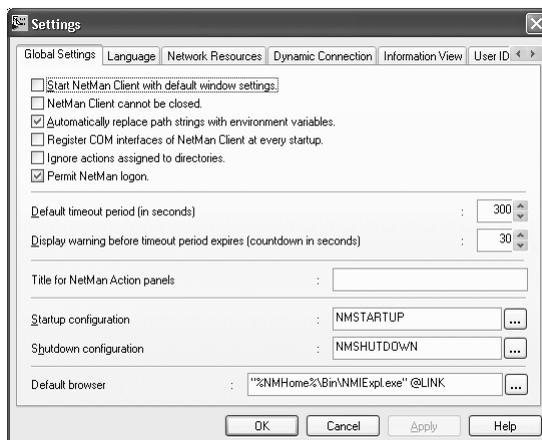




## Note

The NetMan logon dialog can be a practical alternative if the network logon names otherwise used are not relevant or meaningful in the NetMan environment. This may be the case if, for example, you have grouped together several users who work in one department by giving them a single user name under NetMan, or if you create a user name for anonymous users, for purposes of statistical evaluation. For further details on this topic, see chapter 6.

The user name entered in the NetMan logon dialog is not authenticated by the network. This is why the logon dialog only works for users that already exist in the NetMan user database; otherwise, any typing mistake made when entering a user name would result in the creation of a new user account in the NetMan database. Furthermore, logon under a NetMan user name must be specifically permitted in the NetMan Settings.



## NetMan Command Line Call

You can use the *NetMan command line call* to launch an individual NetMan application call, also known as a “NetMan configuration”, when the NetMan desktop is not displayed. This is a useful option if you want to start an application from another interface (for example, from an HTML page) but do not want to miss out on NetMan functionality, such as license management and event logging.

The NetMan command line program, *NMCmd32.exe*, is stored in the working directory of your NetMan installation.

The application to be started is passed to this program as a command line argument:

```
<root directory>\Bin\NMCmd32.exe /ID:<configuration ID>
```



#### Note

*For more information on the NetMan command line program, see “Discussion: The Trace Monitor and the Internal Architecture of NetMan” in chapter 5.*

Enter the command

```
...\bin\NMCmd32.exe /ID:<configuration> /logon
```

to force a NetMan login dialog. Only users who already exist in the NetMan database can use this dialog to log in.

You can also add an argument,

```
...\bin\NMCmd32.exe /ID:<configuration ID> /  
logon:<username>
```

to open the logon dialog box a specific NetMan user account. As described above, this function will only work if the user already exists in the NetMan user database.

## 4. Global NetMan Configurations

This chapter introduces the NetMan Client and describes how you can configure the settings to best suit your working environment and to ensure the system security you require.

### *The NetMan Client*

The NetMan Client is the starting program for NetMan in a Windows environment. From the NetMan Client, you can start NetMan system programs as well as any other program you configure here. The system programs and other applications you can start from NetMan are part of a *NetMan Desktop*. Your initial NetMan installation comes with a pre-configured desktop that you can edit and add to as desired: the *Public Desktop*.

When you start the NetMan Client, the *Public Desktop* is loaded. You can browse through the applications you see here in the same manner as in the Microsoft Windows Explorer. The functions for the view, status and program information, as well as Help and exit functions, are the same as those under Windows.

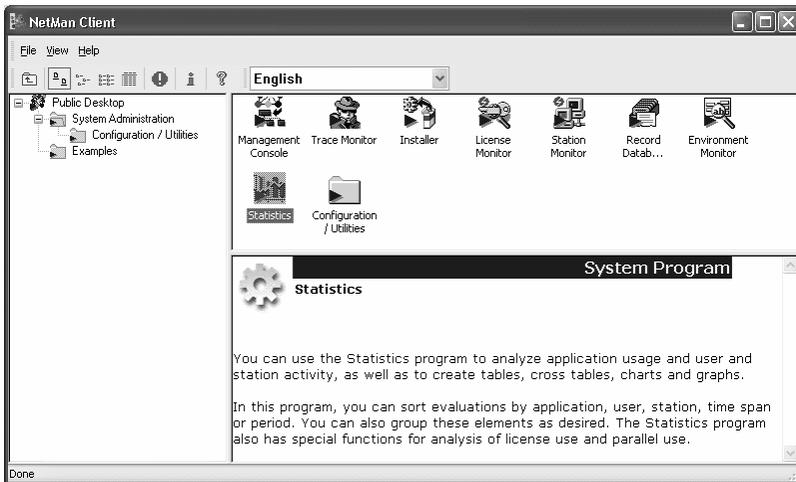
The left-hand pane of the NetMan Client window shows the elements that are linked to NetMan; this is the *structure display*. The initial NetMan installation includes 2 folders under PUBLIC DESKTOP: SYSTEM ADMINISTRATION, which contains NetMan system programs and components, and YOUR APPLICATIONS, for your programs and hyperlinks. The upper right-hand pane of the NetMan Client window, the *folder display*, shows the contents of the folder opened in the structure display. Double-click on a folder entry to activate it. If the entry is a NetMan *configuration*, containing an application call or a hyperlink, it is launched when you activate it; if it is another folder, it becomes your current directory.



#### Note

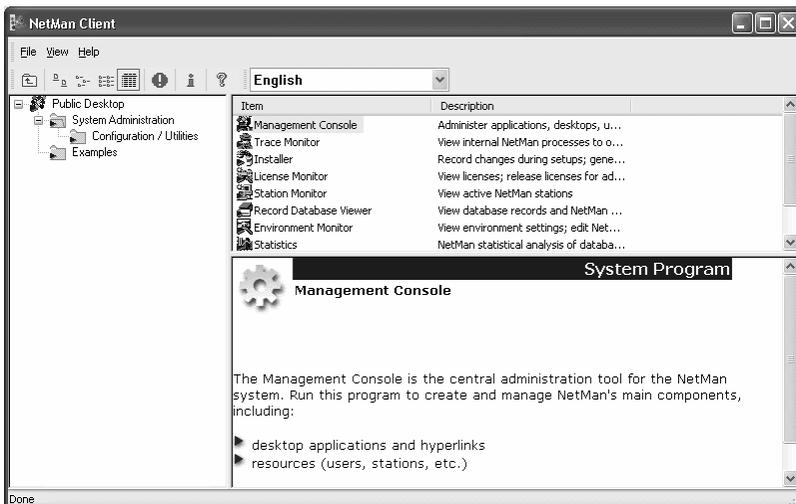
*For a detailed explanation of the terms application, folder, configuration, program, hyperlink, etc., as they are used in this manual, see chapter 5.*

You can have the elements in the folder display shown with large symbols:



Use the buttons in the toolbar or the commands in the VIEW menu to change the display; for instance, to show

- small symbols:
- a list:
- or details:

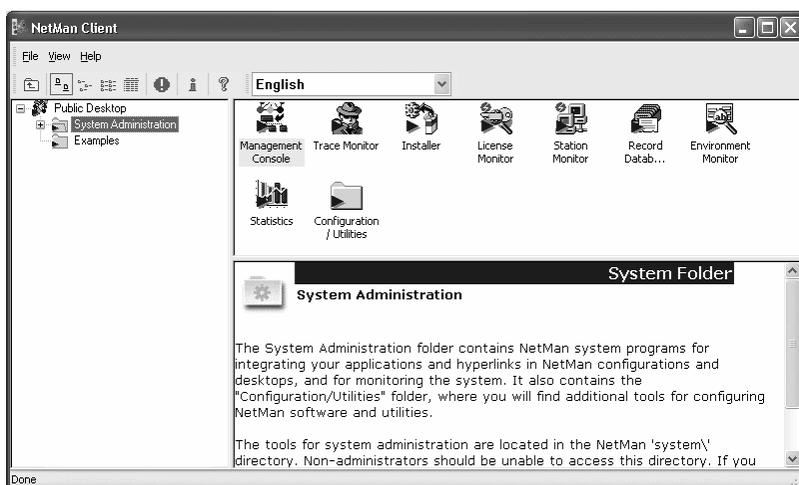


As you have probably already noticed, the *information display* in the lower right-hand pane shows information about the element selected in the folder display; this text is written in HTML.

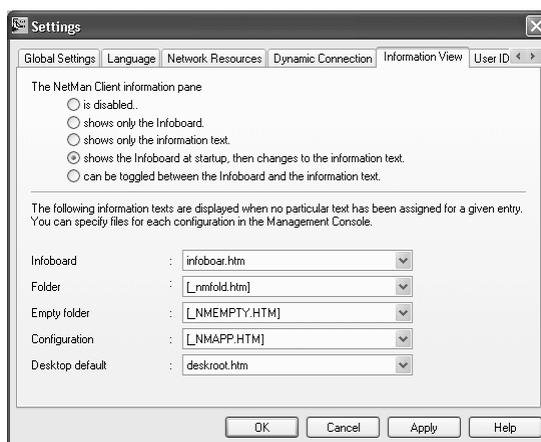
You can switch off the information display, if desired:



Alternatively, you can have an *Infoboard* displayed here:



You can define the contents of the information display on the INFORMATION VIEW dialog page in the NetMan Settings program, which is located under PUBLIC DESKTOP/ SYSTEM ADMINISTRATION/ CONFIGURATION/ UTILITIES/ NETMAN SETTINGS:



## Note

Your NetMan installation comes with an HTML document that is defined as an info-board. If you want to offer an info-board in your system, we recommend creating your own info-board containing the information you want to provide for your NetMan users. To do this, create an .HTM file and assign it on the Information View page under "Info-board." If the information display is configured here to show both the info-board and information files on individual elements, you can activate the info-board display by clicking on the corresponding button in the toolbar as shown here:



## ***NetMan Components***

The following describes the main NetMan programs, which are linked to the Public Desktop in the SYSTEM ADMINISTRATION folder when you first NetMan:

- ***Management Console:***  
This is the central NetMan administration program. From the Management Console you can administer NetMan desktops and all NetMan configurations (NetMan application calls), as well as resources (users, stations, etc.).
- ***License Monitor:***  
This program keeps track of usage of those applications for which you specify licensing functions. It shows the number of application licenses available, whether anyone is waiting in the queue for a license, which users and stations are currently using licenses, and who is waiting in the queue. You can also use this program to release individual licenses.
- ***Station Monitor:***  
This monitor shows you which workstations are currently using NetMan. You can also call up a view of the programs currently running on a given station.
- ***Record Database Viewer:***  
This display shows the event log and the sequential log file as well as the cumulative log file, in which a record of application use is updated monthly.
- ***Statistics:***  
The Statistics program evaluates the runtime data collected on application usage to give you an overview of how applications were used in a given time period; the data collected can be sorted by user or workstation.
- ***Trace Monitor:***  
The Trace Monitor watches over system and program sequences and can help you trace and solve problems.
- ***Environment Monitor:***  
This program lets you view the system's environment variables as well as the current NetMan environment in the client. It also allows you to set, change and delete variables.
- ***Settings / Utilities:***  
This folder contains NetMan settings programs and helper programs.

## ***NetMan Directories and Network Rights***

The standard NetMan setup program installs the following four subdirectories in the NetMan root directory:

- System
- Bin
- Prot
- User

NetMan system administrators must have unrestricted rights in the NetMan root directory.

NetMan users need certain permissions in the following directories:

- **Bin:** contains executable system programs and system data; users should have 'read' permission.
- **Prot:** contains log files as well as user and station databases; users should have 'read' permission in this directory, plus 'write' permission in the event log file (NMLog.DBF)
- **User:** this is the user directory; users should have 'read', 'write', 'create' and 'delete' permissions here.
- **System:** standard users should not be assigned any rights in this directory, as it contains system programs that should be used only by NetMan system administrators.

The following tables summarize the rights required by Microsoft networks and NetWare:

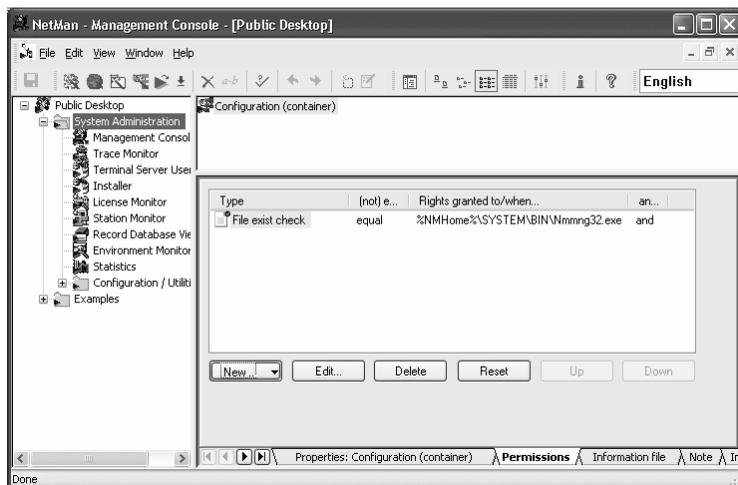
Microsoft Network			
User	User	Service Account	
DIRECTORIES/FILES	bin	Read / Execute	Read / Execute
	prot	No Access	Read / Write
	prot /nmlog.dbf	Read / Write	Read / Write
	user	Full Access	No Access
	system	No Access	No Access

Novell Network			
User	User	Service Account	
DIRECTORIES/FILES	bin	Read / Filescan	Read / Filescan
	prot	No Access	Read / Write / Filescan
	prot /nmlog.dbf	Read / Write / Filescan	Read / Write / Filescan
	user	Full Access	Full Access
	system	No Access	No Access

## Defining NetMan Administrators

The default entries in the SYSTEM ADMINISTRATION folder are for the most part program calls for NetMan system administration components. To safeguard the NetMan system, these components should only be accessible to NetMan administrators. To give you a demonstration of how you can hide folders from unauthorized users, we have assigned access permission to the SYSTEM ADMINISTRATION folder based on a *File Exist Check* condition, which is a NetMan function for defining the executability of actions based on conditions such as, in this case, the existence or visibility of a certain file. In our example, the system checks for the `nmmng32.exe` before opening the folder:



If you have assigned network rights as described above in the section entitled „NetMan Directory Structure and Network Rights“, then this configuration will effectively hide the *system* directory from your standard users, because they do not have the required permissions at the network level. In other words, NetMan users who have not been defined within the network as NetMan administrators will not see the SYSTEM ADMINISTRATION folder in their PUBLIC DESKTOP.

You can define this option as best suits your environment. Select NEW... on the PERMISSIONS dialog page to define other conditions; for example, to assign rights based on Novell, NT or LDAP group membership. You can also create a NetMan user group (under MANAGEMENT CONSOLE / RESOURCES) for NetMan Administrators.

## Configuring Drives and Shares

NetMan uses variables to address logical drives, paths and system states. For example, the variable for the NetMan root directory is *NMHome*.

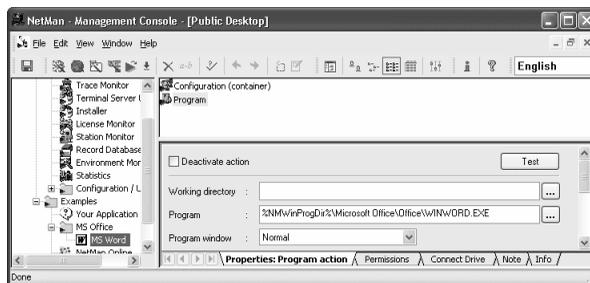
If you want to know exactly why we recommend using variables, read the following Discussion of the NetMan environment:



## Discussion

We recommend using environment variables wherever possible, because a system that consistently uses variables is easier to manage. For example, you can make changes that affect the entire system by adapting one little element: the variable.

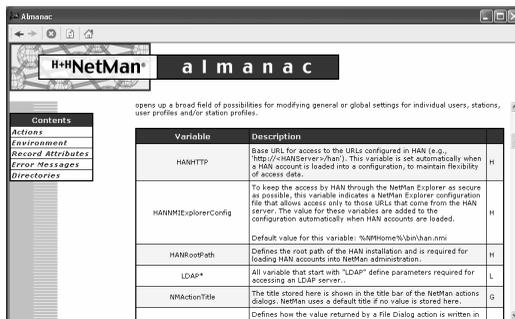
This of course makes system management highly flexible. For example, you might choose to define specific startup routines for individual users or stations (or groups of users or stations). In such cases it is practical to enter the *NMWinProgDir* variable, rather than an explicit path, to call MS Word for Windows in a NetMan configuration:



This gives you a certain independence when defining your configurations. The program call defined above will start Word whether the MS Office package is located in c:\programme, d:\program files, another directory altogether, or even on different drives for different clients.

Some variables can be set by a user activity, which means you can use variables for dynamic administration of system states. With the NetMan Language Module installed, for example, you can use a variable to hide the list box for changing the language in selected NetMan client systems.

For a complete list of NetMan variables, including detailed descriptions, please see the NetMan Almanac in the Management Console:

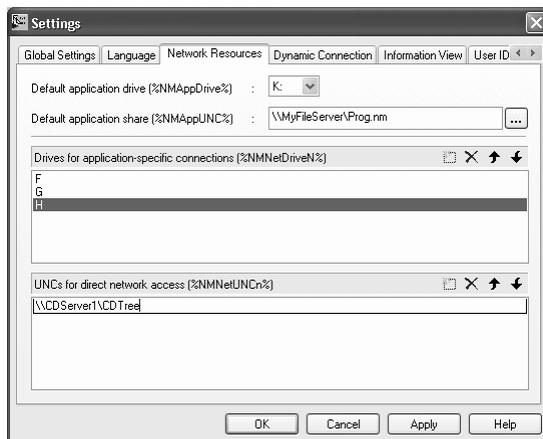


## Application Drive

This is the drive in which applications and programs are installed.

In network systems, applications are often installed in one or more designated application directories. Some applications require that a specific drive letter be entered when the application is installed; for example, if the application cannot use UNC paths.

In the NetMan Settings you can set variables for both an application drive, *NMAppDrive*, and a corresponding UNC path, *NMAppUNC*, defining the drive in which the applications you want to start under NetMan are installed. When you configure a NetMan application call, just enter the corresponding variable rather than a fixed designation. You can DEFINE VARIABLES on the NETWORK RESOURCES dialog page of the *NetMan Settings* (PUBLIC DESKTOP / SYSTEM ADMINISTRATION / CONFIGURATION / UTILITIES / NETMAN SETTINGS / NETWORK RESOURCES).



### Note

The NetMan application drive is mapped automatically when you start NetMan, if you define the *NMAppDrive* and *NMAppUNC* variables in the NetMan Settings program.



### Tip

With 32-bit applications, the paths and working drives are usually entered using UNC syntax. We recommend mapping the application drive as a root drive, i.e. a separate drive exclusively for applications, if 16-bit (DOS) applications are used in your system. (With most DOS applications you can enter only a drive letter, not a UNC path, as the target for installation.) The software is subsequently installed automatically in a subdirectory directly below the root directory on the specified drive.

## Commonly Used Network Resources

It is important to know the location of the network resources that are used most frequently, and whether you have applications that require a volume or a network share mapped to a drive letter, as is often the case in CD-ROM networks.

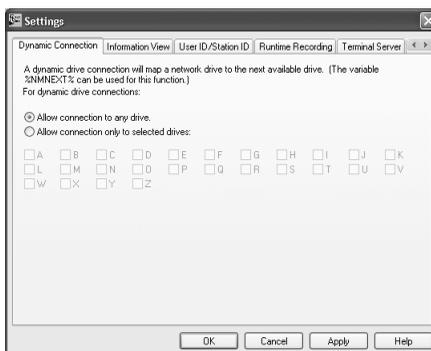


### Tip

Set the `NMNetUNCn` variable to map the resources you need most. Use variables in your configuration, such as `%NMNetUNC1%`, `%NMNetUNC2%`, etc., rather than fixed UNC paths. You can define an `NMNetDriveN` variable for mapping application-specific drive designations and then address `%NMNetDrive1%`, `%NMNetDrive2%`, etc., rather than entering fixed drive letters.

You can set a special function for dynamic drive mapping on the DYNAMIC CONNECTION dialog page in the NetMan Settings program:

Another option is to use the `NMNext` variable to map a drive letter (in place of `NMNetDriveN` or a fixed drive letter). In this case, NetMan assigns the first available drive letter. In the standard setup, all drive letters are allowed. If you only want to allow certain drive letters for this function, you can define them here.



### Note

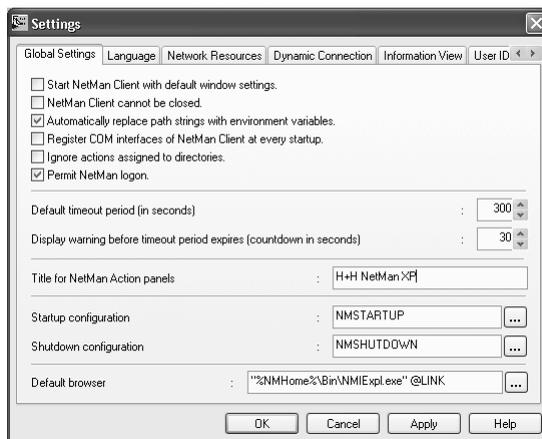
Resources with application-specific data can be dynamically mapped to drive letters only if this is allowed by the application. In many cases, however, the drive letter that contained the resources when the application was installed is registered, and this is the drive letter that is required. See “Notes on Special Types of Applications / CD-ROM Applications” in chapter 5 for details on the best uses of the `NMNext` variable.

## NetMan Startup and Shutdown

All of the elements linked in NetMan are defined in the *NetMan Management Console* as *configurations* (for more information about the structure of NetMan and how to use the Management Console, see chapter 5; for specific information about startup and shutdown configurations, see the section in chapter 5 entitled “Startup and Shutdown Configurations”).

You can define configurations specifically to be processed when NetMan is started or shut down on a client machine. These configurations can have any of a number of functions, including starting programs, mapping drives, logging the user onto the network, or setting environment variables. In short, you can embed practically any NetMan function you choose in a startup or shutdown configuration.

Start the *NetMan Settings* program from the PUBLIC DESKTOP / SYSTEM ADMINISTRATION / CONFIGURATION / UTILITIES FOLDER AND CLICK ON THE GLOBAL SETTINGS tab. *NMStartup* and *NMShutdown* are the default startup and shutdown configurations:

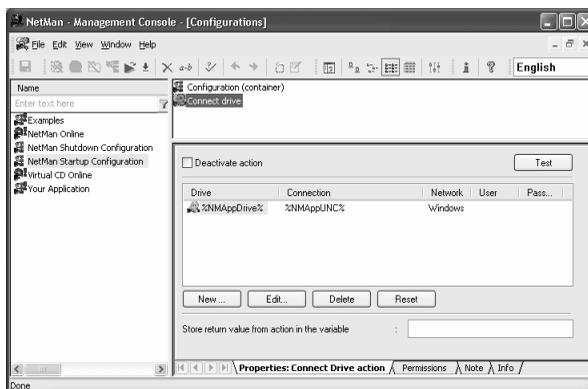


You can edit global startup and shutdown configurations in the NetMan Management Console.



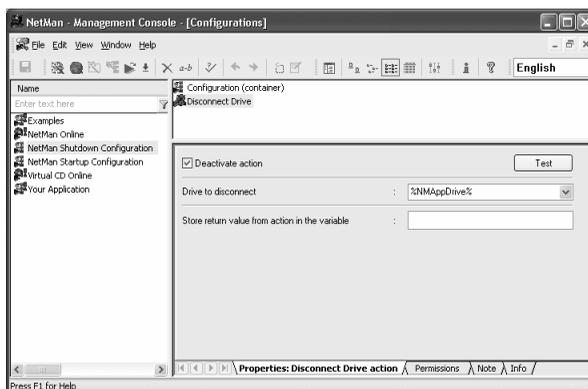
## Note

In the window shown here, the *SELECTION BAR* and the *CATEGORY* column are hidden.



The default *NetMan startup configuration* includes a *Connect Drive* action that maps the application drive. The variables used here are configured in the Settings program (see the section above entitled „Application Drive“).

You can have this drive disconnected automatically (in other words, you can undo the drive mapping) in the *NetMan shutdown configuration*:



## NetMan Settings

The modifications involved in the continuous design of your NetMan Client desktops, as well as in the integration of applications in NetMan, are for the most part made in the central configuration database and in desktop files, edited in the *NetMan Management Console* (PUBLIC DESKTOP / SYSTEM ADMINISTRATION / MANAGEMENT CONSOLE).

Some of the most important global settings for the NetMan system, on the other hand, are configured in the *NetMan Settings* program, located in PUBLIC DESKTOP / SYSTEM ADMINISTRATION / CONFIGURATION / UTILITIES.

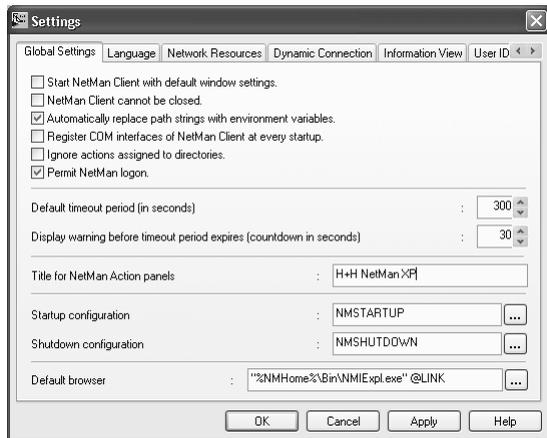
All of the settings relevant to the NetMan Base Module are described in detail below. If you are reading through this manual for the first time and have not used NetMan much yet, you might want to skip this section for now, and come back to it when you are more familiar with the program.



### Note

The illustrations below depicting the individual dialog pages show the default settings that are active when you first install NetMan.

### GLOBAL SETTINGS Dialog Page



**START NETMAN CLIENT WITH DEFAULT WINDOW SETTINGS:** This option is not active in the initial configuration, which means that NetMan saves the current screen

coordinates in the client's Registry when the program is exited, and opens the program window with these coordinates when the program is started. When this option is activated, however, the program always opens a window in the upper left-hand corner of the screen (position 0.0, 640, 480). This option is especially useful in environments with public PCs and when working with terminal servers. It also eliminates problems that can arise when NetMan Client windows are shifted, for example if the Client is started on workstations with different screen resolutions.



### Tip

*Some system administrators prefer not to have this option available. To disable the option, simply add a user-specific Environment action (see chapter 5) to the startup configuration, to set the variable: NMIgnorePosition=0, or define separate startup configurations for users or user profiles.*

- **NETMAN CLIENT CANNOT BE CLOSED:**  
Activating this option sets the NMDenyCloseNM32 variable to "1", in which case your users cannot shut down the NetMan Client. You can define an Environment action to go with this setting, to set the same variable to "0" for certain users. Alternatively, you can use an Environment action to set the value to "1" for a specific user or station profile, rather than using this global setting.
- **AUTOMATICALLY REPLACE PATH STRINGS WITH ENVIRONMENT VARIABLES:** This option is active in the initial configuration, which means that in the various configuration fields, NetMan automatically replaces path names or parts of path-names that it recognizes as values defined in variables (see the **ADVANCED** dialog page and refer to the Almanac). To deactivate this option, remove the checkmark by clicking in the box.



### Note

*If you want to know exactly why we recommend using variables, read the following Discussion of the NetMan environment: "Configuring Drives and Shares."*

- **REGISTER COM INTERFACES OF NETMAN CLIENT AT EVERY STARTUP:**  
NetMan registers itself automatically every time it is started. This can lead to error messages if the user starting NetMan does not have permission to write in the corresponding Registry key. Select this option to stop these error messages from occurring. Keep in mind that NetMan has to be registered

correctly one time (e.g., started under a user account with sufficient permissions to register the COM interface).

- **IGNORE ACTIONS ASSIGNED TO DIRECTORIES:**  
NetMan lets you define any number of actions to be processed whenever a folder is opened. If you select this option to disable folder-specific actions, this can improve the performance in the NetMan Client, as the program no longer checks for the existence of such actions before opening a folder.
- **PERMIT NETMAN LOGON:**  
When this option is enabled, users can logon to NetMan under accounts other than their normal network accounts. With this setting, the NetMan start program accepts the command line argument /Logon:<username>. The NetMan user name must be defined beforehand in the NetMan database (“Resources”). This mechanism can be used to assign a NetMan user name, based on client IP address or host name, for anonymous users in a terminal server environment.
- **DEFAULT TIMEOUT PERIOD (IN SECONDS):** This field defines the “timeout” delay, in seconds. If an application is left unused for a period of this length, it shuts down automatically. The value entered here is the default value in the “timeout” field when a new Program action (see chapter 5) is created.
- **DISPLAY WARNING BEFORE TIMEOUT PERIOD EXPIRES (COUNTDOWN IN SECONDS):** This function acts as a reminder to the user. When the timeout function is running, this “countdown” window is opened before the timeout delay has elapsed (in the example shown here, 10 seconds before the end of the timeout), to inform the user that the application will shut down automatically if it is not used again before the displayed time has elapsed.
- **TITLE FOR NETMAN ACTION PANELS:** The text entered here is used in the title bar of dialog windows generated by NetMan.
- **STARTUP CONFIGURATION:** This is the NetMan configuration processed when NetMan is started. The function of this setting is described under “NetMan Startup and Shutdown.”
- **SHUTDOWN CONFIGURATION:** This is the NetMan configuration processed when NetMan is shut down.
- **DEFAULT BROWSER:** This is the browser used when a hyperlink is started from NetMan. This browser is entered in the NetMan Explorer as the default browser. If no browser is specified here, the browser defined as the default for the workstation is started when a hyperlink is activated.

### ***LANGUAGE Dialog Page***

This dialog page lets you define the language(s) available for NetMan administrators and users.

Both English and German are available for NetMan system programs at all times. In user programs, only the languages you install during setup are available. Multi-lingual options are available for users only if the NetMan Language Module is installed and registered. Refer to the Language Module manual for a detailed description of the language controls available in that module.

## NETWORK RESOURCES AND DYNAMIC CONNECTION Dialog Pages

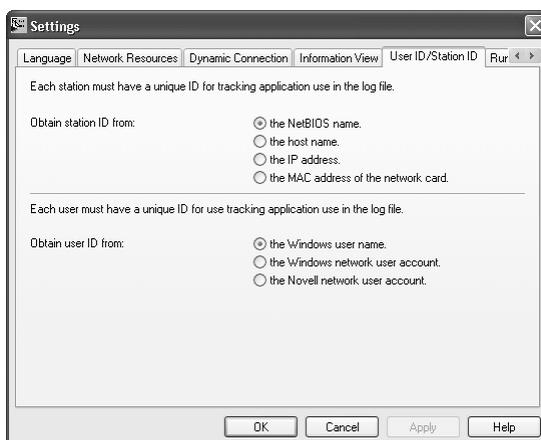
See the section above entitled “Configuring Drives and Shares” for a detailed description of the settings available here.

### INFORMATION VIEW Dialog Page

See the section entitled “The NetMan Client” at the beginning of this chapter for details on the settings available here.

### User ID/Station ID Dialog Page

NetMan automatically records user and station IDs in a database. On this dialog page you can select the manner in which these IDs are defined:



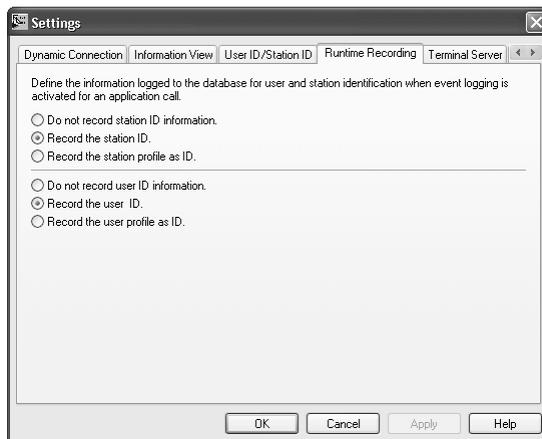
### Note

*For station IDs it is important that you choose an identification method that will always be available in your environment and will always provide a unique name. For instance, if you select “the host name” for identification, make sure that all IP addresses of all potential NetMan clients are entered in the DNS server.*

*It is also important to keep in mind that, if you are using “Named Sites”, changing the method of station identification during system operation will at first result in all stations being counted twice. See chapter 6 for more information on this topic.*

## *RUNTIME RECORDING Dialog Page*

With the default settings, NetMan logs the client's user ID and station ID when an application is started. On this dialog page, you can choose different user-specific and station-specific identification methods.



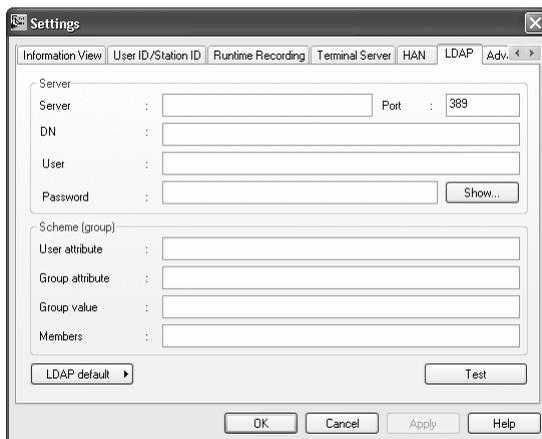
- **DO NOT RECORD STATION ID INFORMATION:** When applications are called from a NetMan Client, the log entry for the calling station is left blank if this option is selected.
- **RECORD THE STATION ID:** With this option selected, the calling station ID is recorded in the log file when an application is called.
- **RECORD THE STATION PROFILE:** With this option selected, the profile of the client station is recorded in the log file when an application is called.
- **DO NOT RECORD USER ID INFORMATION:** When applications are called from a NetMan Client, the log entry for the calling user is left blank if this option is selected.
- **RECORD THE USER ID:** With this option selected, the ID of the calling user ID is recorded in the log file when an application is called.
- **RECORD THE USER PROFILE:** With this option selected, the profile of the user is recorded in the log file when an application is called.

## LDAP Dialog Page

The LDAP page lets you define the access NetMan uses to read and check LDAP privileges. If you do not plan to work with LDAP access permissions, you can ignore this page.

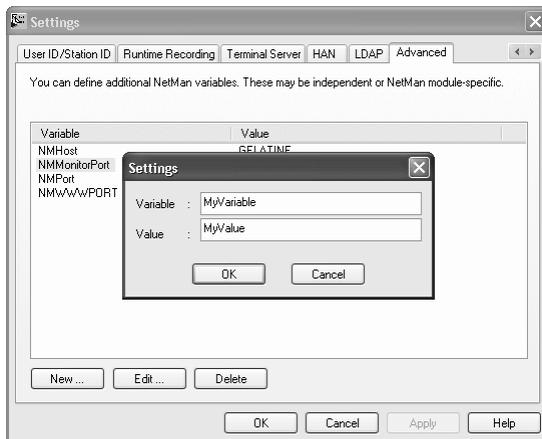
Enter the default settings for the Microsoft and Netscape LDAP server under LDAP default. If the defaults required are not standard, check with the administrator of the LDAP server for the data you need to enter here.

The Test function opens a dialog that connects to the LDAP server using the values entered and displays a list in accordance with the settings. If the values shown here are not correct, then the data entered for LDAP access was incorrect.



## ADVANCED Dialog Page

On this page you can create and define NetMan environment variables.



You can set values here for the variables you integrate in your NetMan configurations.

## 5. Integrating Applications and Hyperlinks

This chapter describes how to integrate applications, on-line access accounts, URLs, and HTML documents in your NetMan system for distribution to your users.

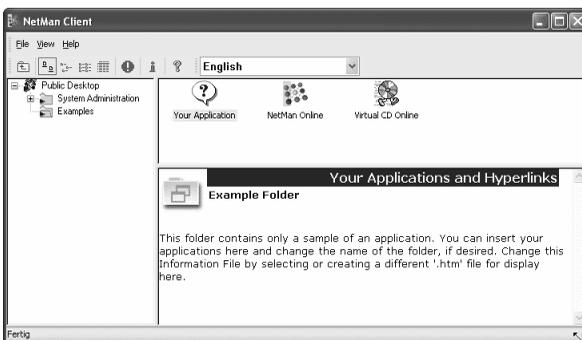
NetMan is accessed by users either through the NetMan Client or from an HTML document. This HTML document can be generated by the *HTML View Module* (or the simplified version of this module, the *HTML Wizard* included with your NetMan Base Module), or you can write it yourself. This chapter describes only the access over NetMan Client, and is divided into the following sections:

- The first section explains some of the basic concepts of the NetMan software, as well as the terminology used.
- The second section, called “First Steps with the Management Console”, introduces the Management Console, your central system program, and describes its operating elements. Examples are given to show you how additional program and hyperlink properties can be activated in your applications. You will also learn how to create your own desktop entries and how to integrate your applications and hyperlinks in the NetMan system.
- The third section presents a closer look at NetMan’s technical implementation of the functions described in the previous sections.
- The fourth section contains specific information on integrating CD-ROM applications.
- The fifth section illustrates the many uses of NetMan actions and provides a number of practical examples.
- The last section describes the procedure for importing data from the NetMan HAN Module.

## The Two Types of NetMan Configuration

As you will have noticed by now, the terms “application,” “hyperlink,” “program” and “NetMan configuration” come up frequently in this manual. The examples below provide a more precise definition of these terms as they are used in the context of NetMan, to give you a better understanding of how NetMan works.

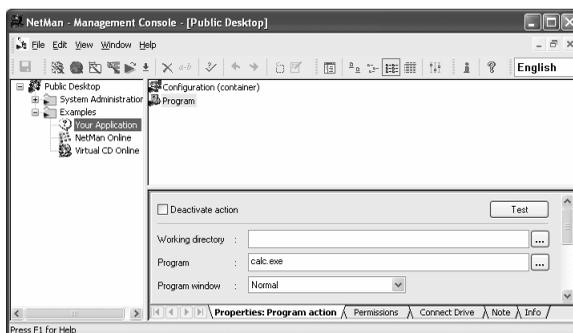
Start the NetMan Client and open the EXAMPLES folder; here you will find a number of desktop entries that have already been configured for you:



All of the entries in a desktop are referred to as *NetMan configurations*. In fact, there are two types of configuration:

1. Container configurations
2. Hyperlink configurations

*Container configurations* contain a number of Windows-based “actions,” which are linked to the Windows operating system and generally have no counterpart within the World Wide Web (WWW). These configurations can be executed (the actions sequentially processed) only on a Windows computer. If a Container configuration is activated by a client running a different operating system, such as Linux or Macintosh, a Windows terminal server is required for processing the actions. In most cases, a NetMan Container configuration starts a program.



For demonstration purposes, we have integrated the Windows „Calculator“ program („Calc.exe“) in this sample entry; we think we can safely assume that this program already exists in your Windows directory.

The question is: Is “Calc.exe” an application or a program?

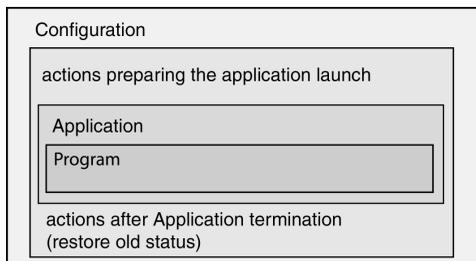
We are of the opinion that the Windows Calculator is both a *program* and an *application*, but this is an exceptional case. As a rule, an application consists of more than just a program. For example, Word is referred to as a “Microsoft Office application” because the program itself (in this case, “winword.exe”) requires a number of other specific files and directories to be present before it can run. Thus the term “application” indicates both a program and an array of other elements.

The term *configuration*, as used in the context of the NetMan software, is even broader; it refers to a completely user-definable logical unit, created by a NetMan administrator. A “NetMan configuration” is like an empty container that you can fill up with ‘execute’ jobs, which NetMan processes in sequence. Hence the term “Container configuration.” An individual job is referred to here as an *action*. In our example, the configuration called YOUR APPLICATION contains only one action; this is a “Program action” configured to call the Calc.exe program.

Let us assume that you did not install the Accessories package, which contains the Calculator application, when you installed Windows on your client workstations. To make the Calculator available to your users during run time, all you have to do is add your choice of actions to the NetMan configuration for Calc.exe; these actions could copy the program, map a drive to the program, or log the user onto the network. You can also add another action, positioned after the Program action, for example to undo the drive mapping or logoff the user. Think about the possibilities this opens up in connection with a “real” application, such as Microsoft’s LexiROM. You can configure a wide range of operations to take place before or after the program is started. You could, for example:

- provide access to a program, or to program components, such as data on CD, through network login or resource mapping.
- create DLL files or Registry entries on the client workstation.
- open a dialog box for password input, or other user input which is then passed to the program on the command line.
- start other programs.
- restore the previous working environment when the program is ended.

The following diagram illustrates the relationship between *program*, *application* and NetMan *Container configuration*:



With the most basic programs, the NetMan configuration will not contain any preparatory or closing actions; the only action is the program call, as is the case in our Calc.exe example.

In many cases, integrating an application in NetMan will consist of no more than two steps: first you create a **configuration**, then you add a single **action** containing the command that starts the application. The number and variety of actions available, however, give you a wide range of possibilities for your configurations. Processing a NetMan Container configuration is like executing a script, because you can define conditions under which any individual action will—or will not—run. Conditions for running an action are defined in the form of ‘execute’ permissions, which can be granted on the basis of user name, station designation, group membership, environment variables, operating system, or any of a number of other factors. Thanks to NetMan’s interface to *Windows Script* (formerly known as the Windows Scripting Host), you can even create your own NetMan actions.

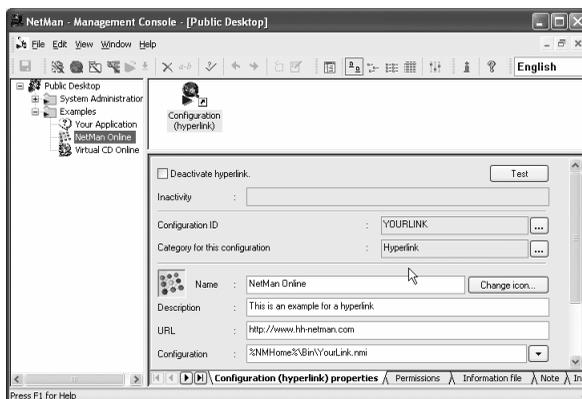
**Thus a Container configuration** is a logical unit that can be executed by a user; it can contain up to 999 **actions**, or none at all. A **program action** starts an **application** that is integrated in a NetMan configuration.



## Note

*Whenever this manual mentions launching a NetMan application call, whether from the NetMan Client, the HTML Client or the command line, it means that the processing of a NetMan configuration is activated. While the Program action contains the command that starts the application, the NetMan configuration can contain practically any number of other actions, which are processed either before or after the Program action.*

A **Hyperlink configuration**, on the other hand, loads an HTML document over HTTP. When a Hyperlink configuration is activated, the NetMan Client starts the NetMan Explorer. The NetMan Explorer is based on the Microsoft Internet Explorer, but allows additional settings affecting the navigational options available to the user. The sample configuration here loads the home page of the NetMan Web server.



Hyperlink configurations are integrated in NetMan's user interfaces in the same manner as folders and Container configurations, and share many of the same properties:

- Like other configurations, they have both a name and a description.
- Their availability can be restricted to specified users, stations, groups, or other identifying characteristics, such as IP address or domain.
- Activation of the configuration can be logged in the NetMan database.
- They can be deactivated.

In addition, Hyperlink configurations have two unique properties:

- The URL field contains the URL that the browser will be directed to.
- In the CONFIGURATION field (input optional) you can specify a configuration file which defines the permitted and restricted URLs, as well as the user controls available in the NetMan Explorer.

The NetMan user interfaces handle Hyperlink configurations as follows:

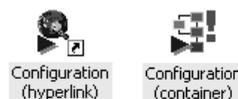
- The NetMan Client starts the default browser (here, the NetMan Explorer (see also chapter 9, "NetMan Explorer")) and loads the URL.
- NetMan HTML View executes the link to the URL.
- NetMan HTML Wizard enters the link in the pages generated.



#### Note

*If you use NetMan start files with your HTML View or HTML Wizard (refer to the corresponding manual or section for details), the HTML View or HTML Wizard program starts a hyperlink as a NetMan configuration using the NetMan command line program.*

The *type of a configuration* is indicated in the Management Console by the symbol shown and the specification (HYPERLINK) or (CONTAINER):



The first time you start the NetMan Client after installation, the *Public Desktop* is loaded automatically. A NetMan desktop is a structured view of your NetMan configurations. NetMan Container configurations can function as folders. Here you see the SYSTEM ADMINISTRATION and EXAMPLES folders in the left-hand pane, called the „folder display.“ The SYSTEM ADMINISTRATION folder contains NetMan system programs, and the EXAMPLES folder has a Container configuration called YOUR APPLICATION, as well as two Hyperlink configurations, called NETMAN ONLINE and VIRTUAL CD ONLINE.

The layout of this desktop can be compared to that of Windows Explorer:

In the *Windows Explorer* you navigate through *folders* to get to *files*; in the *NetMan Desktop* you navigate through *folders* to start *applications* or execute *hyperlinks*.

All of the entries in the NetMan desktop are stored in NetMan databases as *configurations*, as defined above (with the exception of the root entry at the top). The desktop that appears when the NetMan Client is first started has the hierarchical structure that we defined; you can adapt this structure to your own preferences.

Container configurations can be added to a desktop to act as folders. A container used as a folder can also contain your choice of NetMan actions; these are processed when the folder is opened in the NetMan Client. When a NetMan desktop is presented in an HTML document, any ‘folder’ actions are ignored.

The *EXAMPLES* entry is a configuration that does not have any actions, and is defined as a folder. When you double-click on this folder, it opens and shows the elements it contains. The elements within a folder are again either Container configurations (*applications* or *folders*) or hyperlink configurations.

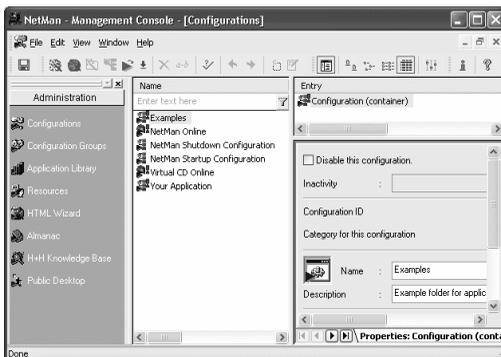
## First Steps with the Management Console

The *Management Console*, located in the *SYSTEM ADMINISTRATION* folder, is the main system program used for integrating applications and hyperlinks in NetMan. We will begin our instructions for taking your “first steps” by introducing the Management Console and its operating elements.

In addition to the usual menus and tool bars, familiar to you from other Windows software, the Management Console has a *selection bar*.

You can hide the selection bar to have more space for the desktop, if desired. When you click on a symbol in the selection bar, a window opens with the corresponding data. We will describe each of the symbols briefly first, before moving on to a detailed discussion of the Public Desktop.

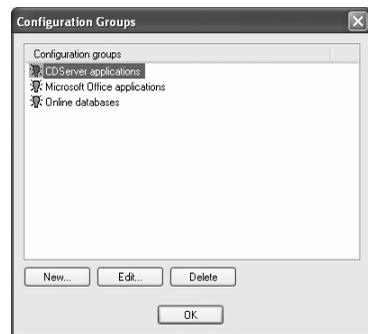
The *Configurations* symbol opens a window showing all of your NetMan configurations. A NetMan desktop, by contrast, contains only those configurations which you explicitly assign to that desktop, for a specific user or set of users.



A red exclamation point on a configuration symbol indicates that the configuration is assigned to at least one desktop. Configurations cannot be deleted as long as they belong to a desktop.

The right-hand pane of the Configurations window gives you the same editing functions that are available in the *Desktop Editor* (= Public Desktop), described below. The main difference between the desktop editor and the configuration editor is the comprehensive listing of all configurations. This means you can edit configurations in this window even if they are not included in any desktop. This is often the case, for example, with Startup and Shutdown configurations.

The *Configuration Groups* symbol opens a window listing your configuration groups. You can activate and deactivate the groups here. A configuration in a deactivated group cannot be launched by users.



The NetMan *Application Library* is a wizard that you can use to add „ready-made“ configurations to an existing NetMan installation. These configurations are available in compressed form in NetMan updates and service packs, as well as from the Internet (for example, from the H +H knowledge base or the Web site of your NetMan vendor). For details, please see chapter 10, under „The Application Library“.

The *Resources* window lets you view and edit *users, stations, user groups and profiles* and *station groups and profiles*. The NetMan RESOURCES are described in detail in chapter 6 of this manual.

The *NetMan Almanac* is part of NetMan’s on-line Help, and contains descriptions of NetMan actions, variables, error messages and directories, as well as other information.

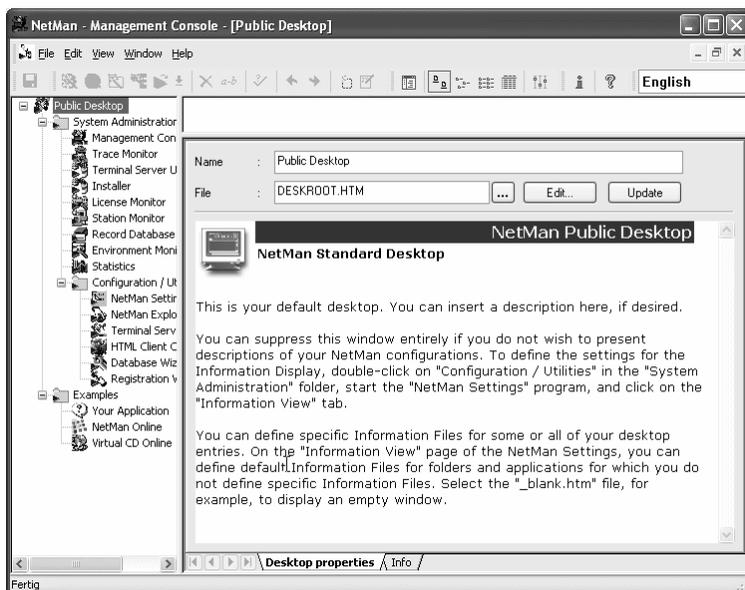
The *H+H Knowledge Base* symbol connects you to the H+H Knowledge Base on our NetMan Web server. This knowledge base contains additional information about the program, as well as tips and tricks for working with NetMan.

The *Public Desktop* is explained in detail in the following section.

## The Public Desktop

When you activate this element, the NetMan Client *Public Desktop* is opened in editing mode. The data contained here defines your configurations centrally; any changes you make here are effective globally; in other words, they affect the NetMan Client interface for all of your users.

The window below shows the fully expanded desktop structure, with the selection bar hidden. The active element in the folder view in this example is the root entry. Since the root entry is not a configuration and does not contain any entries or actions, the upper pane on the right is empty. The lower right-hand pane shows the DESKTOP PROPERTIES and INFO dialog pages. The INFO page always shows information on the entry selected in the upper right-hand pane; when the desktop root is selected, as in this example, the information shown applies to the Desktop Editor itself.

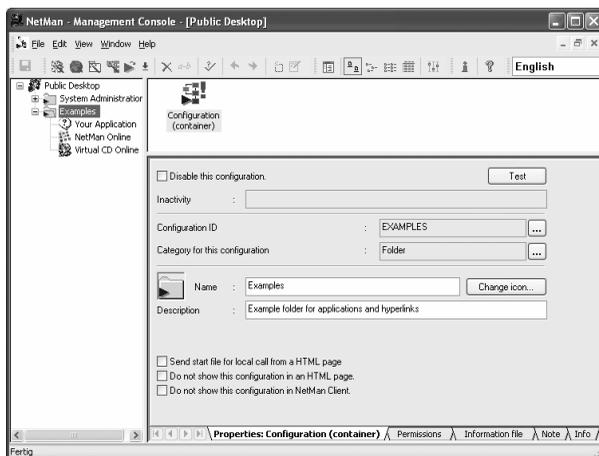


### Tip

When you first start working with NetMan, it is a good idea to read the INFO pages that go with each of the entry types.

Below the root entry, you see all of the configurations integrated in the desktop. Container configurations can also function as folders within a desktop.

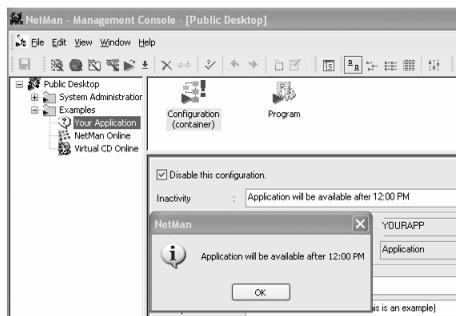
Let us take a closer look at a “folder” entry. Expand the Samples folder; your screen should show the following:



The Entry pane on the upper right shows one „Configuration,“ with its properties shown in the pane below. Since this configuration is used as a folder, no actions are defined for it. If you do not change the symbol, the default folder symbol is shown in both the NetMan Client and the Desktop Editor.

You can edit the name and description of the configuration on the Properties page. Your changes take effect after you have saved the new version.

If you select the `DISABLE THIS CONFIGURATION` option, the configuration will still be visible in the NetMan Client, but if a user tries to activate it, a message tells the user that the configuration is currently inactive. A default message is generated automatically; you can also create your own message text, if desired. This can save you from being asked repeatedly by various users why the application is not working. For example:





The following options let you specify the context for processing configurations:

- Send start file for local call from an HTML page
- Do not show this configuration in an HTML page
- DO NOT SHOW THIS CONFIGURATION IN NETMAN CLIENT

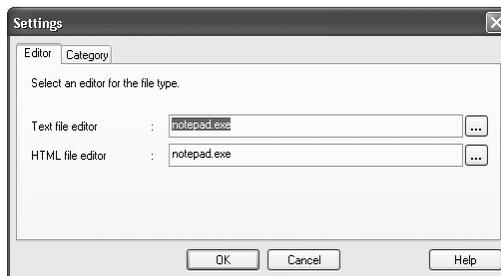
For example, the default settings for the SYSTEM ADMINISTRATION folder specify that this folder is not shown in HTML pages. The option SEND START FILE FOR LOCAL CALL FROM AN HTML PAGE, on the other hand, is useful if your applications are generally started under MetaFrame, but you have one particular application that can run on any client station: when you select this option, the application in question is always launched on the local worked station, never on the terminal server, which conserves resources and licenses.

Each configuration has the following dialog pages:

- PROPERTIES: CONFIGURATION,
- PERMISSIONS,
- information file,
- NOTE, and
- the INFO page described above.

We will take a closer look at the PERMISSIONS page later. In the current example nothing has been entered on that page, which means that any NetMan user can access this configuration and can see and open this entry as a desktop folder.

Click on the INFORMATION FILE tab to view the information page for this configuration. Because you have opened the Public Desktop from the Management Console, the desktop is in editing mode and you can edit the information text, if desired. The default editor for this is "Notepad.exe." To use a different editor, select SETTINGS... from the VIEW menu and enter the command line call for the desired program:

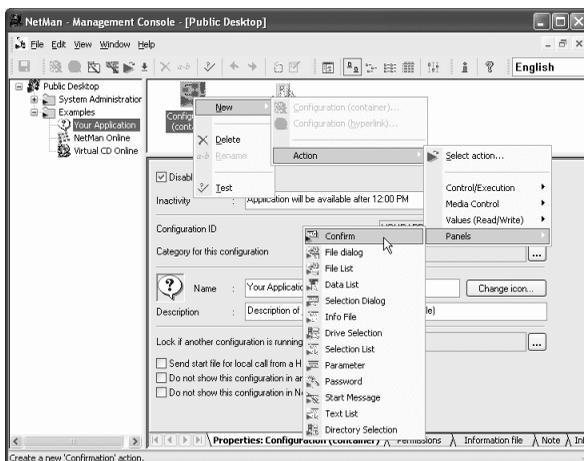


**Tip**

Click on the INFO tab and read the text about the "#Configuration" entry.

The “Note” page presents an editable field in which you can enter comments relevant to use of the configuration, such as a description of its functions or information on the application it starts (such as licensing codes or other special requirements).

If desired, you can add actions to a configuration that is defined as a folder (rather than as an application). To do so, position the mouse cursor in the Entry pane and right-click to open a list of actions to choose from:



If you add a **Password** action, for example, the folder will still be visible to all your users, but can be opened only if the correct password is entered, and the user will be prompted for the password every time the desktop is loaded or refreshed.

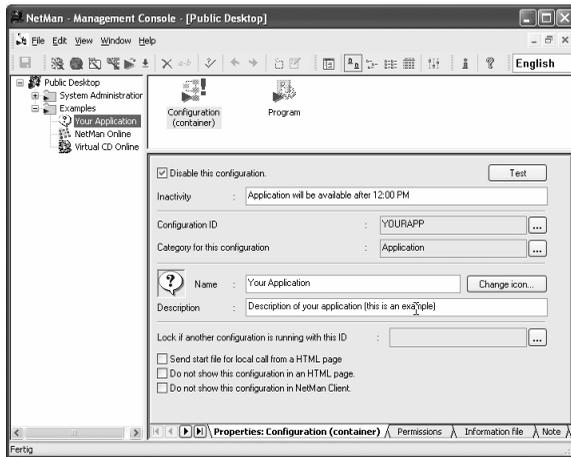


## Note

*Actions assigned to folders are ignored in NetMan’s HTML interfaces (HTML Wizard and HTML View).*

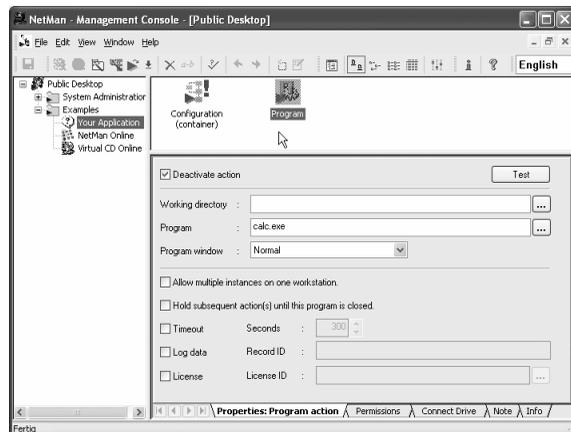
## Program Actions

Now let us look at a desktop entry defined as an *application*. Select the YOUR APPLICATION configuration, in the EXAMPLES folder. This configuration contains only a single Program action:



An „application“ configuration, as opposed to a „folder“ configuration, has another option on the Properties dialog page: the field called ID TO BLOCK CONFIGURATION IF ANOTHER IS RUNNING WITH THIS ID. This option can be used to prevent incompatible applications from being launched on a single workstation. Applications that have the same ID entered here will be mutually blocked. In other words, if a user is running an application that has an ID entered here (referred to as a „Block ID“ for short), they will not be able to launch any other application call that has the same Block ID while the first application is running. This can be useful if you have applications (or separate instances of the same application) that interfere with one another. For example, one application might try to access data that another application locks during use, or an application might be internally designed to run in only a single instance on a given machine.

Now select the Program action in the Entry pane, click on the INFO tab and read the information about this action.



The *Program* action has the following properties:

- **WORKING DIRECTORY:** NetMan will start the program from the directory entered here.
- **PROGRAM:** The program to be executed is entered here.
- **START AS:** You can select the mode in which the program window is opened (normal, maximized, or minimized).
- **ALLOW MULTIPLE INSTANCES ON ONE WORKSTATION:** Defines whether more than one instance of this program can run at one time on a given workstation. With this option activated, NetMan permits users to start an unlimited number of instances of this program.
- **HOLD SUBSEQUENT ACTION(S) UNTIL THIS PROGRAM IS CLOSED:** Before you decide whether to activate this option, think about the definition of a NetMan configuration, given at the beginning of this chapter: a user-defined sequence of almost any number of actions. With this option selected, the subsequent actions (following this Program action) will not be executed until the user exits the activated program; otherwise, any actions in this configuration that come after this Program action will be executed as soon as this program has been launched.
- **TIMEOUT:** Select this option to define a period of time after which the program will be shut down if it has not been used. The default number of seconds is defined in the NetMan Settings (see “NetMan Settings” in chapter 4) and can be overwritten here. This option is particularly useful for applications with a limited number of user licenses. The timeout option may not work with all programs, however; this depends in part on the way a given program works. You cannot assign a timeout for a DOS program, for example.
- **LOG DATA:** With this option selected, entries are written in event logs when the program is started and when it is closed, so you have a record of the program running time. How events are logged is defined in the NetMan Settings. The “Record ID” you define here is included in the log file entries.
- **LICENSE:** Activate this option to limit the number of workstations that can run this program simultaneously. You can create a new license ID or assign an existing ID to this Program action.

Note that the Program action also has a Connect Drive dialog page. Here you can map a drive designation to the network resource required by the program.

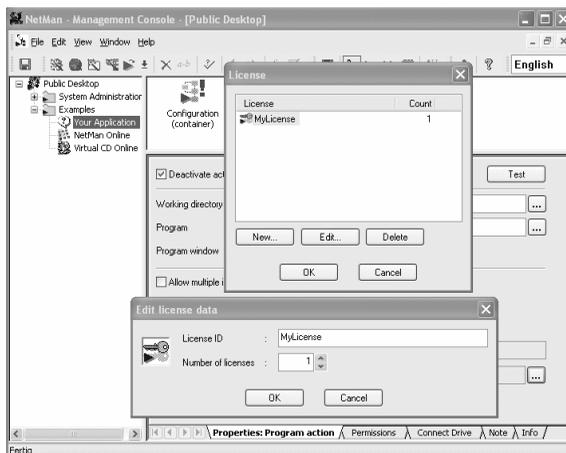
### ***Additional Program Properties***

Now you know enough to take your own first steps. In the following, you will activate three of the additional program properties available in NetMan:

- timeout,
- event logging, and
- licensing.

In the dialog box below, these properties have been activated. Click on the button to

the right of the “License ID” field to open a dialog box for creating, deleting and assigning licenses.



### Note

*The number of licenses for a given application is not stored directly in this configuration. This means that you can assign the same license to more than one configuration. You may wish to do so, for instance, if different NetMan configurations have to share a single license.*

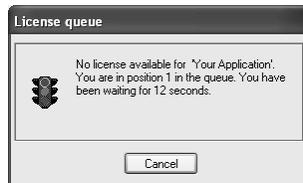
The settings configured here are effective in the NetMan Client as soon as they are saved. You can test your changes before saving the settings; the Test function is available in the toolbar, in the EDIT menu and in the shortcut menu in the Entry pane. If only one action is selected when you activate the Test function, the action alone is tested; if you select ‘Configuration’ at the top of the Entry list, the entire sequence is tested.



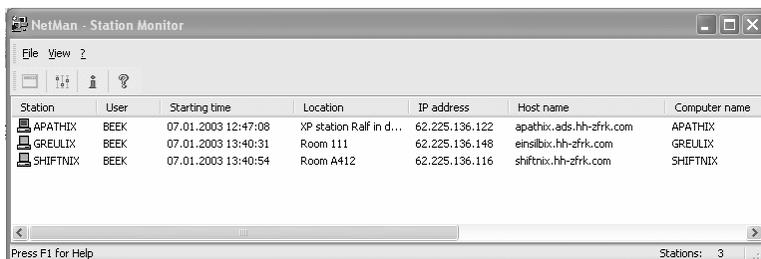
### Note

*Testing a licensed application from within the Management Console does not reduce the number of licenses available for actual users.*

Now we will start Your Application on three different workstations. The following message is displayed on the second workstation:



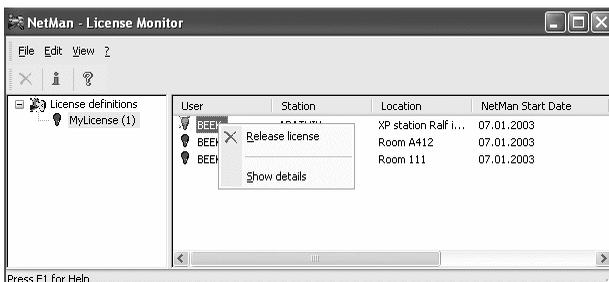
The message displayed on the third workstation indicates that the user is second in line. The next step is to call the *Station Monitor* from the System Administration folder and see what is going on with our three workstations:



## Note

*The example above does not show all of the available information. For example, this monitor can also show the name of the logged program. To specify which items are included here, select **SETTINGS** from the **VIEW** menu. This manual gives only a few examples of the operating features available in system programs. For more detailed information, please refer to the on-line help.*

You can call the **License Monitor** to see which licensed applications are in use, and to release licenses if desired. In this example, you can release the license for the Calculator, in which case the user who had been first in line (the second workstation) can start the program right away:



## Note

*If all the licenses for a given application are in use and you release a license for another user, this does not close the application on any workstation where it is already running. Thus releasing a license may result in a breach of the software licensing agreement for the application in question.*

To test the timeout function, wait until the defined delay has elapsed:



Once the timeout period has been reached on all three workstations, let us take a look at the **Record Database Viewer**:

	Protokoll-ID	Startdatum	Stopdatum	Startzeit	Stopzeit	Benutzerkennung	Stationskennung	Protokol
6	Taschnrechner	18.09.2002	18.09.2002	13:34:39	13:35:42	BEEK	APATHIX	/AS
5	Taschnrechner	18.09.2002	18.09.2002	13:32:51	13:32:51	BEEK	GREULIX	/NL3
4	Taschnrechner	18.09.2002	18.09.2002	13:32:45	13:34:20	BEEK	APATHIX	/AS/WL_20
3	Taschnrechner	18.09.2002	18.09.2002	13:31:56	13:32:43	BEEK	GREULIX	/WL_365
2	Taschnrechner	18.09.2002	18.09.2002	13:25:44	13:36:24	MEYER	GELATINE	/AS
1	Taschnrechner	18.09.2002	18.09.2002	13:19:31	13:21:51	BEEK	GREULIX	

Two of the events listed here show values in the „Record Attribute“ column indicating the number of seconds spent waiting for a license (WL); this attribute can be summa-

alized in the statistics program, by application and by time period, to get an idea of where „bottle-necks“ occur with licensed applications.

### ***Additional Properties of NetMan Hyperlinks***

The configuration described above is a Container configuration, which means it can contain any sequence of actions that can execute on a Windows workstation, regardless of its function in the NetMan Desktop (i.e., as a folder or an application). The other two example configurations, “NetMan Online” and “Virtual CD,” are Hyperlink configurations.

A Hyperlink cannot function as a folder, nor can it contain actions. On the other hand, it has two special properties of its own:

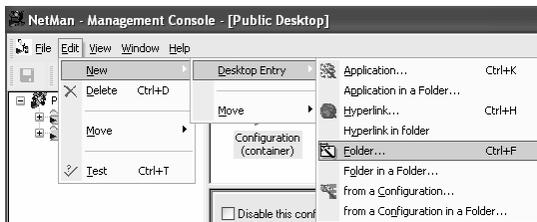
- Like a Program action, its launch can be recorded in the event log.
- If you use the NetMan Explorer, the Hyperlink configuration lets you define properties that control the user’s navigation options.

These two sample Hyperlinks are described in detail in chapter 9, under “Examples in the Public Desktop.”

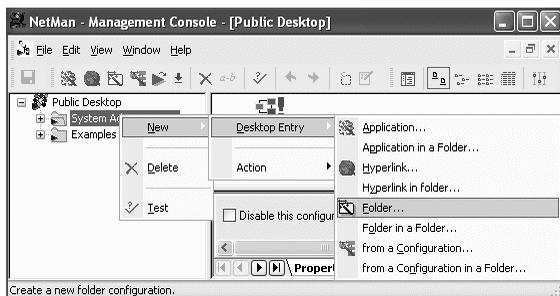
### ***Creating and Deleting Desktop Entries***

In the examples above, we added new program properties to the existing Calculator program. Now it is time to create your own desktop entries.

Just select NEW, either from the EDIT menu...



... or from the shortcut menu (right-click) in a desktop element.

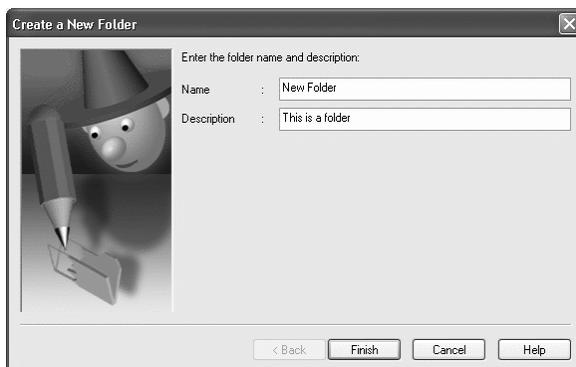


The menu for creating a new entry contains the following choices:

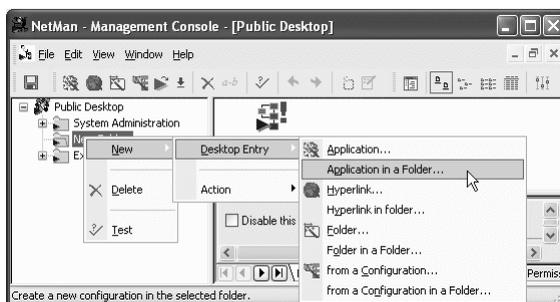
- **Application or Hyperlink:** Creates an “application/hyperlink” entry (below the YOUR APPLICATIONS folder).
- **Application or Hyperlink in a Folder:** Creates an “application/hyperlink” entry at the top of the list (under YOUR APPLICATIONS).
- **Folder:** Creates a “folder” entry (below the YOUR APPLICATIONS folder).
- **Folder in a Folder:** Creates a “folder” entry at the top of the list (under YOUR APPLICATIONS).

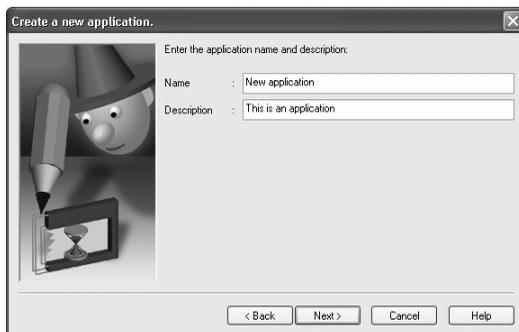
You can also click on the corresponding button in the tool bar to *Create a new application* or *Create a new folder*.

In the following example, we will create a folder called NEW FOLDER.

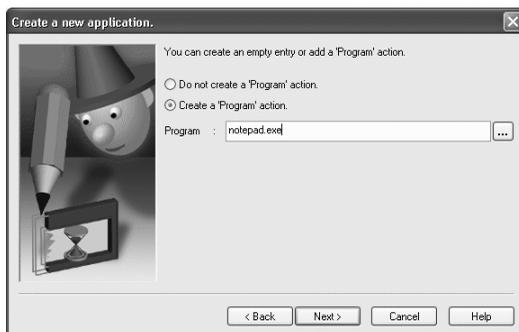


Here we enter a name and a brief description of the new folder and click on „Finish.“ Next, we create an entry within this folder; this time it is an „application“ entry:





Again, we enter a name (New Application) and a brief description; then we can go on to define a Program action for our new NetMan configuration, because it was created as an application, rather than a folder:



On the last page of this Wizard, you are asked to confirm (or edit) two entries which are automatically generated by NetMan:

- The ID of the new configuration (in this example, “New”)
- The category to which the new configuration is assigned (in this example, “Application”)

The *ID* of a configuration must be unique, because it is used to call the configuration; for example, from the command line, or as part of a URL.

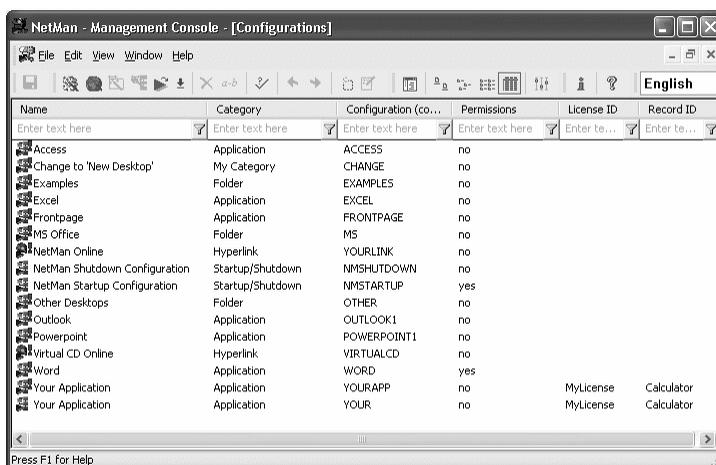


### Tip

*In many cases, it is important to modify the configuration ID to make it more meaningful. For example, if you name your configuration “MS Word,” The ID automatically*

generated by NetMan is “MS.” If you accept this ID and subsequently create a configuration named “MS Excel,” NetMan will generate the ID “MS2.” To modify the automatically generated ID, simply overwrite it on this dialog page.

The *Category* of a configuration is basically a sorting criterion. To illustrate this function, we open the list of all configurations (mentioned above) and double-click on the “Category” column to sort them:



The screenshot shows a window titled "NetMan - Management Console - [Configurations]". It contains a table with the following columns: Name, Category, Configuration (co...), Permissions, License ID, and Record ID. The table lists various configurations such as "ACCESS", "CHANGE to 'New Desktop'", "EXAMPLES", "EXCEL", "FRONTPAGE", "MS Office", "NetMan Online", "NetMan Shutdown Configuration", "NetMan Startup Configuration", "Other Desktops", "Outlook", "Powerpoint", "Virtual CD Online", "Word", "Your Application", and "Your Application". The "Permissions" column indicates whether certain actions are allowed (yes/no). The "License ID" and "Record ID" columns show specific identifiers for each configuration.

Name	Category	Configuration (co...)	Permissions	License ID	Record ID
ACCESS	Application	ACCESS	no		
CHANGE to 'New Desktop'	My Category	CHANGE	no		
EXAMPLES	Folder	EXAMPLES	no		
EXCEL	Application	EXCEL	no		
FRONTPAGE	Application	FRONTPAGE	no		
MS Office	Folder	MS	no		
NetMan Online	Hyperlink	YOURLINK	no		
NetMan Shutdown Configuration	Startup/Shutdown	NMSHUTDOWN	no		
NetMan Startup Configuration	Startup/Shutdown	NMSTARTUP	yes		
Other Desktops	Folder	OTHER	no		
Outlook	Application	OUTLOOK1	no		
Powerpoint	Application	POWERPOINT1	no		
Virtual CD Online	Hyperlink	VIRTUALCD	no		
Word	Application	WORD	yes		
Your Application	Application	YOURAPP	no	MyLicense	Calculator
Your Application	Application	YOUR	no	MyLicense	Calculator

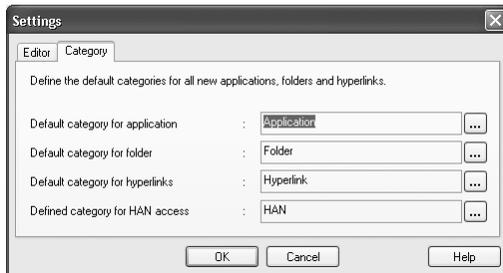
Keep in mind that this table can be very long, depending on how many configurations you have. The use of categories can help you to keep track of your configurations, and to find a particular configuration more easily.



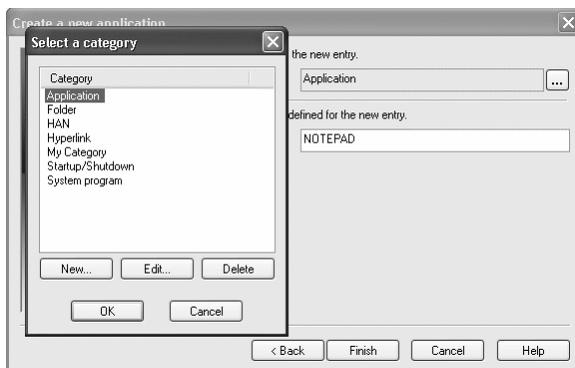
### Tip

The list of all configurations also shows at a glance whether ‘execute’ conditions, licenses and run-time recording are configured. In the line marked ENTER TEXT HERE you can set filters for the individual columns to reduce the number of entries shown.

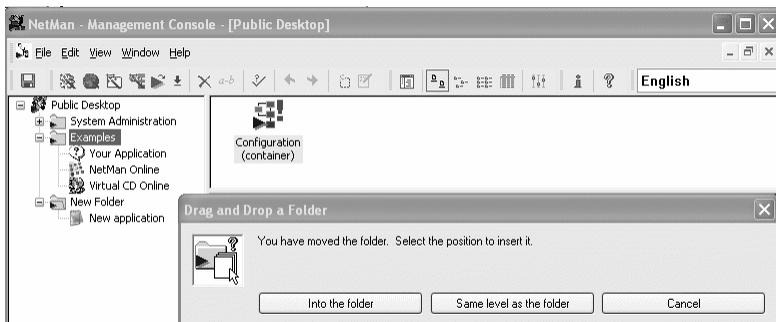
Let us return to our example for a “New application.” NetMan assigned the category “Application” automatically, based on a function that you can modify under VIEW / SETTINGS:



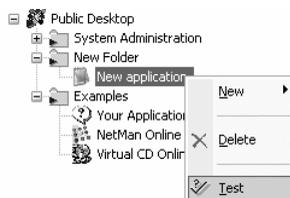
As you can see here, you can define your own categories and specify defaults. In our example, a new category called „My Category“ is assigned to the „NOTEPAD“ configuration:



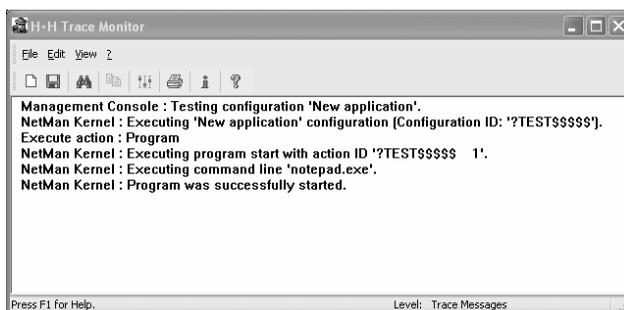
Bob takes over from here and completes the new configuration. Afterwards, however, we decide we would rather have the new folder positioned above the Your Applications folder, so we use the mouse to drag it up to the SYSTEM ADMINISTRATION folder. Now we have to make a choice:



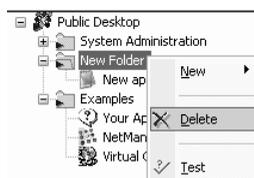
We choose SAME LEVEL AS THE FOLDER.



Now we run the Trace Monitor (located in the System Administration folder) and test the new application, and lo and behold, „Notepad.exe“ is started.



Since this was only an example, we can delete this folder now:



We are now asked whether we wish to delete the entry entirely, as opposed to simply removing it from the active desktop:

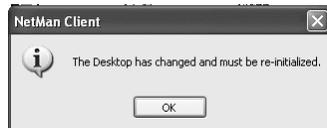


This question is always asked when you delete a configuration that is not assigned to any other desktop. If the entry is still linked in another desktop, it is simply removed from the active desktop when you select the „Delete“ command, without prompting for confirmation, and is still available in the list of all configurations.

In this example, we answer “Yes” since the entry was created only for demonstration purposes.

Next, we delete the pre-configured sample configurations, but answer “No” at the prompt, so that these configurations are merely removed from the active desktop, but remain in the list of configurations.

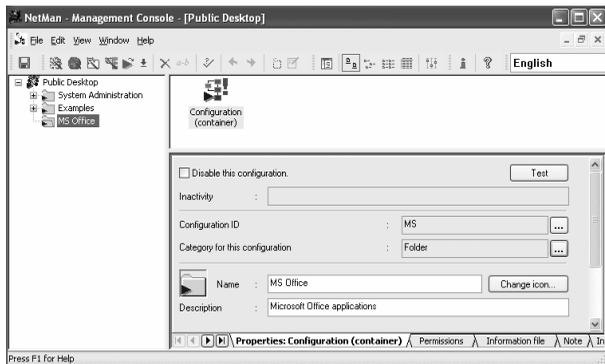
In the preceding steps, we made several changes in the desktop structure. If we save the changes now, or did so at any point during the steps performed above, any NetMan Client interfaces that were already running would have to refresh their desktops before using the desktop. If a user tries to activate an entry that is no longer available or no longer exists, the following message is displayed:



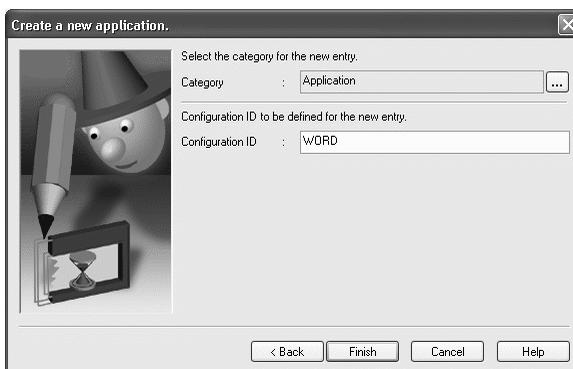
### *Your First Application*

Now we will show you how to integrate an application of your own in NetMan. For this demonstration, we will use the Microsoft Word application, which is already installed on the workstation in question. This example shows you the options for integrating the application in your NetMan databases; it does not deal with the topic of MS Word installation.

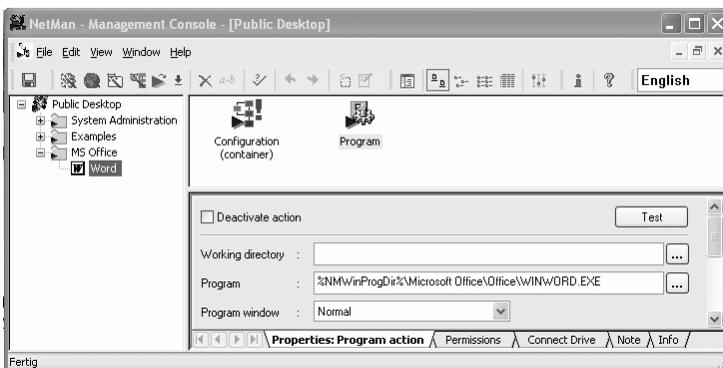
We begin by creating a folder called MS OFFICE:



Then we create a new „Application“ configuration as described in the previous section.



NetMan automatically extracts the icon from „Winword.exe“ to `NMHome\Bin\Data\Icons` and uses it as the symbol for this configuration.



## Note

*With the default global settings, NetMan automatically inserts environment variables in place of specific path designations whenever a path or part of a path is recognized. In this example, `C:\Program Files` is replaced by the `NMWinProgDir` variable. The advantage of this is that the program is found on each workstation, because it is always installed in the Windows directory on the local drive, whether the drive letter is `C:` or `D:` and no matter what the directory is called.*

Finally, we activate the licensing and event logging functions.

On the INFORMATION FILE page, you can create and assign a special HTML file for providing information to users, if desired.



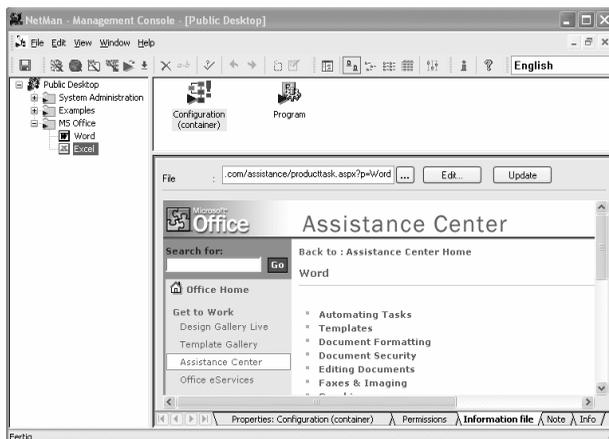
## Note

*It is not absolutely necessary to create an HTML information file. As an alternative, you could either have the information display show the Infoboard and maintain this as your information source, or simply hide the information display altogether. For details on these alternatives, see “The NetMan Client” in chapter 4.*

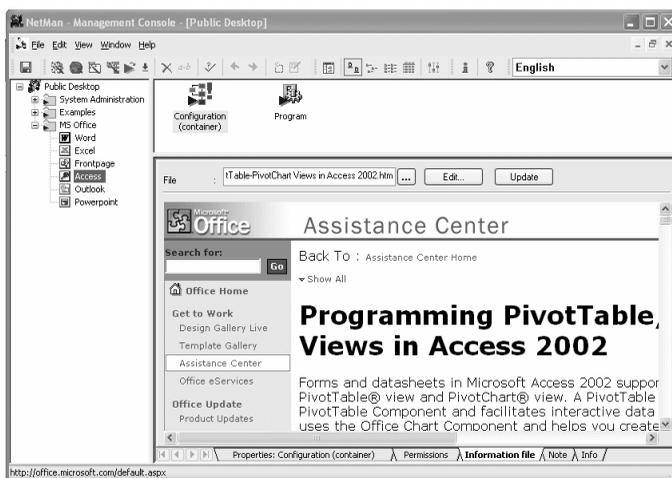


## Tip

*In some cases, you may find an information file in HTML format in the installation directory of the application, or on the Web site of the software manufacturer. In such cases, you can enter a URL here rather than specifying a file. In the window shown below, the information is provided by the “How to...” article on MS Word, from the Microsoft.com home page. The subsequent window shows an information file that was download from that site.*



NetMan can now be put into operation with your first application. If you were to integrate all Microsoft Office applications in the same manner, the NetMan Client would look something like this:



## Note

The NetMan Client desktop on your workstations may still show the System Administration folder, which contains the NetMan system programs. For information on hiding elements and restricting user access, see “NetMan Directories and Network Rights” and “Defining NetMan Administrators” in chapter 4.

Up to now we have described each step in great detail, because these were your “first steps” and because we wanted to acquaint you with the program’s internal logic. From this point onward we will be operating on the assumption that you know how to create, edit, delete and move desktop entries, and will provide details only on other aspects of NetMan operating elements.

### **Permission to Access Configurations and Actions**

You can restrict access to configurations and actions to specific users, user groups, user profiles, stations, station groups, station profiles, and/or and network groups. You can also grant or deny access permission based on any of a number of defined conditions.

For example, you can have a given configuration displayed only for members of a certain

- **global NT group** (primary domain controller required),
- **local NT group**,
- **LDAP group** (LDAP server required), or
- **NetWare group**.

This mechanism provides full support for the groups used in the most common network operating systems. You can use the rights structures that are already in place in your network without having to create new definitions within the NetMan system.

Since all of your user and workstation names are automatically copied into NetMan databases, you have the option of linking access rights not only to **user's network login names**, but also to **workstation names**, as well as **user and station groups and profiles**.

With this feature, NetMan closes a **gap in network operating systems** that assign permissions solely on the basis of user accounts.

Moreover, NetMan lets you control access to configurations according to specified **conditions as well** – another feature that takes you beyond the realm of conventional network capabilities. You can make configuration access dependent on the existence of one or more specified elements on the client machine, which can include the following:

- a **file**,
- a **path**,
- a **drive**,
- a **Registry entry**,
- an **INI file entry**, or
- a **value in an environment variable**.

Furthermore, you can choose to show or hide configurations based on the client workstation's

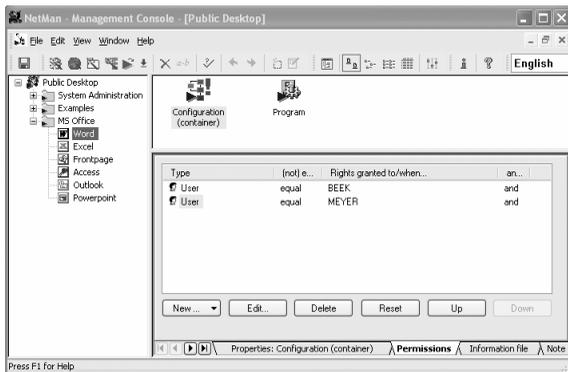
- **operating system**,
- **IP address**, or
- **host name**.

The variations on the rights structure can be used in any combination and linked with logical **ANDs** and **ORs**, and can be formulated in the positive or the negative. In the simplest cases, you grant 'execute' permission to

- **users**,
- **stations**,
- **local NT groups**,
- **global NT groups**, or
- **NetWare groups**.

Here is an example of an *invalid* assignment of permissions:

Select the configuration and click on the RIGHTS tab. Click on the NEW button and select NETMAN USER. Enter a second user to the list in the same manner as the first:



This definition, where the second user is linked by a logical *AND*, would make it impossible to launch this configuration.

The entries in the RIGHTS list are evaluated logically by NetMan: each entry is a proposition that is either true or false. The assignment of 'execute' rights for this configuration will depend on the truth value resulting from the evaluation of all entries in the RIGHTS list. The expression

User = "Beek" and User = "Meyer"

is always false (due to the *AND* operator), while the expression

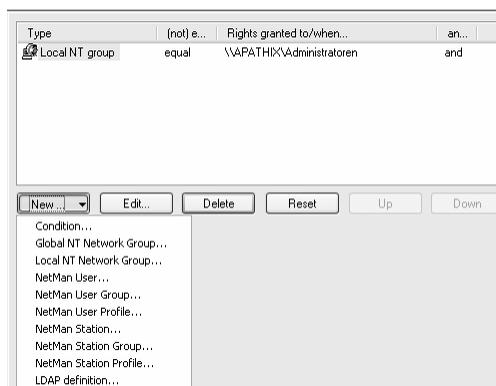
User = "Beek" or User = "Meyer"

is true whenever the user's name is either "Meyer" or "Beek" (logical *OR* in place of *AND*).

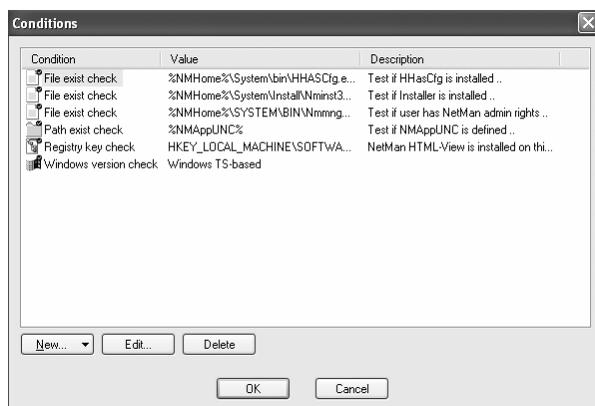
The next example illustrates a truly practical use of the *AND* link:

Say your program runs only on *Windows NT workstations*, but you want to make it available to administrators in a network that also has *Windows 98/ME* stations.

To do this, just link the 'execute' rights for the corresponding NetMan configuration to your domain administrators and then create a new *condition* for these rights:



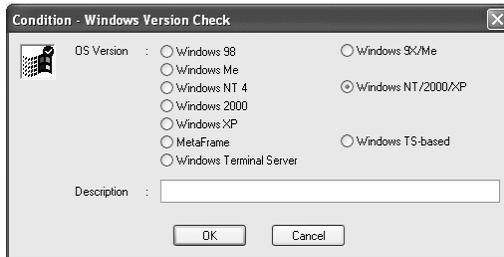
To do this, click on **NEW** and select **CONDITION**; this opens a list of conditions that you can choose from.



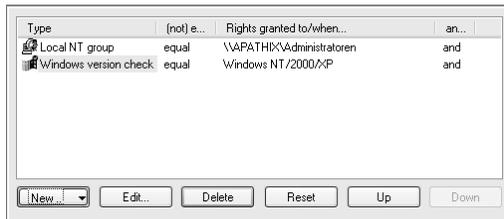
## Note

The conditions listed here were configured automatically by the NetMan installation program, to hide the names of system programs that are not installed and to hide the System Administration folder from users who do not have rights in this directory (see *Defining NetMan Administrators* in chapter 4). You can delete any of these entries that you do not need and, if desired, remove the desktop entries for system programs that are not installed.

Since the condition you require does not appear in this list, you have to create it. To do this, click on **NEW** and select **OPERATING SYSTEM CHECK**; in the next dialog box, select **WINDOWS NT/2000/XP**.



And that is all you have to do:



The other conditions you can configure are:

#### ***Environment Check***

Determines whether the workstation has a given NetMan or system variable.

#### ***Variable Check***

Determines whether a given action return value is equal to the value specified.

#### ***INI Entry Check***

Determines whether a given variable in a Windows INI file is set to the value specified. INI files are for the most part used by 16-bit Windows programs, while 32-bit Windows uses Registry entries (see below).

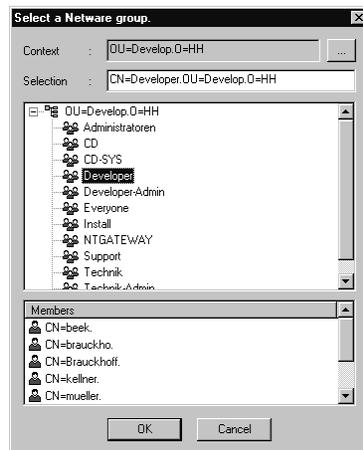
#### ***Registry Check***

Determines whether a given key in the Registry is set to the value specified.

#### ***Host Name or IP Address Check***

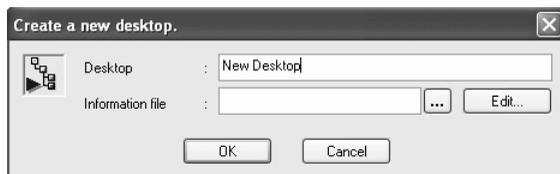
Determines whether the workstation host name or IP address matches a specified host name (wildcards permitted), IP address or range of addresses.

In conclusion, we must point out that NetWare Directory Services (NDS) can be accessed only if the IntraNetWare Client interface from Novell is installed on all workstations in your network:



## Creating Additional Desktops

To create new NetMan desktops, select FILE / CREATE DESKTOPS...:



The brand-new desktop is empty. You can add your choice of the following elements:

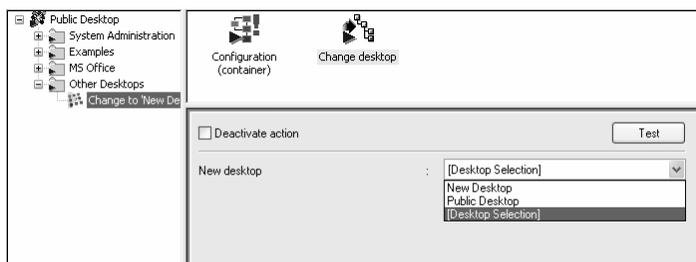
- New configurations (folders, applications, and hyperlinks). This involves creating new NetMan configurations.
- Existing configurations. This involves creating NetMan desktop entries that refer to the existing configurations.
- Desktop entries from other desktops. This involves using *Drag and Drop* to move or copy desktop entries. Whether an entry is moved or copied, and whether the configuration it refers to is duplicated or not, depends on whether you press the CTRL key, both the CTRL+SHIFT keys, or no key at all while dragging and dropping the element:
- Drag-and-drop without holding down a key: The desktop entry is moved; the configuration it refers to is unaffected (not duplicated).
- Drag-and-drop while holding the **CTRL** key: The desktop entry is *copied* and the configuration in question is duplicated.
- Drag-and-drop while holding the **CTRL** and **SHIFT** keys: The desktop entry is *copied* and the configuration in question is not copied. In this case, the desktop entry refers to the existing configuration; this is an efficient way to create entire folders.



### Tip

*If you want the same set of configurations available in several different desktops, we recommend that you **not** duplicate the configurations, as this would create extra work in configuration management. Rather, have the entries in the different desktops refer to a single instance of the configuration. When you do make copies of configurations, however, make to sure change the data entered in the licensing and event logging fields (on the **PROPERTIES** page of the Program action), unless you wish to have the use of the program recorded under the same name for all users, and/or run under the same license.*

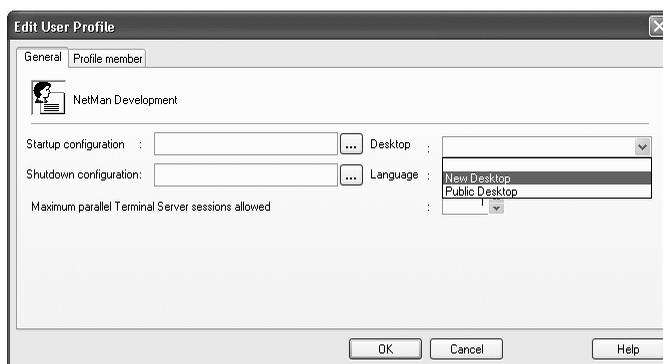
You can insert a *Change Desktop* action to load a desktop other than the default NetMan Client desktop. If no desktop is specified for the change, this action lets users select from among all existing desktops.



### Tip

*We strongly recommend creating a reference to the Change Desktop action (by copying the desired desktop entry without duplicating the action) for testing purposes, so you can change back to your original desktop at any time. This prevents you from “getting stuck” in the new desktop during testing. To prevent your users from changing to a particular desktop, assign ‘execute’ permissions within the Change Desktop action accordingly.*

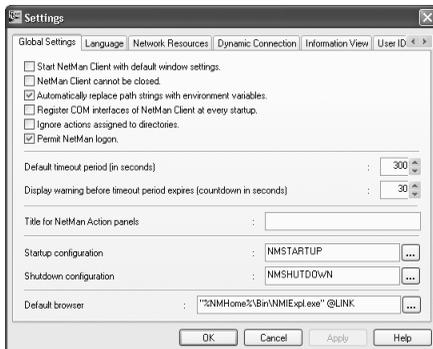
You can assign a given desktop as the “start” desktop for a user, user profile or station profile:



## Startup and Shutdown Configurations

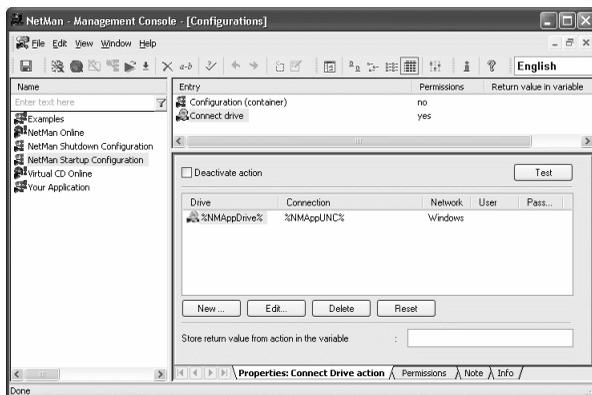
These configurations are not absolutely necessary, but can be quite useful. To create global “Startup” and “Shutdown” configurations, simply enter the IDs of the desired configurations in the corresponding fields on the GLOBAL page of the NetMan Settings:

When you first install NetMan, the configurations with the IDs *NMStartup* and *NMShutdown* are your global startup and shutdown configurations. The only function of the *NMStartup* configuration is to map an application drive (Connect Drive action); if you already have an application drive in your network, you can deactivate this configuration.



### Note

*If the `NMAppDrive` and `NMAppUNC` variables are not defined in the NetMan Settings (see chapter 4, “Application Drive”), the ‘execute’ permission for the drive mapping action is not granted.*



The shutdown configuration can be used to disconnect the drive (undo drive mapping).

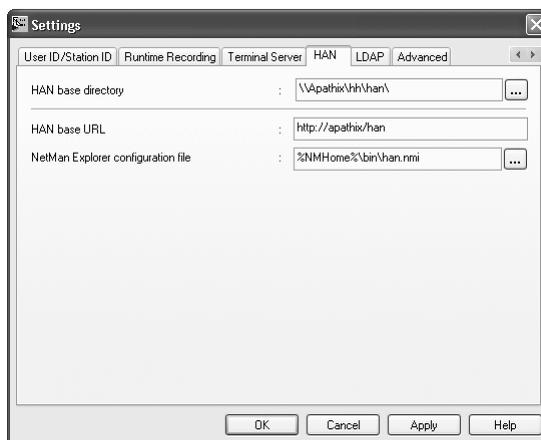
You can edit these configurations to meet your own requirements. In general, startup

configurations are used to set up a specific working environment for NetMan when it is started, and shutdown configurations to restore the previous environment when the NetMan system is shut down. Many system administrators will want to create an environment that has a number of user-specific settings; you can do this by assigning startup and shutdown configurations to individual user profiles, users, station profiles and stations. These configurations are processed in this order after any global startup configurations have been processed.

On the other hand, you do not have to create a number of separate startup configurations in order to have several actions executed at startup. Since you can assign 'execute' rights to individual actions within a configuration, the effects of any given configuration can be made to vary in accordance with your assignment of permissions.

### ***Integrating HAN Accounts***

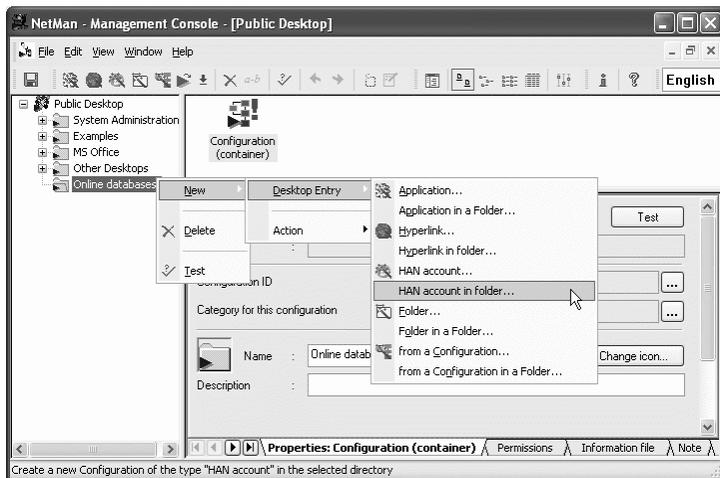
If you have purchased and registered the HAN Module, you can load HAN accounts as NetMan configurations. To do this, you need to enter the path to your HAN Module in the NetMan Settings:



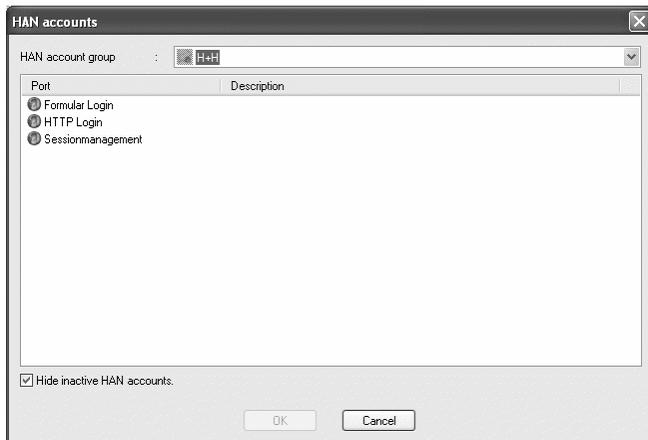
Enter the path to your HAN installation under HAN BASE DIRECTORY. The URLs for NetMan Hyperlink configurations are now made up of the HAN BASE URL you enter here and the name of the HAN account.

Entering a NetMan Explorer configuration file is optional; the file you specify here defines default settings applied to all HAN accounts imported into NetMan.

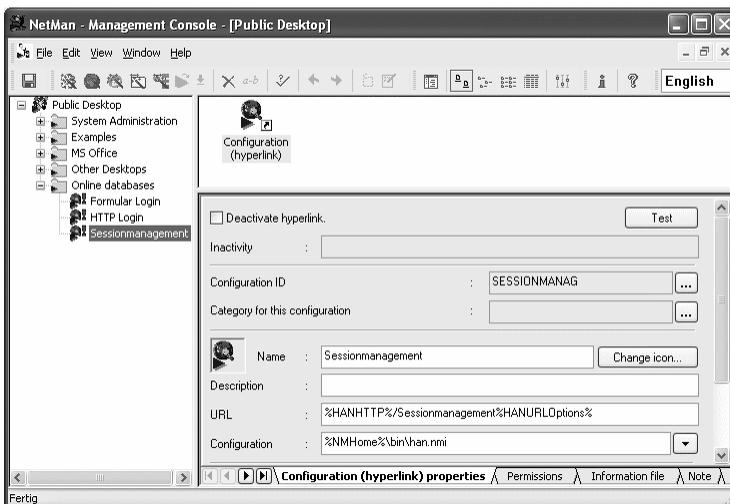
When these settings have been configured, the HAN accounts are available in your Management Console and can be integrated in NetMan desktops.



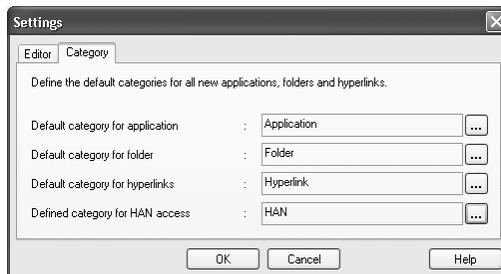
Mark the accounts you wish to import as NetMan configurations.



A HAN account is integrated as a normal Hyperlink configuration. The new Hyperlink configuration inherits the properties (name, description, URL) already defined for the HAN account in question.



Select **VIEW / SETTINGS** to change the default category for data imported from HAN. In the initial NetMan installation, this category is called „HAN.“





## Discussion

## The Trace Monitor and the Internal Architecture of NetMan

When you launch a NetMan Container configuration, a sequence of actions is set in motion. If anything goes wrong, you need a tool that helps you diagnose the problem.

This is where the Trace Monitor comes in. This program is the second element in the System Administration folder. For the following demonstration, start the Trace Monitor first, and then launch the sample configuration that calls “Calc.exe” (look in the list of all configurations if this is no longer found in your NetMan desktop). The Trace Monitor should output the following:

```

H+H Trace Monitor
File Edit View Z
[Icons]
NetMan Client 32 : Execute configuration 'Your Application'.
NetMan Kernel : Executing 'Your Application' configuration (Configuration ID: 'YOURAPP').
Execute action : Program
NetMan Kernel : Executing program start with action ID 'YOURAPP  1'.
NetMan Kernel : Updating position: 'MyLicense' 0
NetMan Kernel : Executing command line 'calc.exe'.
NetMan Kernel : Program was successfully started.
Press F1 for Help. Level: Trace Messages
  
```

Although it is easy to see how this information can help you locate sources of error, the amount of data output here probably poses more questions than it answers. For example, you might be wondering what the *NetMan Kernel* is.

This discussion of NetMan’s internal structure provides detailed information that could be useful to you as background knowledge about the system. It is not, however, a prerequisite for working with NetMan.

The Trace Monitor is a supplemental program for localizing problems that may occur when you run NetMan configurations or programs. You can start the Trace Monitor for diagnosis purposes when a problem appears, or have it running before the NetMan Client is launched, to help keep start problems to a minimum. To start the Trace Monitor before NetMan runs, call *HHTrace.exe* from the NetMan root directory (`%NMHome%\Bin`).

Now we shall take a closer look at the Trace Monitor output generated in the example above:

```
NetMan Client 32: Execute configuration 'Your Application'.
```

This line is output by the NetMan Client in response to the double-click on the “Your Application” configuration.

NetMan Kernel: Executing configuration 'Your Application'.

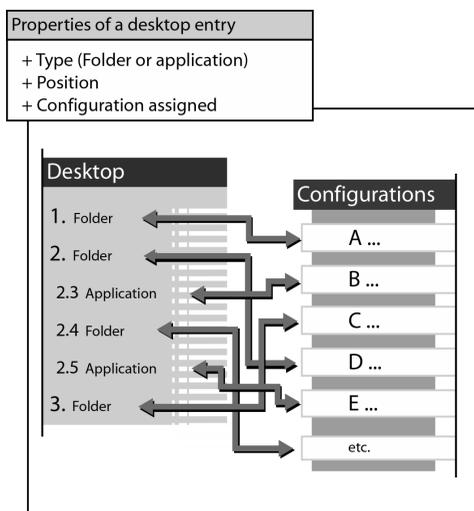
Now the NetMan Kernel takes over. This module performs its tasks invisibly. This "division of labor" is necessary because the NetMan Client is not the only user interface in NetMan. *HTML View*, for example, generates HTML pages that can contain NetMan application calls. The *NetMan Command Line Program* can be used to launch a NetMan configuration from any interface (such as a browser). In either of these two cases the NetMan desktop is not displayed, but the program properties that NetMan adds to applications and hyperlinks are still active.

To launch NetMan configurations from interfaces other than the NetMan Client, call the NetMan command line program and enter the ID of the desired configuration as an argument:

```
NMcmd32.exe /id:<configuration ID>
```

Because your NetMan configurations can be launched from your NetMan desktops or from the command line or a hyperlink, you need a special program window that lets you view all of your NetMan configurations at once, independent of desktop entries. This overview is provided by the Configuration Editor, which is opened when you select the CONFIGURATIONS element in the selection bar of the Management Console:

The following diagram illustrates the relationships between NetMan configurations and desktop entries. Think of a desktop as a customized index to some or all of your configuration database:

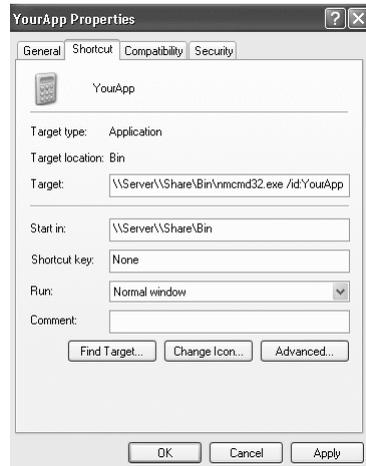


The configuration editor shows you the configurations independent of any desktop structures.

As you see here, the configuration ID for the sample configuration (“Calc.exe”) is “YourApp”. This ID is the argument passed to the NetMan command line program:

```
%NMHome%\Bin\NMCmd32.exe /id:YourApp
```

In the next example, the same command line call is also used in a desktop link:



In this case, the command line program („nmcmd32.exe“) passes the configuration to the kernel for execution. The same command is executed in the background when you launch this configuration from the HTML View.

Let us return to the status messages in the Trace Monitor. After the message from the kernel, the “*NetMan Actions*” module comes into play:

```
NetMan Actions: Program
```

As you see, the kernel does not execute all jobs itself; rather it coordinates them and assigns the different tasks to the various NetMan components, which are implemented as DLL files. In the message above, the *Actions DLL* informs the kernel that there is a program to be executed.

Select the **SETTINGS** item from the **VIEW** menu to see the options available for the Trace Monitor. For example, you can

- filter output according to program components;
- assign font colors according to components so you can identify certain steps at a glance;
- define the level of output; for example, to obtain even more detailed output about certain internal sequences;
- save output; for example, to append it to your support questions.

The window below shows the Trace Monitor output after “YourApp” was started from the command line:

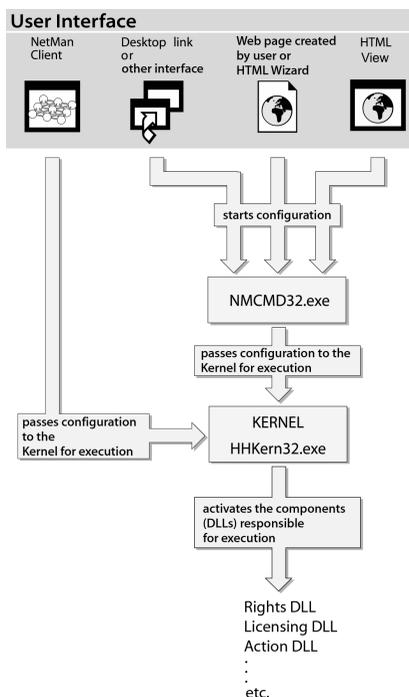
```

Trace Monitor
File Edit View Z
NetMan Kernel : Starting Kernel.
NetMan Kernel : Connecting to NetMan Service:
Station registration : Station 'APATHIX' will be registered.
Station registration : Writing data on 'APATHIX' in the station database.
Station registration : Recording application start by user 'BEEK'.
NetMan Kernel : Executing 'NetMan Startup Configuration' configuration [Configuration ID: 'NMSTARTUP'].
Evaluate rights : Checking "The path " exists": This is false.
Evaluate rights : Access permission not granted.
Execute action : Action cannot be executed [insufficient rights].NetMan Kernel : Executing 'Your Application'
configuration [Configuration ID: 'YOURAPP'].
Execute action : Program
NetMan Kernel : Executing program start with action ID 'YOURAPP' 1'.
NetMan Kernel : Updating position: 'MyLicense' 0
NetMan Kernel : Executing command line 'calc.exe'.
NetMan Kernel : Program was successfully started.
Press F1 for Help. Level: Trace Messages

```

In the first step, the kernel was started in the background. The kernel resides in memory as long there is an application running – even if you had been using the NetMan Client interface but have since closed it.

The following diagram shows how NetMan components work together:



## CD-ROM-based Applications

A CD-ROM-based application (referred to in the following as “CD application” for short) is an application that refers to data on a CD during run time.

Installing CD applications in a network can sometimes be a complex operation:

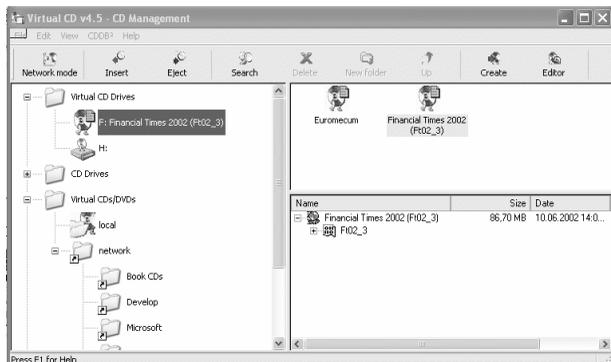
- CD applications often run only from the same drive in which they were installed.
- The drive entered during setup is often stored in the Registry, in INI files or in non-editable files, which means it can be changed only by re-installing the program.
- The more CDs belong to a given application, the more difficulties are created by the problems mentioned above.
- In a network that has a lot of CD applications, there may be competition among them for a limited number of drive letters.
- CD applications often look for their CD data in a physical CD drive.

In the following, we demonstrate the installation of a CD application in NetMan. The following parameters apply for this example:

- The application will be installed on K:, the central application drive. Our application drive has already been defined in the NetMan Settings; with these settings, clients access the applications that are installed on the network at K: (*NMAppDrive*).



- The „Virtual CD“ program is used to map the CD data. The (virtual) application CD is inserted in the (virtual) F: drive using the Virtual CD Management program.



Now we begin the installation:



The Setup program offers us the option of specifying the CD drive or searching for the disc.



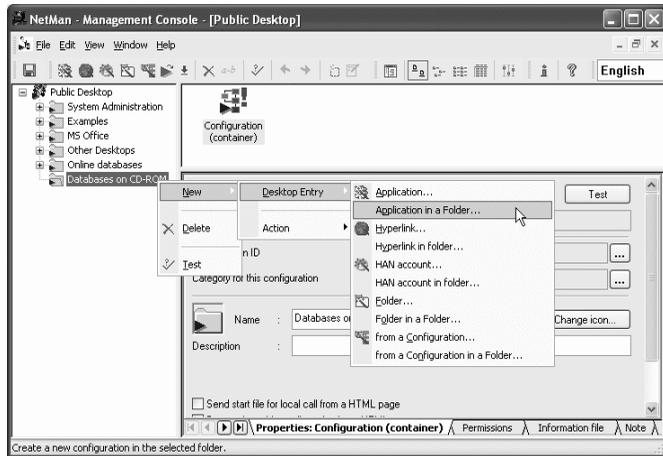
We elect to search for the installation disk, and it is found in the F: drive. The program is then installed to drive K: and a new entry is added to the Start menu:



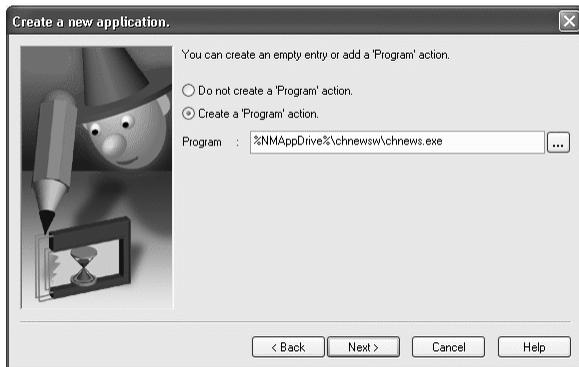
We start the application from here, and find it has no difficulty locating its CD data. Thus the new CD application is ready to use.

The next step is to distribute this application over the network.

We begin by creating an application in the “Databases on CD” folder:



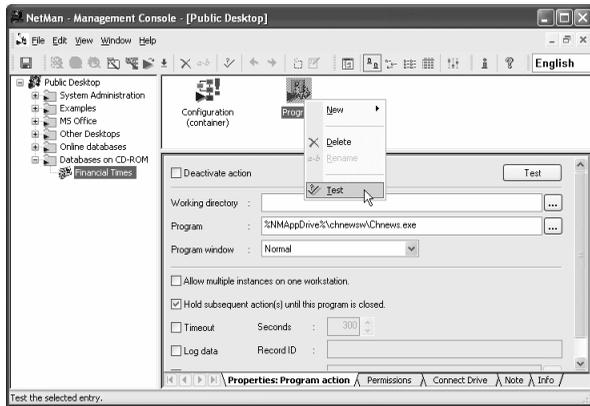
We highly recommend copying the program call from the new link in the Start menu and pasting it into the Program action:



## Tip

*Copying the program call ensures that the command line and any arguments required are entered correctly, as well as the working directory, if it differs from the program directory.*

NetMan automatically converts “K:” to “%NMAPDrive%” in the command line. Our first test of the Program action is successful.

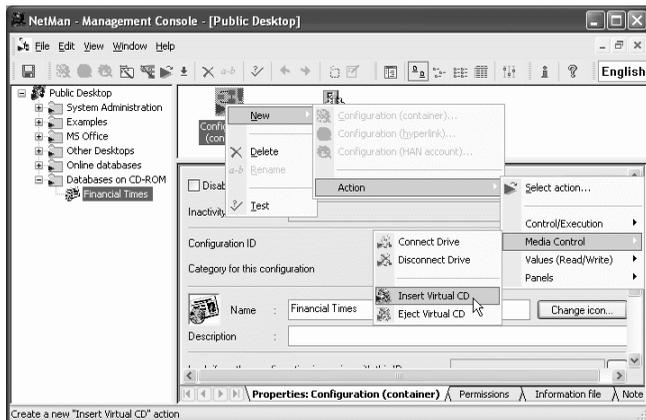


There are still two more functions to be configured:

1. We want the CD to be mapped automatically when the application is started.
2. We want to be able to start the application on any workstation.

NetMan has two actions specifically designed to support Virtual CD:

- Insert Virtual CD
- Eject Virtual CD



We add these two actions to the configuration, bracketing the Program action. NetMan automatically sets the *NMVCDDrive1* (or *NMVCDDrive2*, 3, etc.) variable(s) on

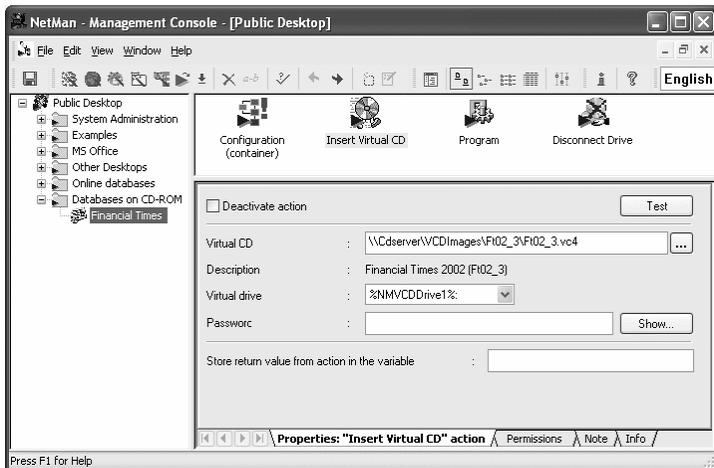
the client workstation in accordance with the Virtual CD drive when the application is launched.



### Tip

*Because many CD applications look for their CDs in the same drive that was used for installation, it is important to use consistent Virtual CD drive configurations throughout the network—for example, by using a modified Client Network setup—so that your (virtual) CDs can use the same drive letter on every station.*

Enter the path to the desired Virtual CD image file in the VIRTUAL CD field:

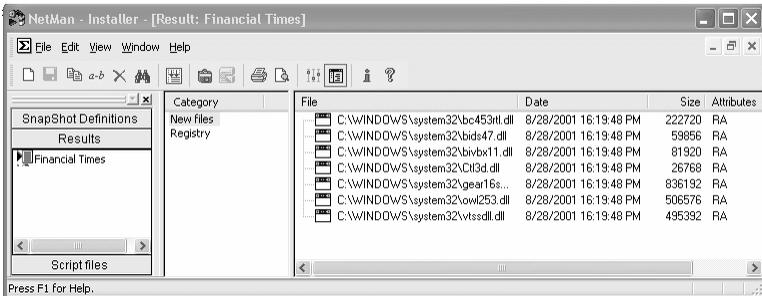


### Tip

*Any time you have trouble with a configuration that contains multiple actions, it is a good idea to run the Trace Monitor to diagnose the problem.*

When this application is started on a different station than the one it was installed on, a message appears stating that required DLLs are missing. There are a number of ways to address this problem:

1. You can repeat the entire Setup procedure on the station in question.
2. You can copy the missing DLLs to the application's working directory. In this case, the application runs once we copy a total of 4 DLL files to the %NMAppDrive%\chnews directory.
3. You can use the NetMan Installer to monitor the initial installation procedure. Once the Setup is complete, the Installer tells you what changes were made to the system by the installation:



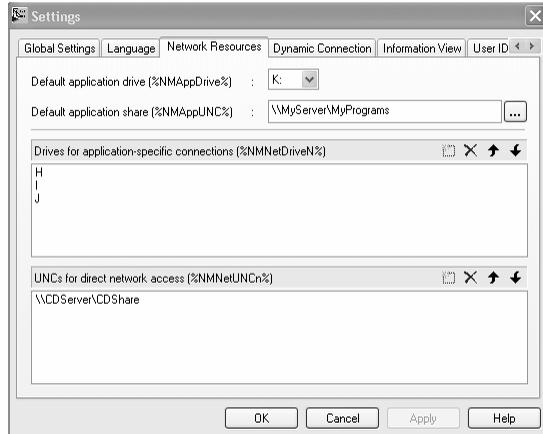
The NetMan Installer can generate a script based on this Before/After comparison, which you can add to the Program action to have the required system modifications performed automatically when the application is launched for the first time on a given station (for details, see the NetMan Installer Module manual).

In the case at hand, however, all you need to do is copy the required DLLs (reported by the Installer) to the application's working directory.

The demonstration above is application only if you have the Virtual CD program. Without Virtual CD, you can still map CDs in NetMan with relatively little difficulty. For example, if you have a CD server integrated in your network that provides CD as shares or volumes, simply proceed as follows:

- CD servers usually allow access to all CDs as a directory structure under a UNC path. Enter the shares for these CD directories in the NetMan variables *NMNetUNC1*, *NMNetUNC2*, etc.
- Reserve a block of drive designations which the applications can use for runtime access to their data CDs. The number of drives you should reserve depends on the number of CDs that might require drives in parallel at any given time. Enter the reserved block in the variables *NMNetDrive1*, *NMNetDrive2*, etc.

In an environment with heavy CD-ROM usage, the definition of Network Resources in the NetMan Settings might look something like this:



There are two CD servers, each of which permits access to all of its CDs. The drives H: through J: are reserved here, for temporary run-time mapping of local drives for applications.



### Note

*The variables for reserved, temporary drive mapping do not contain colons because some applications expect their data source reference as a drive letter without a colon.*

Under these conditions, you can distribute your CD applications in NetMan as follows:

Try at first to run the application setup in the network environment using the *NMNetUNCn* variable. If this does not work, you can assume that the application requires a fixed drive designation. Map the required drive (from the reserved drives) at run time for the application.

In some cases, you can switch the mapped drive to a UNC path at a later point.

If you find that the application can access its data CD under different drives, either because it can search all drives or because the drive designation can be passed in the command line, use *NMNext* as the drive designation. In this case, the first available CD drive found on the workstation is used for mapping and written in the *NMNext* variable. You can specify how NetMan stores a value in *NMNext* on the DYNAMIC CONNECTION

page of the NetMan Settings (for details, see “Configuring Drives and Shares” in chapter 4).

### Example: Mapping a Share to a Reserved Drive

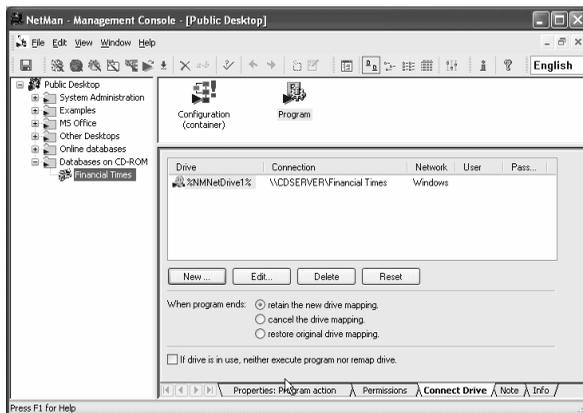
The drive in question has to be shared in your operating system first.

Then you can map a drive before the program starts, using the *MMNetDrive1* variable, to connect the CD to the reserved drive designation.



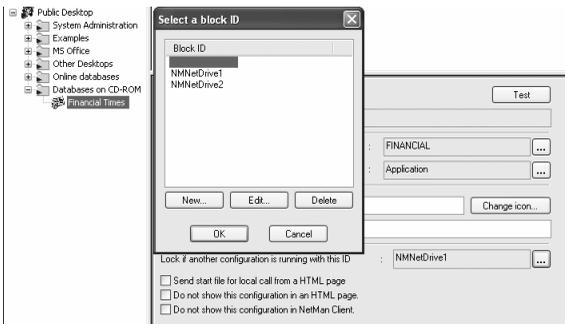
### Tip

Use the drive mapping function that is integrated in the Program action, as this is much more powerful than the Connect Drive action. The latter is best used for other functions, such as startup and shutdown configurations for example, in which Program actions are not allowed.



The example shown here blocks the drive that is in use; in other words, other applications that require this drive cannot be started on the same workstation on which (in our example) „Financial Times“ is running.

You can configure this as follows:



Applications that have the same “Block ID” cannot run simultaneously on one machine. When a second application with the same Block ID is started, the following error message is output:



### Example: Mapping a Share to a Specific Drive

If you know exactly where your application gets its data (i.e., which drive the required CD is in), drive mapping is even easier.

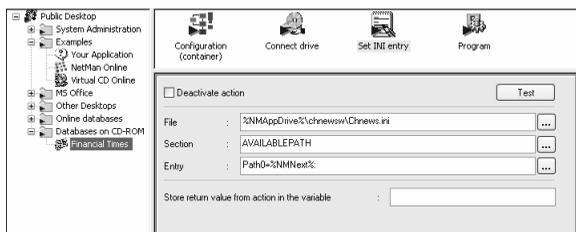
The “Financial Times” application has an INI file with the following sections:

```
[DISCLOCATION]
```

```
[AVAILABLEPATH]
```

```
Path0=f :
```

In cases like this, you can also use the *MMNext* variable for the drive designation. You simply have to “tell” the application that its drive is stored in this variable. In our example, this is done by inserting a Set INI Entry action:



With this setting, the value determined for *MMNext* is written in the INI file before the application starts. If the application reads its drive from the Windows Registry, you can use a Set Registry Key action to write the *MMNext* value in the Registry when the program is launched.

### Example: UNC-based Access

A very convenient alternative is to write the UNC path in the INI file, if the application can process UNC syntax. Fortunately for this demonstration, this is the case with the “Financial Times” application.

```
[DISCLOCATION]
```

```
[AVAILABLEPATH]
```

```
Path0=\\Apathix\CDSshare\FT302
```

In this case, you require neither a special share for the CD (“CDSshare” is all you need) nor an available drive letter, which saves you the trouble of mapping a drive before the program is launched.

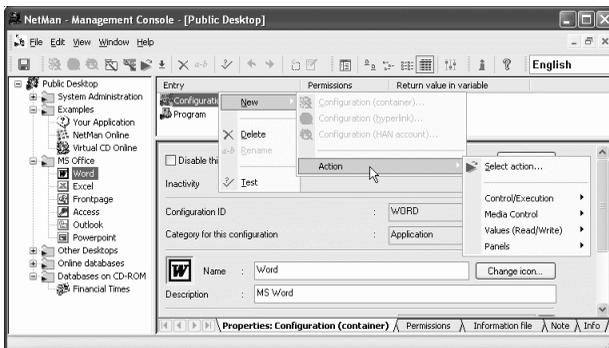
This method can have disadvantages in certain instances. For example, users can recognize the location of the application data, and can load additional data (if there is any) for the same retrieval interface, which you might not wish to allow.

### NetMan Actions

Throughout this manual we have repeatedly mentioned the broad range of possibilities afforded by the variety and number of actions you can add to your NetMan configurations. In this section, we present details on the different types of actions, and point out the convenience afforded by adding other actions to your NetMan configurations, rather than simply using Program actions on their own.

NetMan actions are divided into the following categories:

- Control/Execution
- Media Control
- Values (Read/Write)
- Panels



**Note**

*Each action type is described in detail on the corresponding **Info** page found shown in the Management Console. For a complete list of all available actions, with their **Info** page descriptions, refer to the *NetMan Almanac*.*

We have already presented a demonstration of the most important action, the **Program** action. We would like to point out once more that a NetMan configuration is a **user-definable sequence of actions**.

Any type of action, including Program actions, can occur repeatedly in a given NetMan configuration and can be used in any combination.

Thanks to NetMan's Windows Script interface, this broad range of possibilities is extended even further.

**Note**

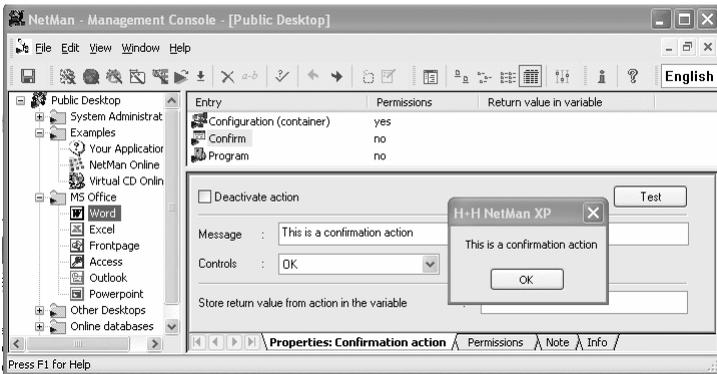
*It is not necessary to know all about every type of action. If all you need are Program actions, you do not have to bother with the entire spectrum of other actions. In the following, we present just a few practical examples involving some of the other actions, to give you some idea of the best uses for NetMan in your own network environment.*

Actions can generate **Return values** (process variables). The values stored in return variables, which can be the result of user input, are available for processing by any or all of the subsequent actions in a given configuration.

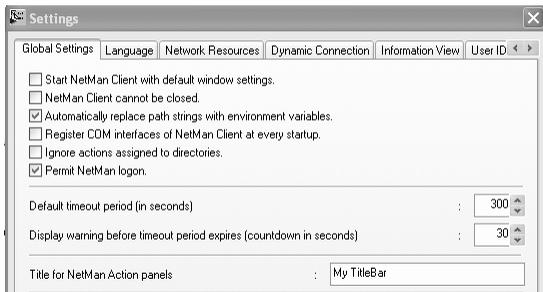
This is implemented by defining a **Variable Check** condition within an individual action, to determine the results of preceding actions.

## Controlling an Action Sequence

You can insert a Confirmation action to pass information to the user before the program starts.



The text shown in the title bar of windows opened by NetMan actions is defined in the NetMan Settings.

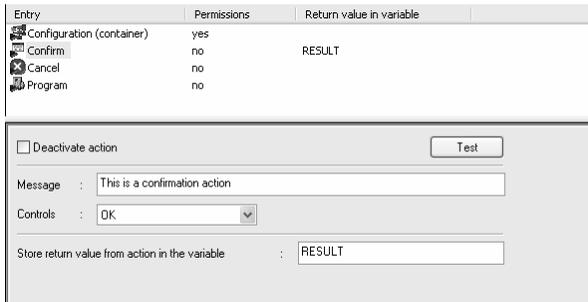


The title bar text now reads „My TitleBar“:

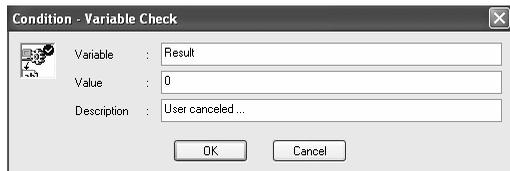


If „Cancel“ is selected here, the NetMan configuration is canceled; if „OK“ is chosen, configuration processing continues.

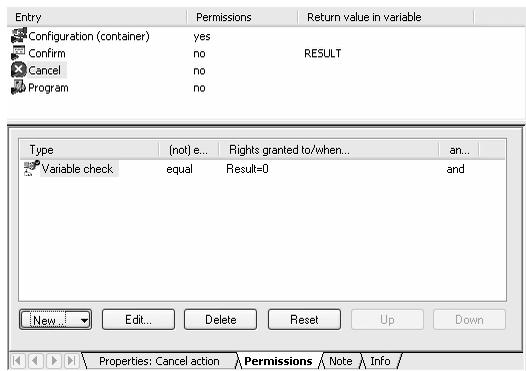
With the return variable functions, you can have the result of user input written in a return variable and use it to control subsequent processing; for example, by inserting a “Cancel” action.



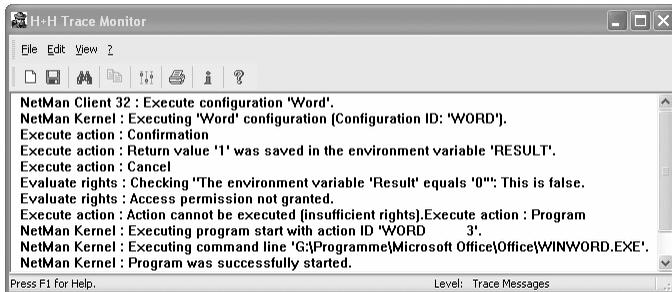
For this purpose we define the condition „User canceled...“:



The execution of the „Cancel“ action is made dependent on the Variable Check condition.



When the user selects „OK,“ the following output is seen in the Trace Monitor:



```

H+H Trace Monitor
File Edit View ?
NetMan Client 32 : Execute configuration 'Word'.
NetMan Kernel : Executing 'Word' configuration (Configuration ID: 'WORD').
Execute action : Confirmation
Execute action : Return value '1' was saved in the environment variable 'RESULT'.
Execute action : Cancel
Evaluate rights : Checking "The environment variable 'Result' equals '0'": This is false.
Evaluate rights : Access permission not granted.
Execute action : Action cannot be executed (insufficient rights).Execute action : Program
NetMan Kernel : Executing program start with action ID 'WORD - 3'.
NetMan Kernel : Executing command line 'G:\Programme\Microsoft Office\Office\WINWORD.EXE'.
NetMan Kernel : Program was successfully started.
Press F1 for Help. Level: Trace Messages
  
```

The 'execute' permission is evaluated logically: because the return value is not „0“, permission to execute the „Cancel“ action is denied and the subsequent action is processed.



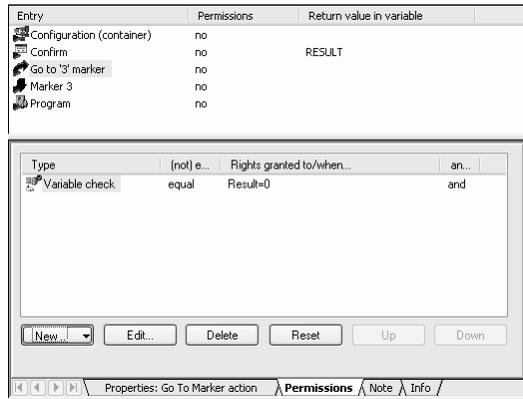
#### Note

*The following example should help to illustrate the logic behind this process: Say you have inserted a “Password” action at the beginning of a configuration, to ensure that only authorized users can launch the configuration. Because NetMan administrators do not require permission to launch the configuration in question, however, you configure a condition that denies ‘execute’ permission to the Password action for users operating under an administrator account.*

*When administrators launch this configuration, they are not prompted for a password, and the following output is shown in the Trace Monitor “Action cannot be executed (insufficient rights).” This message could be confusing, as administrator accounts are generally associated with unlimited rights.*

*This particular function has proved to be very useful for allowing administrators to perform management-level tasks on end-user workstations without having to initiate a separate logon.*

Returning to the example of the “Cancel” action: the same purpose can be achieved by inserting a Go to MARKER action. Here, too, the execution of the action is dependent on the return value resulting from user input. If the user chooses the “Cancel” button in the panel generated by the Confirmation action, processing skips to the end of the configuration and the Program action is skipped entirely.

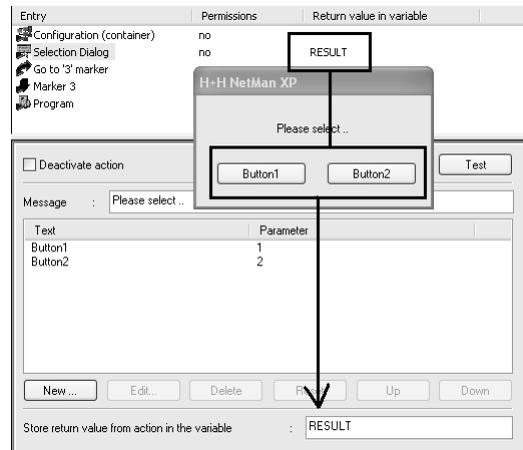


The „Go to Marker“ action is very useful for skipping an entire series of actions, where you would otherwise have to define ‘execution’ conditions for each action individually. You can also use it to jump **back** to an action located at an earlier position in the sequence. This lets you create logical loops; for example, to „execute Action Y repeatedly until Condition Z no longer exists.“

### Simple Examples of the Most Frequently Used Actions

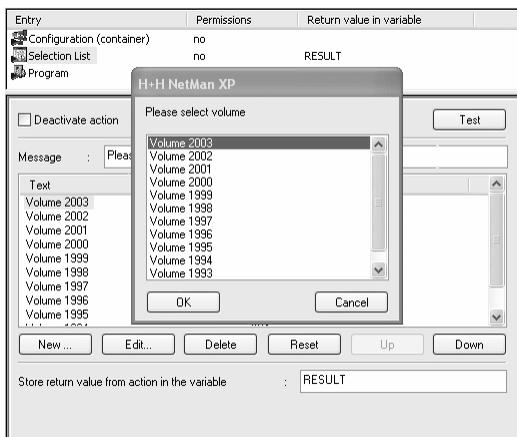
Similar to the *Confirmation* action, the “Selection Dialog” lets you offer the user a choice of responses, in the form of buttons for which you define both text and function.

Each possible response writes a specified value to the return variable resulting from this action. This value can in turn be evaluated based on subsequent conditions or used in following actions.



If you want to present the user with a large number of options, rather than just two, you might insert a *Selection List* action instead of the *Selection Dialog*; the function is similar, but the choices are presented in a list rather than on buttons.

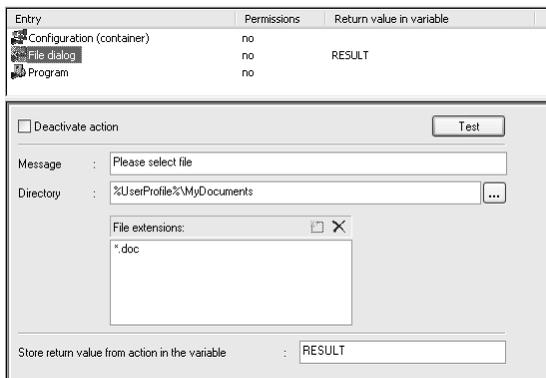
Because you can assign a text for the end user to each parameter, users can be presented with meaningful choices rather than the cryptic texts often found in such cases.



*Selection and File Dialogs* are generally useful for generating values to be passed to programs in the form of command line arguments. For example, the *File Dialog* lets the user select a file.

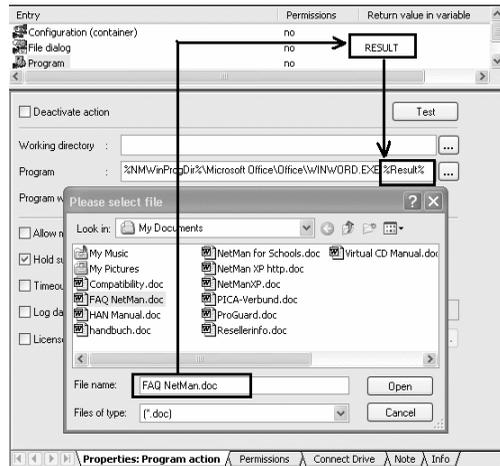
In the example below, the File Dialog action presents the DOC files stored in a given directory for user selection.

As you see here, you can also use system variables in the path name (%USERPROFILE% is a system variable in Windows NT); these are resolved by NetMan at run time.

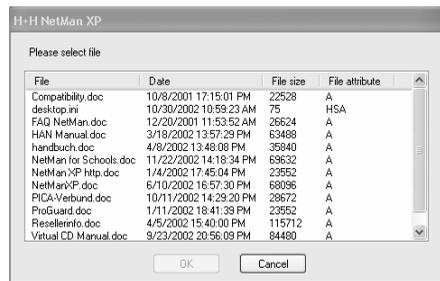


When it is executed, the File Dialog action opens the standard Windows File/Open

dialog; the file selected is written in the „Result“ variable and passed to the „Winword.exe“ program:



If you use a *File List* action, as opposed to *File Dialog*, the user cannot browse through the various folders and drives in your network; rather, a list of files is shown for selection. You as administrator specify which files are included in this list by indicating the location (directory) and file type. You can define whether the selection window shows the file size, date and/or attributes, and specify the maximum number of files that a user can select.



The *Parameter* action opens a dialog for user input to be passed to the program in command line arguments. The following two features are available for customizing this dialog:

- If you use square brackets in the “Parameter” definition, the user will see only what is inside the square brackets and nothing else that is in the “Parameter” field. The square brackets might contain spaces, or a default parameter that the user can overwrite. Text outside the square brackets is

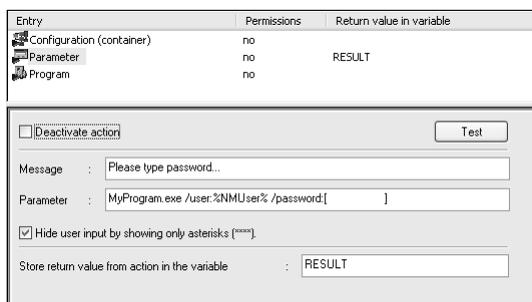
passed to the program in the command line without modification.

You can have the user input hidden (only asterisks displayed during input).

The following example illustrates one possible use of the Parameter action: Say you have a resource for which logon is required, entailing input of a user name and a password. A *Password* action is not particularly well-suited for use here, as it serves in an action sequence to determine whether the configuration is processed or not. (for example, when it involves opening a certain folder in the NetMan Client). Assuming the following syntax for the required command line input:

```
/user: <username> /password: <password>
```

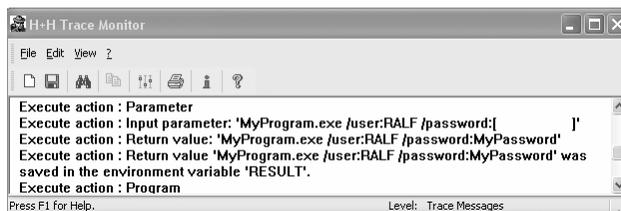
you can configure your Parameter action as follows:



The user name is known to the system, and passed on using the „NMUser“ variable. The function of a password prompt is taken over by the Parameter action; all that the user can see - and edit - in this case are the 10 spaces, represented by asterisks:



As always, it is helpful to look at the output in the Trace Monitor if any problems occur during testing. In our example, the following is shown:



The syntax of the NET USE command is similar to that used in the example above:

```
NET USE
[Devicename | *] [\\Computername\Sharename [\Datamedium]
[Password | ]
[/USER: [Domainname\]Username]
```

Thus you could conceivably use this command for logging on to a network resource; for example, by writing this command in an *Execute* action. The *Execute* action has fewer options than the Program action, and unlike the Program action, can be included in NetMan Startup and Shutdown configurations.



### Tip

*With these types of actions, the NetMan helper programs can be practical as well. Descriptions of these programs are included in chapter 10, with lists of the valid arguments, under “Helper Programs for the ‘Execute’ Action.” The “NMNCon32.exe” and “HHCcmd.exe” programs could be useful in the above example.*

In the following action, the NET USE command is executed by the NetMan “HHCcmd.exe” helper program, which is started by an Execute action:

Entry	Permissions	Return value in variable
Configuration (container)	no	
Parameter	no	RESULT
Execute	no	

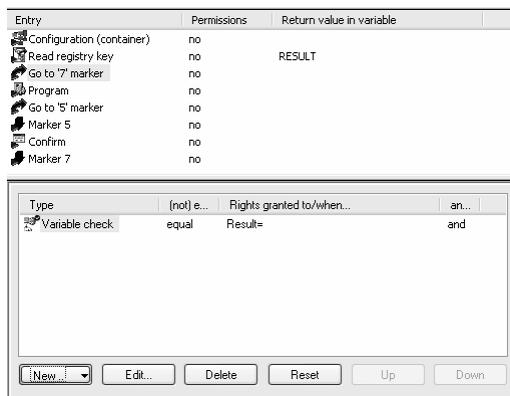
Deactivate action 
  
 Program : hhcnd.exe net use x: \\server\share %Result% /user:%NMUser% 
  
 Working directory : %NMHome%\Bin\
   
 Program window : Normal 
  
 Hold subsequent action(s) until this program is closed.
   
 Store return value from action in the variable :

## Complex Actions

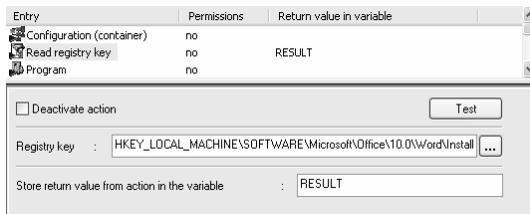
For the next example, we return to our MS Word configuration.

Let us assume you want to find out where the Microsoft Office directory is located on a given workstation, and then start Word from that directory.

You can configure this sequence as follows:



The Office path is written to the „Result“ variable; if no value is stored here, the configuration skips to a Confirmation action which announces that the Word program was not found. The Office path is determined as follows:

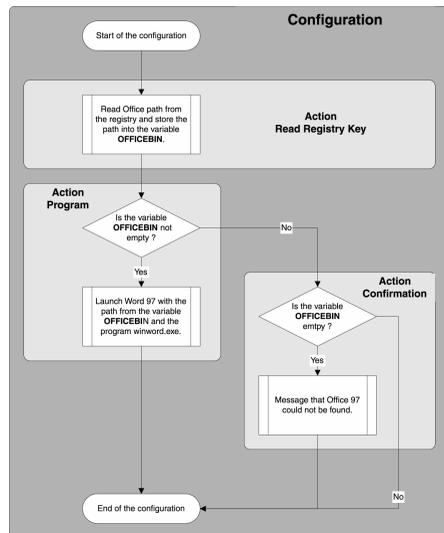


If the path is detected, it is written to the variable which is used to call the program:

```
%Result%Winword.exe
```

If the Word program is found, the configuration skips to a marker placed at the end of the configuration (subsequent to the Confirmation action).

The following diagram illustrates this sequence:



Our “MS Word” configuration clearly demonstrates the logical structure of NetMan “Container” configurations. With one small addition, this can be used to address a particular problem that often comes up in the areas in which NetMan is used:

Let us say NetMan is used by an information service in a large enterprise that provides Word documents on terminal servers as information sources. In this case, documents can be provided for selection using File Dialog, File List or other types of actions. A Parameter action following the file selection determines whether the chosen document can be edited by the user (opened with “Winword.exe”), or is opened in “read-only” mode (by opening it with “WordView.exe,” for example). The former variant is applied for members of staff in the Information Services department, and the latter for other users.

Entry	Permissions	Return value in variable
Configuration (container)	no	
File List	no	
Parameter	no	EDITOR

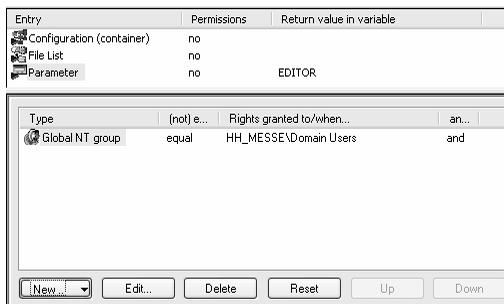
  

Deactivate action Test  
 Message :   
 Parameter :   
 Hide user input by showing only asterisks (\*<sup>\*\*\*\*\*</sup>).  
 Store return value from action in the variable :

The Parameter action inserted here does not prompt user input, as the „Parameter“



field in this action does not contain square brackets. The „Editor“ variable is set in the background to „WordView.exe“ for non-domain users.



A similar solution can be used to:

- start the NetMan Explorer rather than the default browser for certain user groups,
- open the enterprise Web site in a browser or in an HTML editor (e.g., Front Page), or
- open different programs for a given task, depending on client operating system.



## Discussion

### ***The Windows Script Action***

The Windows Script action lets you run scripts written in JScript, VBScript and WSH. VBScript and JScript can be combined within WSH (=Windows Scripting Host) scripts.

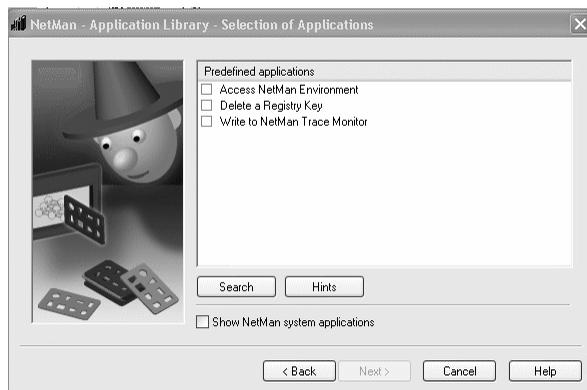
The option of writing your own scripts represents an expansion of the range of NetMan functions, and combines the powerful functions of NetMan actions with those of Windows Script. NetMan is particularly well suited for this, because all system parameters are stored in variables; a script once written is universally valid throughout your NetMan system.



## Note

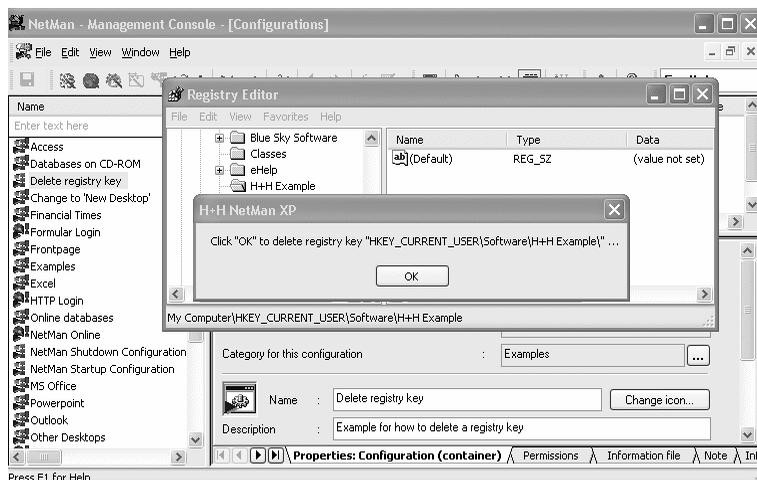
*The following information describes the NetMan interfaces for Windows Script and are relevant only to users who are familiar with JScript, VBScript and/or XML.*

It is difficult to give you an impression of the performance capability of these functions within the scope of this manual. The three examples described below are available from the NetMan Application Library:



### Passing Arguments to Scripts

Parameters can be passed to scripts in command line arguments. In the VBScript *DeleteRegKey.vbs* the Registry key to be deleted is passed. The script is configured to generate output to the Trace Monitor if the HHMes32.dll is found on the client station; otherwise, no output is generated.



There are a number of sample scripts available in the Internet, and in appendices to textbooks. The most important factor for working with NetMan is the way in which

parameters are passed. For this reason, we include below an example of a JScript that passes command line arguments:

```
var objArguments = WScript.Arguments;

if (objArguments.length == 0)
{
    for (var i=0; i < objArguments.length; i++)
    {
        switch(i)
        {
            case 0: cParam1 = objArguments(i) ;break
            case 1: cParam2 = objArguments(i) ;break
            case 2: cParam3 = objArguments(i) ;break
            ....
        }
    }
}
```

### ***Output to the Trace Monitor***

The HHMes32.dll has to be registered as a COM server for this purpose:

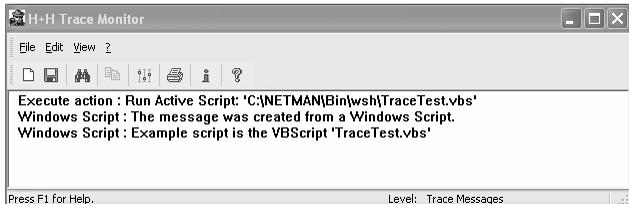
```
regsvr32 /s HHMes32.dll
```

In our sample configuration, this command is executed automatically. A new instance must be created at run time:

```
Dim objHHMes32
Set objHHMes32 = CreateObject ("HHMes32.HHMsg32.1")

objHHMes32.Level = 2           ' sets the output level to
"trace messages".

objHHMes32.Module = "Windows Script" ' sets the name of
the module
objHHMes32.Trace "<message>" + Chr(10) ' creates the out-
put
```



## Access to the NetMan Environment

The HHEnv32.dll has to be registered as a COM server for this purpose:

```
regsvr32 /s HHEnv32.dll
```

In our sample configuration, this command is executed automatically. A new instance must be created at run time:

```
Dim objHHEnv
Set objHHEnv = CreateObject("HHEnv32.HHComEnv")
Var = HHEnv.HHEnvGet("<var">)           ' reads from
the environment
bRC = HHEnv.HHEnvSet("var", "<value>")   ' writes to
the environment
```

In this example, the NMUser and NMHome variables are read from the environment, and a test variable is written to the environment:

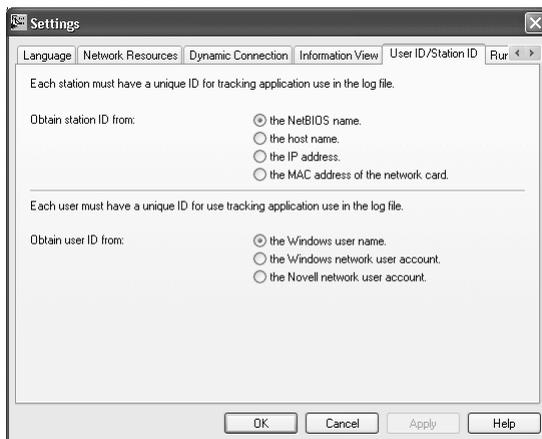




## 6. Users, Stations, Groups and Profiles

The first time you start NetMan, the users and workstations in your network are automatically added to the NetMan user and station databases. When a new user or station starts NetMan for the first time subsequent to your first NetMan startup, a new data record is created. The key field in these data records is the “User/Station ID”.

Data records are stored under the ID you specify in the NetMan Settings:



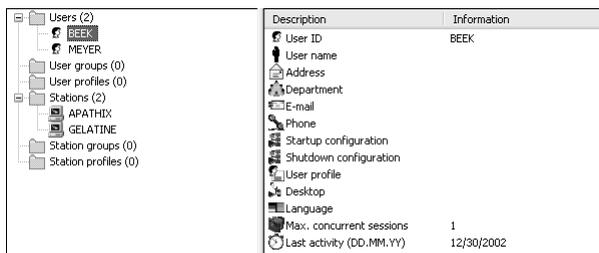
To view or edit these data records, open the **RESOURCES** program window from the Management Console:



## NetMan Users

In our example, we have chosen to use the Windows NT network user login name as the user ID. The format of this ID in the user database is “domain\user”. NetWare user names are written with NetWare syntax, and can be detected only by the IntraNetWare Client from Novell. If a NetWare user name cannot be determined, the data record is stored under the Windows NT user name.

You can create, edit, re-name and delete user data records.



To create new user, select CREATE from the EDIT menu and enter a User ID. This opens the following window:

The screenshot shows the 'Edit User Data' dialog box for user 'BEEK'. The dialog has a title bar with 'Edit User Data' and a close button. Below the title bar is a header area with a user icon and the name 'BEEK' on the left, and 'Last activity (DD.MM.YY)' on the right. The main area contains several fields:

- Name: Ralf in der Beek
- Password: (empty field) with a 'View...' button
- Address: (empty field)
- Phone: (empty field)
- Startup config: (empty field) with a '...' button
- Shutdown config: (empty field) with a '...' button
- Profile: (empty field) with a '...' button
- Desktop: (empty field) with a dropdown arrow
- Language: (empty field) with a dropdown arrow
- Department: Installation and Support
- E-mail: beek@hh-zfk.com
- Maximum parallel Terminal Server sessions allowed: 1 (with a dropdown arrow) and a checked 'Overwrite default' checkbox

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

The LAST ACTIVITY DATE in the upper right-hand corner cannot be edited; it is updated every time the user starts NetMan.

The fields for ADDRESS, DEPARTMENT, E-MAIL and PHONE are not required for NetMan operation; they are for your administrative purposes only and can be referred to by a *Data List* action.

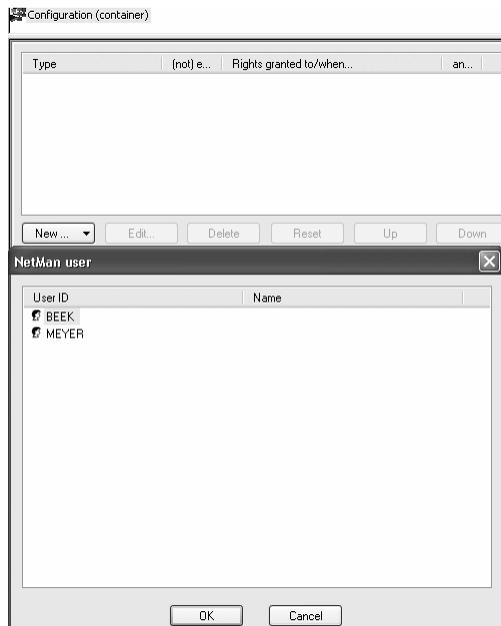
The NAME you enter here is separate from the User ID; this name is recorded in user lists for statistical evaluation purposes.



### Note

*You can define user-specific STARTUP AND SHUTDOWN CONFIGURATIONS HERE. The LANGUAGE field is active only if you have the Language Module; it allows you to define the starting language for a given user. The field for MAXIMUM PARALLEL TERMINAL SERVER SESSIONS ALLOWED is active only if you have the Terminal Server Module.*

You can open a list of users compiled from this database when assigning ‘execute’ conditions for configurations and actions in the Management Console:



You can create a user data record „manually“; for example, to

- enter a new user who has never started NetMan, or
- create a NetMan user account which does not correspond to any existing network user.



For example, you can create a user account that is used in a *NetMan Logon* action, or assigned to anonymous users on the basis of IP-address or host name through the “Terminal Server Access Control” program. We recommend assigning a password to this type of account.

### Example:

A “Guest” user in your network can start NetMan using a special NetMan user name, to which you have assigned specific NetMan permissions.



### Note

*NetMan permissions are independent of network rights; they are equivalent to ‘execute’ rights for NetMan configurations.*

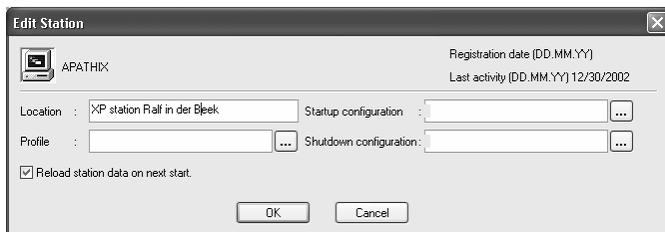
## NetMan Stations

In the example illustrated at the beginning of this chapter, we determined that the computer’s host name would be used as the station ID. The format of these entries in the database is “computername.domain”. If the host name cannot be determined, the IP address is entered instead; if this cannot be determined either, then the computer name is used.

You can create, edit, re-name and delete station data records.

Description	Information
Station ID	APATHIK
Obtain station ID from	NetBIOS name
Registration date (DD.MM.YY)	12/20/2002
Last activity (DD.MM.YY)	12/30/2002 - 10:43:13 AM
Location	beek
Station profile	
Startup configuration	
Shutdown configuration	
Reload data on next start	no
System	AT/AT COMPATIBLE
BIOS	ASUS - 56582e31/Award Modular BIOS v4.51PG
BIOS date	9/7/1999
Operating system	Windows NT 5.1 (Build 2600) Service Pack 1
RAM	384 MB
CPU	Intel Processor
Monitor	
Graphic card	
Keyboard	Extended IBM keyboard (101 or 102 keys)
Code page	850
Default printer	\\M\ACCARONIX\QL3e8000N_PCL;wmspool;Net01;
Installed printer	\\dotanix\Canon IP5000-6000 PCL5 \\M\ACCARONIX\QL3e8000N_PCL
Network card	
Network configuration	IP address: 62.225.136.122 Gateway: 62.225.136.1 Subnet address: 255.255.255.0 Host name: apathik.asi.hrv-zfrk.com Domain/Server: HRV-ZFRK Station name: APATHIK MAC address: 00D0E768A372
Network protocols	NetBios über TCP/IP
Network provider	Microsoft Windows-Netzwerk Microsoft Terminaldienste

A station data record contains the following fields:



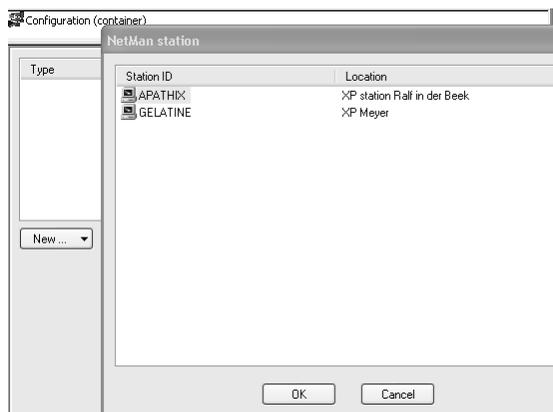
The **LAST ACTIVITY DATE** in the upper right-hand corner cannot be edited; it is updated every time the station starts NetMan.

The **REGISTRATION DATE** is important if you are using the *Named Sites* licensing scheme; the station's NetMan license is revoked 40 days from this date. If there is no license active on the date that this station starts NetMan, this date is overwritten.

The **LOCATION** field is for your information only; it can help ensure a clear overview in the lists of stations shown in programs for statistics, license administration, station monitoring and rights assignment. No entry is required here for NetMan operation. NetMan automatically enters the name of the user under whose account the station database record was created; you can overwrite this entry, if desired.

Some of the fields in the station data base can be referred to in a *Data List* action.

You can open a list of stations compiled from this database when assigning access rights to configurations or actions in the Management Console:



You can create station data records “manually”; for example, to create a new station that has never started NetMan.

NetMan detects the following data for inclusion in the station database record:

- BIOS data
- hardware
- cards installed and peripheral devices connected
- network configuration, including drivers and protocols installed
- software defaults (mail client, browser)

All of this data is recorded the first time this station starts NetMan. You can also have this data updated every time this station starts NetMan by activating the `RELOAD STATION DATA ON NEXT START` option.

## **NetMan User Groups**

You can create groups for your users.

The advantage of *NetMan user groups* is not immediately apparent, since NetMan supports your existing NT, NetWare and LDAP user groups; besides, proprietary groups are generally regarded as a disadvantage, because they are associated with additional administration tasks. But NetMan groups are active on a totally different level: they define only permissions to *NetMan configurations*, and have nothing to do with rights in directories, files or other network resources.

You may find that your existing network groups provide sufficient control over NetMan configurations. In this case, you do not need to use NetMan user groups at all.



### **Tip**

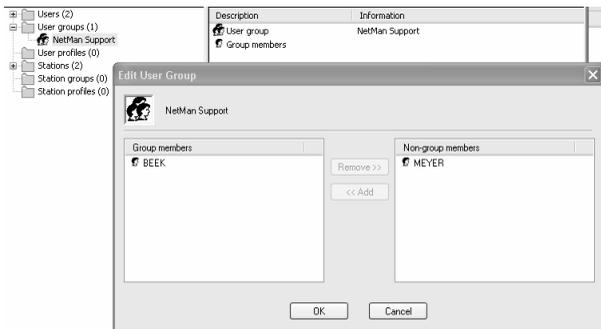
*It is best to use existing network groups wherever possible, to avoid generating extra work unnecessarily. But if you find that the existing groups cannot be used to configure the control you need, you may find it easier to create NetMan groups than to create new network groups.*

NetMan user groups are particularly useful if any of the following is true for you as NetMan administrator:

- You cannot modify existing network groups
- Your network can be accessed from other domains and networks; for example, by anonymous users through the Terminal Server (you can create a NetMan group exclusively for remote users and assign permissions accordingly)
- Your network has groups that are not supported by NetMan (for example, if you are using Banyan Vines or a large Microsoft network with no domain controller).

You can create, edit, re-name, and delete NetMan user groups.

The example below shows a NetMan user group created specifically for the NetMan system: each of the four users are in different networks, but they are all NetMan administrators:

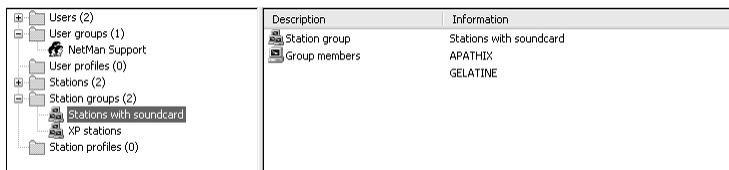


## NetMan Station Groups

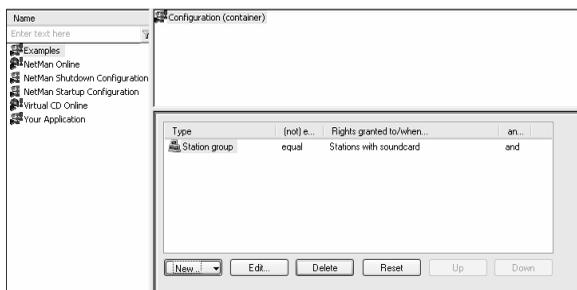
With NetMan, you can join workstations in groups.

This is a feature that is not available in network operating systems. There are a number of situations in which grouping workstations can be useful.

For example, some applications have specific requirements regarding the computer's internal hardware or peripheral devices:



If you have an application that requires a sound card, for example, you can create a „Sound Card“ group of workstations and limit the ‘execute’ permissions for the NetMan configuration to this group.



You can create, edit, re-name and delete NetMan station groups.

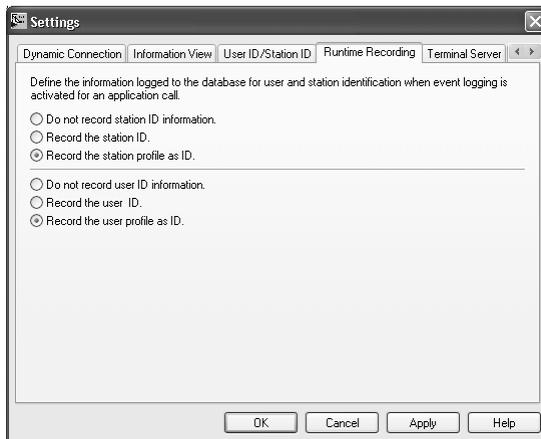
## NetMan User and Station Profiles

On the GLOBAL page of the *NetMan Settings*, you can define global settings for the starting language (if you have the *Language Module*) as well as startup and shutdown configurations for the NetMan system. These global settings can be modified for individual users and stations in the user and station databases.

As a rule, however, it is not for individual users or stations that you wish to define different settings, but for groups of users and stations. NetMan groups cannot be used for this purpose, because a given user or station can belong to any number of different groups.

To apply system configurations to a group of users, you need to work with **disjunct groups**: a user or a station can only belong to one such group. These groups are called **profiles**.

You can also use the user/station profile in place of the user/station ID as the identifier in NetMan data log and statistics program. This is configured on the RUNTIME RECORDING page of the NetMan Settings:

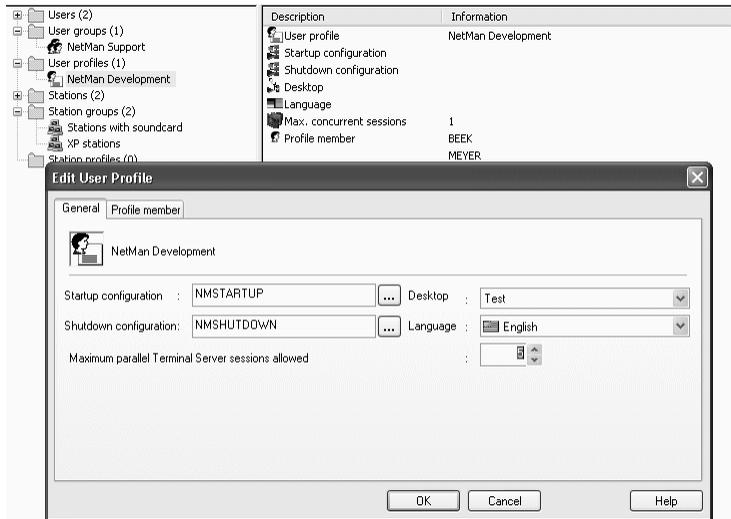


You can create, edit, re-name, and delete NetMan user and station profiles.

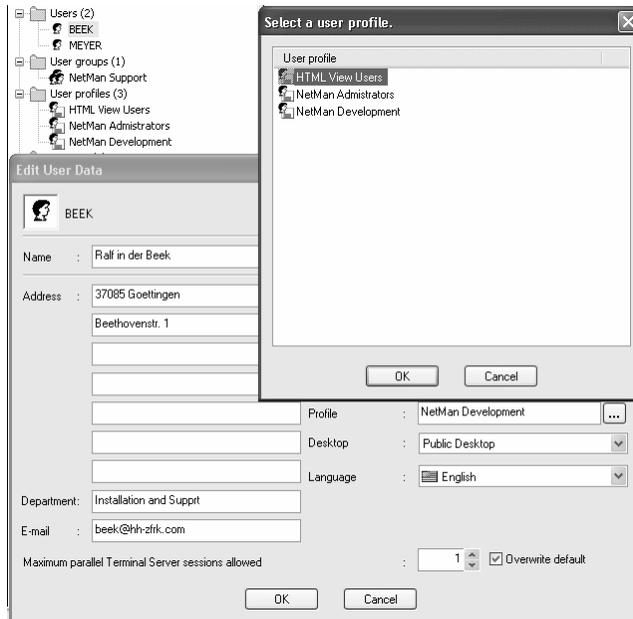
## User Profiles

The following preferences are defined in the user profile:

- Startup configuration
- Shutdown configuration
- Starting desktop
- Starting language (if you have the *Language Module*),
- Number of parallel terminal server sessions allowed (if you have the *Terminal Server Module*).



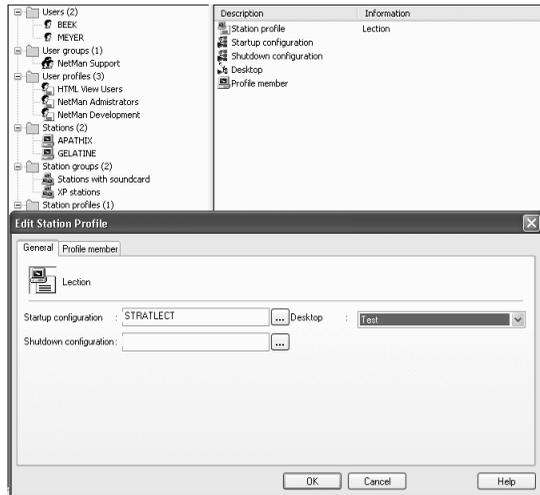
Belonging to a profile is a property of a user, and can be entered in the user database:



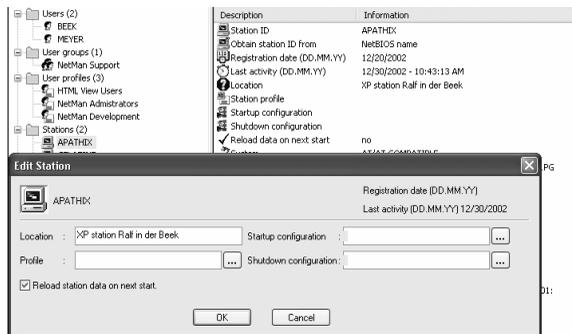
If you wish to add several users to a profile, however, it is easier to do this by editing the profile than by modifying each of the respective user data records. When you assign a user to a profile, any existing membership in another profile is overwritten automatically.

## Station Profiles

In the station profile you can define preferences for the startup and shutdown configurations:



Belonging to a profile is a property of a station, and can be defined in the station database:



If you wish to add several stations to a profile, however, it is easier to do this by editing the profile than by modifying each of the respective station data records. When you assign a station to a profile, any existing membership in another profile is overwritten automatically.



## 7. Statistical Analysis of Log Files

When you select the LOG DATA option in the Program action of a NetMan configuration, events involving that program are logged and can be analyzed with the NetMan STATISTICS program. There are a number of practical uses for these statistical evaluations, ranging from an overview of system use to an accounting of application usage. You can also create parallel-use spreadsheets to determine the number of licenses used by an application. This chapter describes the functions available in the Statistics module, and presents a practical demonstration using the log files in an existing NetMan installation.



### Tip

Refer to the on-line Help for detailed information on the numerous settings available in the Statistics program.



### Note

In the NetMan Settings you can define whether and how users and stations are identified in the event log (see “NetMan Settings: RUNTIME RECORDING Dialog Page” in chapter 4).

To view data in log files, run the **Record Database Viewer** in the SYSTEM ADMINISTRATION folder:

Record ID	Start date	Stop date	Start time	Stop time	User ID	Station ID
26794	26.08.2002	26.08.2002	11:13:13	11:13:33	ABCISI	210.96.220.110
26793	26.08.2002	26.08.2002	11:07:55	11:08:21	ABCISI	210.96.220.49
26792	26.08.2002	26.08.2002	10:42:48	10:43:43	ABCISI	210.96.220.49
26791	26.08.2002	26.08.2002	10:36:29	10:41:21	IMPBPC	99.176.5.30
26790	26.08.2002	26.08.2002	10:08:22	10:14:15	IMPBPC	99.176.5.30
26789	26.08.2002	26.08.2002	09:32:45	09:33:47	IMPBPC	99.176.5.30
26788	26.08.2002	26.08.2002	09:32:32	09:56:56	IMPBPC	99.176.5.30
26787	26.08.2002	26.08.2002	07:23:44	07:26:44	MPGV	222.174.202.68
26786	23.08.2002	23.08.2002	17:27:43	17:47:05	MPV-HD	99.149.21.101
26785	23.08.2002	23.08.2002	17:27:22	17:52:07	ABCISI	210.96.220.104
26784	23.08.2002	23.08.2002	16:46:32	16:48:32	ABCISI	210.96.220.217
26783	23.08.2002	23.08.2002	16:36:57	17:47:39	ARC IAO	703.251.5R.10R

This data forms the basis for evaluations performed by the NetMan Statistics program.

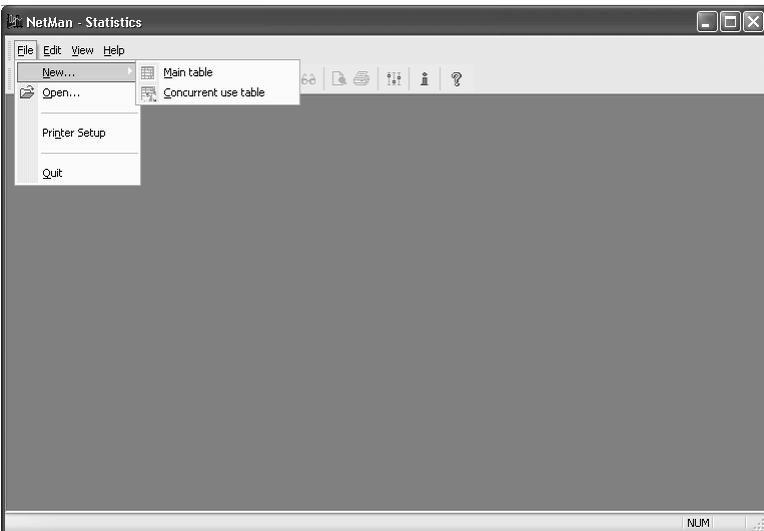
## Statistical Analysis with the NetMan Statistics Program

First, open the SYSTEM ADMINISTRATION folder in the NetMan desktop and start the Statistics program.



When you start the Statistics program, the main program window is opened. You can choose from two types of table in this window:

- Main table
- Table of concurrent use



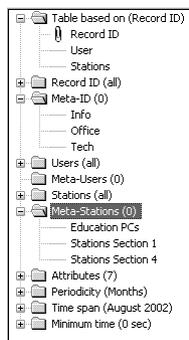
No spreadsheet is loaded when you start the statistics program. Under SETTINGS/SELECTION you can specify a type of spreadsheet to be loaded at program start.

## The Main Table

The main table offers the following selection options:

Under “Table based on...” you can define whether application call and usage data is calculated according to application, user or station. Depending on your selection, each data line in the main table shows the data on a single application, user or station.

You can group applications, users, or stations for purposes of statistical analysis under the selections “Meta-IDs”, “Meta-users” and “Meta-stations”. The results in the main table show the aggregate data under the defined group name as a data line. You can choose from defined *Attributes* to record additional information about application calls.



Attribute	Description
none	No attributes recorded
/AS	Terminal server session
/CC	Connection to client interrupted
/Link	Execution of a Hyperlink configuration
/MF	Mount error
/NE	Program could not be executed
/NL	No license available
/Test	Test call from the Management Console
/WL	License waiting period

If you select the /TEST attribute, for example, the record viewer shows which application calls were launched for test purposes only. You can also determine the periodicity and calculation period.

The *Minimum time* setting lets you define how long an application must be in use before its usage is included in your statistical analysis. If the “Microsoft Word” application runs for only 20 seconds, for example, it can be assumed that the program was not actually used in any meaningful way, so you may not wish to include these 20 seconds in your statistics. You can change the value entered here if desired.



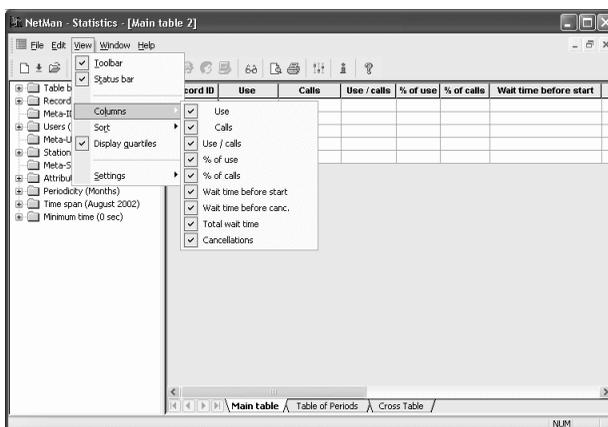
### Note

*If no license is available and the user cancels the call rather than waiting for a license, the call is recorded with a usage time of 0 seconds. If you wish to include such events in your statistical analysis, set MINIMUM TIME to “0 seconds.”*

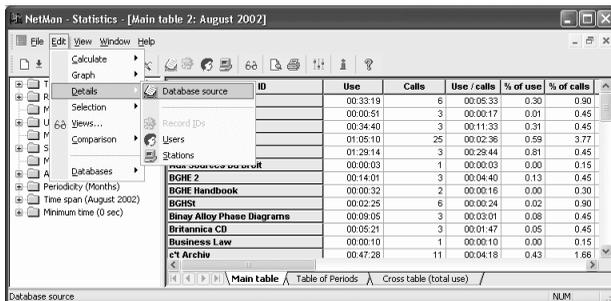
The main table shows the following values:

- Period of application use in hours:minutes:seconds
- Number of application calls
- Average use period per call (the SUM line shows the average use in square brackets, because this value was not arrived at through summation)
- Percentage of the use time of this application in relation to all application use
- Percentage of the application calls of the application in relation to all application calls
- Time spent waiting for a license before the application started (“/WL” attribute)
- Time spent waiting for a license before canceling the application call (“/NL” attribute)
- Total time spent waiting in line for a license (“/NL” and “/WL” attributes)
- Number of cancellations while waiting for licenses

You can adapt the table to your requirements by selecting which columns will be shown in the main table. To do this, select VIEW/COLUMNS:



You can choose *Record ID*, *User* or *Station* as the basis for calculation. Whichever you choose, you can view a calculation based on either of the other two elements by selecting „EDIT / DETAILED INFORMATION/...“ and the desired element.



Database source	ID	Use	Calls	Use / calls	% of use	% of calls
		00:33:19	6	00:05:33	0.30	0.90
		00:00:51	3	00:00:17	0.01	0.45
		00:34:40	3	00:11:33	0.31	0.45
		01:05:10	25	00:02:38	0.59	3.77
		01:29:14	3	00:29:44	0.81	0.45
		00:00:03	1	00:00:03	0.00	0.15
		00:14:01	3	00:04:40	0.13	0.45
		00:00:32	2	00:00:16	0.00	0.30
		00:02:25	6	00:00:24	0.02	0.90
		00:09:05	3	00:03:01	0.08	0.45
		00:05:21	3	00:01:47	0.05	0.45
		00:00:10	1	00:00:10	0.00	0.15
		00:47:28	11	00:04:18	0.43	1.65

You can also save any of these calculations as a special View of your data by selecting EDIT/VIEWS. A View can be activated at any time or loaded at program start.



## Note

When you select a View of a complete statistics period, the View is saved automatically. The data in this View is not deleted when you delete the original log files the View was based on. This means that these tables, once calculated, remain available for later analysis. Another advantage of saving Views is that it improves performance by accessing data that has already been calculated.



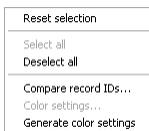
Views	Use / calls	% of use	% of calls
Workgroup	00:05:33	0.30	0.90
Workgroup1	00:00:17	0.01	0.45
	00:11:33	0.31	0.45
	00:02:38	0.59	3.77
	00:29:44	0.81	0.45
	00:00:03	0.00	0.15
	00:04:40	0.13	0.45
	00:00:16	0.00	0.30
	00:00:24	0.02	0.90
	00:03:01	0.08	0.45
	00:01:47	0.05	0.45
	00:00:10	0.00	0.15
	00:04:18	0.43	1.65



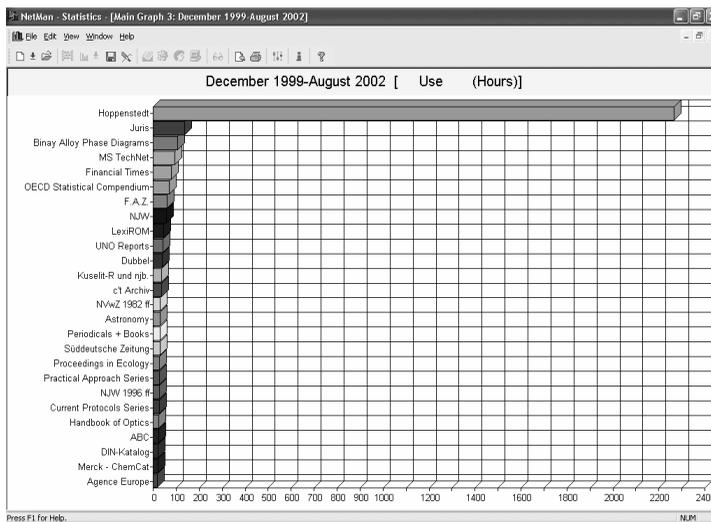




Because we have a very large volume of data available, we mark a selection of data records in the table. To save time when processing large amounts of data, you can have the assignment of colors for graphic output generated automatically. To do this, double-click in the Selection window and select **GENERATE COLOR SETTINGS** from the shortcut menu. To assign colors to individual record IDs, select the record ID, right-click on it to open the shortcut menu, and select **COLOR SETTINGS**.



Now we choose a suitable graph type for our data and generate graphic output:



## Note

Select **EDIT / GRAPH /** to define which values are represented in your graph.

The “Juris” labor law application was used the most. As can be seen in the **Use / CALL** column in the table on the previous page, the average duration of use per application call was by far the longest of any application.

Next, we sort the table by application call. We also activate the **DISPLAY QUARTILES** option, so we can recognize the highest and lowest values in the column at a glance.

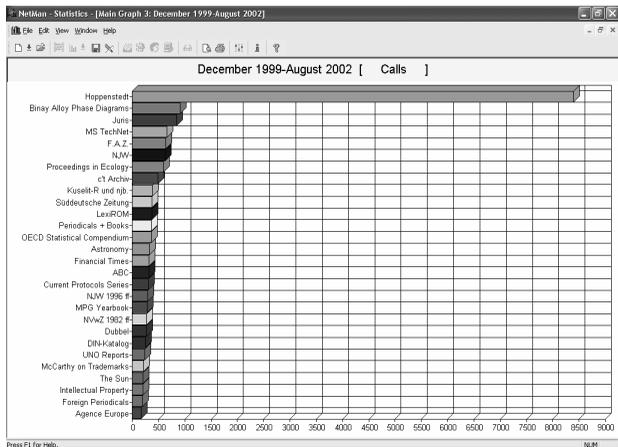
This option marks values with one of four colors, to differentiate the following categories:

- high values (=75 - 100% of the highest value)
- fairly high values (=50-74%),
- fairly low values (=25-49%),
- low values (=0-24%).

In the table below, you can tell at a glance which are the highest values in each of the columns (sorted by use):

Record ID	Use	Calls	Wait time before canc.	Total wait time	Cancellations
Hoppenstedt	2855.22.31	830	0.00.28	00.25.02	11
Juris	135.45.51	844	00.00.00	00.00.54	1
Binsy Alloy Phase Diagrams	107.02.57	911	00.00.00	00.00.00	0
MS Techlet	84.40.24	666	00.00.00	00.00.00	0
Financial Times	78.07.42	324	0.04.39	01.23.09	4
OECD Statistical Compendium	62.26.44	363	00.00.06	00.06.51	2
F.A.Z.	60.14.04	643	00.01.03	00.05.42	6
NJW	58.26.54	630	00.11.23	00.16.26	19
LexisOM	44.41.06	372	00.00.20	00.11.52	7
UNO Reports	2.31.12	270	00.00.00	00.04.47	0
Dubbel	40.36.43	270	00.00.00	00.00.00	2
Kuselt-R und rjb.	36.34.37	354	00.19.41	00.56.08	13
CT Archive	35.55.36	469	00.00.06	00.00.08	11
NWvZ 1982 ff	31.14.21	278	00.00.04	00.00.12	3
Astronomy	31.13.33	335	00.00.00	00.08.29	0
Periodicals + Books	29.58.96	369	00.00.02	00.04.50	3
Süddeutsche Zeitung	28.16.25	362	00.00.11	00.00.11	2
Proceedings in Ecology	28.39.10	591	00.00.00	00.00.00	0
Practical Approach Series	28.38.26	159	00.02.10	00.09.27	4
NJW 1996 ff	27.36.54	291	00.00.45	00.06.45	2
Current Protocols Series	25.66.53	308	00.02.42	00.31.56	9
Handbook of Optics	25.33.05	65	00.00.17	00.11.23	1
ABC	24.40.09	398	00.00.25	00.01.04	2
DIN-Katalog	21.26.58	247	00.00.00	00.00.01	1
Merck - ChemCat	20.35.39	106	00.00.14	00.00.21	1
Agence Europe	18.31.03	174	00.00.08	00.00.08	2
Foreign Periodicals	17.19.21	192	00.03.59	00.13.50	6
NJW 1994 ff	16.62.30	149	00.00.00	00.21.01	0
Goettingen	14.41.02	142	00.00.00	00.00.00	0
Dictionary	14.10.58	103	00.00.00	00.00.00	0
MPG Yearbook	13.62.03	280	00.00.00	00.00.00	0
SoFD	13.36.47	109	00.00.00	00.00.00	0
DGWS	13.16.07	118	00.00.00	00.00.00	0
McCarthy on Trademarks	12.41.32	221	00.01.07	00.01.12	7
The Sun	12.20.19	208	00.00.14	00.05.24	1
Intellectual Property	12.14.00	203	00.00.41	00.00.41	4
JURIS-Red	12.12.58	103	00.06.42	00.06.42	2

Sorted by number of application calls, the graph looks like this:



Here we have sorted the table by time spent waiting for a license:

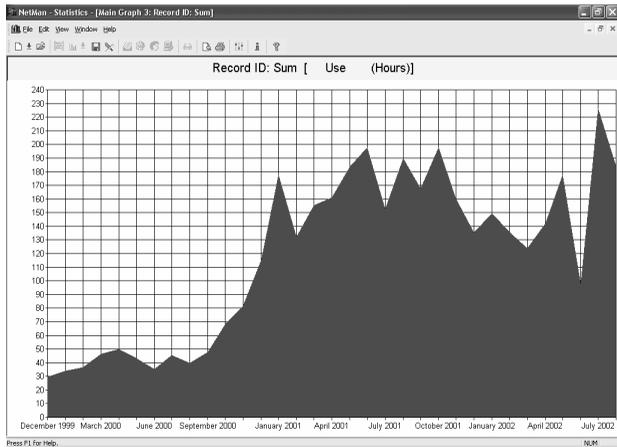
Record ID	Use	Calls	Wait time before canc.	Total wait time	Cancellations
<b>Financial Times</b>	76,07,42	324	00:44:59	01:23:09	4
<b>Russell-R and sub.</b>	26:34:37	394	00:19:41	00:46:09	19
<b>Current Protocols Series</b>	26:56:53	308	00:02:42	00:31:56	9
<b>Hogpensoft</b>	2268:22:31	6390	00:06:28	00:26:02	11
<b>NW 1994 ff</b>	16:53:30	149	00:00:00	00:21:09	0
<b>NW</b>	56:26:54	630	00:11:03	00:16:26	19
<b>Foreign Periodicals</b>	17:19:21	192	00:03:51	00:13:50	5
<b>LeaWeb</b>	46:41:06	372	00:00:20	00:11:52	3
<b>Heute /Zürcher Zeitung</b>	08:30:34	110	00:00:00	00:11:44	0
<b>JURIS Low</b>	12:08:34	102	00:06:55	00:11:27	2
<b>Handbook of Optics</b>	25:33:05	85	00:00:17	00:11:23	1
<b>Practical Approach Series</b>	26:38:26	159	00:00:00	00:09:27	4
<b>Astronomy</b>	31:13:33	335	00:00:00	00:08:29	0
<b>OECB Statistical Compendium</b>	68:28:44	369	00:00:08	00:06:51	2
<b>JURIS RCN</b>	18:21:56	109	00:06:42	00:06:42	2
<b>RefWorld</b>	03:55:51	68	00:05:49	00:05:49	1
<b>F.A.Z.</b>	60:14:04	643	00:01:03	00:05:42	6
<b>NW 1981 - 89</b>	11:12:05	146	00:00:00	00:05:27	0
<b>The Sun</b>	12:20:19	206	00:00:00	00:05:24	1
<b>UNO Reports</b>	42:11:05	239	00:00:00	00:04:56	0
<b>Periodicals + Books</b>	28:59:56	369	00:01:02	00:04:50	3
<b>Business Law</b>	02:41:00	115	00:00:58	00:04:18	2
<b>Treaties with Canada</b>	07:49:49	72	00:00:00	00:03:28	1
<b>PAVIS</b>	06:25:17	164	00:02:02	00:02:18	7
<b>MEDLINE</b>	01:10:21	61	00:01:50	00:01:50	2
<b>Biology and Ecology</b>	02:50:49	53	00:00:00	00:01:30	0
<b>Stats Bund 2000</b>	06:04:23	53	00:00:00	00:01:17	0
<b>OSCE - Documents</b>	03:26:26	53	00:00:00	00:01:13	0
<b>McCarthy on Trademarks</b>	12:41:32	291	00:01:07	00:01:12	7
<b>ABC</b>	24:40:09	324	00:00:25	00:01:04	2
<b>Juris</b>	136:48:51	844	00:00:00	00:00:54	1
<b>WB</b>	06:59:25	105	00:00:00	00:00:46	0
<b>NW 1996 ff</b>	27:38:54	291	00:00:45	00:00:45	2
<b>Intellectual Property</b>	12:14:00	203	00:00:41	00:00:41	4
<b>Library Issues</b>	01:39:59	59	00:00:41	00:00:41	1
<b>PC Professional Archive</b>	02:58:32	70	00:00:16	00:00:29	2
<b>Social Security Worldwide</b>	12:08:59	96	00:00:35	00:00:35	1

The Financial Times is at the top of the list.

In the table of periods for a given line, the SUM line shows the total use of all applications in each period, which is useful for detecting trends:

Period	Use	Calls	Wait time before canc.	Total wait time	Cancellations
<b>December 1999</b>	29:37:02	547	00:00:06	00:07:59	1
<b>January 2000</b>	34:04:06	594		00:31:10	1
<b>February 2000</b>	36:52:37	598			
<b>March 2000</b>	46:34:22	489		00:05:49	
<b>April 2000</b>	49:55:37	599		00:04:44	
<b>May 2000</b>	43:22:59	362		00:00:09	
<b>June 2000</b>	36:36:20	272		00:00:09	1
<b>July 2000</b>	46:24:39	513			
<b>August 2000</b>	39:57:11	496	00:01:47	00:22:55	3
<b>September 2000</b>	47:43:18	405	00:01:47	00:13:23	6
<b>October 2000</b>	66:22:33	711	00:00:15	00:00:28	4
<b>November 2000</b>	81:43:47	766	00:06:24	00:37:54	12
<b>December 2000</b>	115:01:12	901	00:13:42	00:39:54	15
<b>January 2001</b>	177:56:33	1631	00:07:44	00:08:56	21
<b>February 2001</b>	152:29:20	1034	00:01:40	00:21:17	4
<b>March 2001</b>	155:23:50	1018	00:43:34	00:50:49	8
<b>April 2001</b>	160:59:58	1065	00:00:11	00:18:35	2
<b>May 2001</b>	163:35:29	1106	00:03:44	00:04:17	9
<b>June 2001</b>	190:05:10	1188	00:00:01	00:17:21	9
<b>July 2001</b>	152:49:58	1161	00:00:33	00:02:02	3
<b>August 2001</b>	190:00:36	1270	00:00:56	00:00:56	6
<b>September 2001</b>	167:52:46	891	00:00:37	00:00:46	6
<b>October 2001</b>	197:45:07	1130	00:02:26	00:27:47	4
<b>November 2001</b>	160:19:17	892	00:00:37	00:05:01	2
<b>December 2001</b>	136:40:34	629			1
<b>January 2002</b>	146:52:50	901	00:00:26	00:03:09	6
<b>February 2002</b>	136:54:12	715	00:04:38	00:10:57	10
<b>March 2002</b>	123:54:00	654	00:13:21	00:13:21	3
<b>April 2002</b>	142:12:18	944	00:01:10	00:13:00	2
<b>May 2002</b>	176:09:58	993	00:00:05	00:08:28	4
<b>June 2002</b>	97:47:34	569	00:01:11	00:02:23	3
<b>July 2002</b>	226:23:46	944	00:05:43	00:14:43	2
<b>August 2002</b>	184:18:35	663	00:02:07	00:02:07	2
<b>Sum</b>	<b>3252:57:31</b>	<b>26126</b>	<b>02:01:56</b>	<b>06:26:21</b>	<b>148</b>

The graphic representation of usage distribution over time periods (a different type of graph was chosen for this example) shows a significant increase in usage since October 2000.



The cross table below shows the periodic distribution of the Use column for all applications (due to the large volume of data, only an excerpt can be shown here):

Record ID	February 2002	March 2002	April 2002	May 2002	June 2002	July 2002	August 2002	Sum
Agence Europe	00:23:03	00:16:50	00:59:30	00:46:18	03:34:39	01:37:41	00:00:51	1631:03
ALC_BC		00:06:31		00:00:45				00:19:47
ALC_HO								00:08:31
Algebra	00:02:44	00:06:02	00:01:07	00:06:57	00:03:05	00:00:17	00:34:40	02:26:51
Annual Review Index								00:01:59
Astronomy	00:41:06	00:03:27	00:12:17	00:30:30	00:06:51	00:06:34	01:05:10	313:53
ATCC Catalogs								00:00:53
AUA	00:00:06	00:20:39	00:00:12	00:00:27	00:07:30	03:10:11	01:29:14	08:56:15
Aux Sources Du Trait								00:04:25
BRNAV	00:00:17	00:01:18	00:01:48	00:01:25	00:11:00			02:16:40
BGHE 2	00:06:52	01:16:48	00:20:21	00:03:16	00:14:28	01:21:32	00:14:01	03:42:44
BGHE Handbook	00:01:29	00:00:19	00:01:03	00:02:29				00:08:20
BGHE	00:02:27	00:25:25	01:42:22	00:27:01	02:57:27	00:03:13	00:02:25	133:67
Binary Alloy Phase Diagrams	00:26:38	02:30:21	01:49:37	04:39:52	00:56:38	02:45:37	00:09:05	103:27
Biochemistry								01:25:44
Biology and Ecology	00:15:37		00:03:35	00:22:15				02:50:49
Britannica CD	02:18:15	00:02:22	00:02:33	00:03:35			00:06:21	08:21:58
British BCI-CD	01:26:56	01:06:03	00:00:32	00:01:06	00:04:48	00:00:19	00:02:25	08:26:20
Business Law		00:02:17	00:00:37		00:01:32	00:24:18	00:00:10	06:41:00
CV Archive	00:05:24	01:06:20	00:06:02	02:21:59	02:26:37	00:00:34	00:47:28	36:55:36
CASII		00:56:44	00:02:13					16:26:29
Codices				00:00:32			00:00:42	02:06:43
Computing		00:17:15		00:00:13		00:45:05	00:00:26	03:18:47
Copyright	00:03:47	00:05:19	00:13:23	00:02:12	00:04:39	00:03:26		04:01:20
Corpus Augustinianus Giesense								01:43:51
Current Protocols Series	00:19:59	00:00:09	02:25:56	00:20:31	00:37:50	00:46:30		25:06:53
Código de Derecho Interno								00:12:03
Código de Derecho Internacional	00:00:25							00:07:43
Dictionary	00:59:27	00:00:41	01:15:13	00:01:12		00:13:19	00:12:38	14:10:58
Dirk-Katalog	00:27:23	01:10:01	01:07:22	01:32:29	01:38:45	01:25:19	00:36:13	24:26:58
Dubnet	00:07:42	01:09:23	00:30:09	01:45:02	01:12:06	01:25:42	00:15:19	06:36:43
Duden	00:28:59		00:17:06	00:10:19		00:00:50		05:04:26
Ecology and Biology		00:01:08		00:00:41				02:10:46
Ergonomics	00:01:29	00:02:19			00:01:56	00:03:16		03:10:16
Espace Legal			00:01:34			00:01:21	00:01:16	00:53:19
Fit-Kernno			00:03:47	00:12:14		00:07:46		01:48:34

With the default settings, the cross table calculates the absolute calls, sorted by record ID, for the selected period (Record ID/Period).

Right-click anywhere on the table to access the expanded functions available for cross tables. You can compare the record IDs for users or stations for the six values chosen.

Record ID	February 2002	March 2002	April 2002	May 2002	June 2002	July 2002	August 2002	Sum
Agence Europe	00:23.03	00:19.50	00:59.30	00:46.18	03:34.39	01:37.41	00:00.51	16319.03
ALC_BIC		00:05.31		00:00.45				06:19.07
ALC_ID								00:00.34
Algebra	00:02.44	00:08.02	00:01.07	00:08.57	00:03.05	00:00.17	00:34.10	62:28.21
Annual Review Index								06:01.59
Astronomy	00:41.06	00:03.27	00:12.17	00:30.30	00:06.51	00:08.34	01:05.10	313:13.33
ATCC Catalogs								00:19.53
AUA	00:00.06	00:20.39	00:00.12	00:00.27	00:07.30	03:10.11	01:29.14	08:55.15
Aux Sources Du Inuit								00:42.25
BIFHW		Global Settings...	30:01.18	00:01.49	00:01.25	00:11.00		62:14.49
BIGHE 2		Table Settings...	01:16.48	00:20.21	00:03.16	00:14.28	01:21.32	03:42.44
BIGHE Handbook			20:03.19	00:01.03	00:02.29	00:00.32		06:09.29
BIGHE			30:25.25	01:42.22	00:27.01	02:57.27	00:03.13	135:19.07
Binary Alloy Phase Diagrams		Create report		04:31.52	00:58.38	02:45.37	00:09.05	107:42.57
Biochemistry		Calculations						01:25.44
Biological and Ecology		00:15.37				Record ID/Periods		62:50.49
Birnamica CD		00:19.15				Record ID/Users		09:15.50
British BCH-CD	Total usage					Record ID/Stations	00:02.25	06:28.29
Business Law	Use as percentage of total usage				00:01.52	00:24.18	00:00.10	09:41.88
CH Archive					02:21.59	00:28.37	00:00.34	04:47.28
CASSI	Use as percentage of total usage for record ID							16:28.29
Codices	Total calls				00:00.13	00:45.05	00:00.42	62:18.47
Computing	Calls as percentage of total calls				00:02.12	00:04.38	00:03.26	01:15:10
Copyright								01:43.51
Corpus Augustinianum Graese	Calls as percentage of total calls for record ID							25:55.53
Current Protocols Series								00:12.03
Código de Derecho Intern.				00:20.31	00:37.50	00:48.30		06:07.43
Código de Derecho Internacional								14:19.58
Dictionary	00:59.27	00:03.41	01:15.13	00:01.12		00:13.19	01:12.38	212:26.58
DIN Catalog	00:27.23	01:10.01	01:07.22	01:32.29	00:38.45	01:28.19	00:28.13	46:36.43
Dubbel	00:07.42	01:00.23	00:30.03	01:45.02	00:12.08	01:25.42	00:18.19	65:04.26
Duden	00:29.59		00:17.06	00:10.19		00:00.50		62:18:45
Ecology and Biology		00:01.08		00:06.41				03:18:16
Ergonomy	00:01.29	00:02.19			00:01:56	00:03:16		06:53:19
Espace Legal				00:01.34		00:01.21	00:01:16	01:19:34
FL Review			00:09.47	00:17.14		00:01:48		

All of the calculations demonstrated above for applications can also be made based on users or stations:

Table based on (User)

- Record ID
  - User
  - Stations
- Record ID (all)
  - Meta-ID (0)
  - Users (all)
  - Meta-Users (0)
  - Stations (1992)
  - Meta-Stations (0)
  - Attributes (7)
  - Periodicity (Months)
  - Time span (December 1999-August 2002)
  - Minimum time (0 sec)

Table based on (Stations)

- Record ID
  - User
  - Stations
- Record ID (all)
  - Meta-ID (0)
  - Users (all)
  - Meta-Users (0)
  - Stations (1992)
  - Meta-Stations (0)
  - Attributes (7)
  - Periodicity (Months)
  - Time span (December 1999-August 2002)
  - Minimum time (0 sec)

Sorted by applications calls, the 'Users' table shows the following..

User	Use	Calls	Wait time before conc.	Total wait time	Cancellations
ABC	1377.2605	5246	00:00:14	00:16:26	6
IMPV-ID	279.4854	3515	00:12:58	00:54:29	22
IBIL_CDINH	76.4745	2899	00:01:38	00:07:46	7
IMPPOC	305.5250	2496	00:05:59	01:10:43	21
IMPAP	167.3443	2362	00:18:01	00:48:23	45
IMPFG	256.1934	1708	00:46:20	01:42:39	15
ABCPSA	377.4525	1687	00:04:14	00:09:41	8
BSCRM	161.0711	1626	00:32:29	01:19:04	22
ABCISI	305.0159	953	00:00:00	00:00:00	0
IMPMP	109.2852	837	00:00:00	00:00:00	0
IMPGGV	134.5941	824	00:00:00	00:00:54	1
IBIL	74.3247	631	00:00:06	00:00:11	1
ABC_ZV	58.4646	212	00:00:00	00:00:00	0
ABC_ADO	58.1846	168	00:00:00	00:00:00	0
IBIL_COTESTUSER	03.3553	153	00:00:00	00:00:00	0
DEMOGR	06.1544	91	00:00:00	00:00:00	0
ABC_ILI	15.4746	81	00:00:00	00:00:00	0
ABC_ILI	36:38:06	80	00:00:00	00:00:00	0
ABC_IPK	15.3624	65	00:00:00	00:00:00	0
IMPWC	02.4111	65	00:00:00	00:00:00	0
ABC_GMB	11.4107	51	00:00:00	00:00:00	0
ABC_IS	16.2655	47	00:00:00	00:00:00	0
ABC_IFT	06.5350	24	00:00:00	00:00:00	0
ABC_IPAM	01.5542	22	00:00:00	00:00:00	0
ABC_IBR	13:11:46	22	00:00:00	00:00:00	0
ABC_MNS	06:52:17	22	00:00:00	00:00:00	0
ABC_IGD	06:44:40	21	00:00:00	00:00:00	0
ABC_IPP	02:44:26	17	00:00:00	00:00:00	0
IBIL_COPPUHSTETA	00:44:47	16	00:00:00	00:00:00	0
ABC_IPT	02:09:24	14	00:00:00	00:00:00	0
ABC_HST	06:39:27	14	00:00:00	00:00:00	0
ABC_HRAVK	13:24:19	13	00:00:00	00:00:00	0
ABC_ESK	02:09:42	10	00:00:00	00:00:00	0
ABC_IPF	00:43:41	9	00:00:00	00:00:00	0
ABC_PNKI	01:46:20	8	00:00:00	00:00:00	0
ABC_MMI	01:19:04	8	00:00:00	00:00:00	0
ABC_IZM	00:11:00	7	00:00:00	00:00:00	0

...and the 'Stations' table looks like this:

Stations	Use	Calls	Wait time before canc.	Total wait time	Cancellations
15.225.136.111	45:40:19	2379	00:00:57	00:05:06	6
222.474.132.60	91:35:19	1226	00:05:48	00:09:30	8
15.225.136.122	26:40:24	556	00:00:34	00:04:48	2
202.76.11.251	05:24:21	494	00:00:09	00:02:42	2
99.148.15.499	14:54:54	491	00:00:14	00:11:50	2
206.140.99.205	42:43:59	305	00:00:00	00:00:00	1
202.76.11.173	05:48:57	254	00:00:00	00:00:00	0
202.76.299.231	21:33:02	223	00:00:00	00:00:00	0
224.199.12.122	10:04:02	218	00:00:00	00:00:07	0
206.140.99.110	07:00:16	210	00:00:00	00:00:00	0
99.148.12.82	06:02:29	207	00:00:00	00:00:00	0
222.474.135.160	31:05:01	205	00:00:00	00:00:00	0
202.76.11.177	09:54:29	181	00:00:06	00:00:22	2
221.124.250.144	14:29:03	173	00:00:08	00:00:08	1
202.76.11.102	19:04:10	156	00:00:00	00:00:05	0
202.76.11.86	21:21:43	151	00:00:00	00:00:00	0
223.84.219.92	05:31:31	151	00:00:00	00:00:00	0
223.84.219.89	05:59:38	146	00:00:16	00:00:16	3
15.225.136.96	00:57:52	134	00:00:00	00:00:00	0
202.76.100.118	31:55:43	133	00:00:00	00:00:00	0
202.76.215.169	37:28:39	132	00:00:00	00:00:00	0
223.84.219.109	10:41:33	112	00:00:00	00:00:00	0
99.148.15.104	11:00:47	112	00:00:12	00:01:33	1
15.225.136.91	02:32:42	110	00:00:00	00:00:00	0
202.76.299.1	02:32:06	109	00:00:00	00:00:19	0
99.148.12.44	24:19:37	107	00:00:00	00:00:06	0
222.474.132.14	03:55:45	106	00:04:34	00:04:34	15
202.105.40.196	08:36:32	102	00:00:00	00:00:00	0
210.86.220.49	20:16:51	101	00:00:00	00:00:00	0
221.124.250.102	05:27:02	96	00:00:19	00:12:42	3
222.174.132.1	06:56:46	92	00:00:21	00:00:21	2
99.148.12.59	05:28:44	90	00:00:00	00:00:00	0
99.148.15.107	09:59:31	89	00:00:00	00:00:00	0
210.87.80.141	14:07:56	88	00:00:00	00:00:00	0
99.148.14.157	07:25:40	88	00:00:26	00:00:26	1
222.174.135.163	21:20:16	87	00:00:00	00:00:00	0
202.105.18.233	14:46:04	84	00:00:00	00:00:00	0

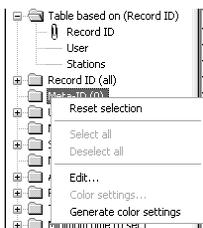
The calculations can be made not only according to all users, stations or applications, but also for selected

- applications,
- users,
- stations, and
- attributes.

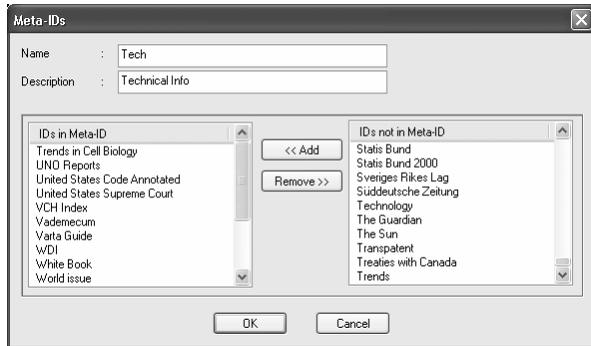
Furthermore, you can change the periodicity (quarterly, half-yearly, yearly or none), select different time spans, or set the minimum time to another value.

For the last demonstration, we shall generate calculations for Meta-users, Meta-stations and Meta-IDs (applications). These give less detail, and provide a clear overview of the selected period.

To do this, we first define groups of applications by right-clicking on *Meta-IDs* for a shortcut menu, from which we select Edit:



In this window we group our applications:



and then repeat the calculation, this time based on our new Meta-IDs:

Record ID	Use	Calls	Wait time before canc.	Total wait time	Cancellations
Office	17:10:59	565	00:02:07	00:02:07	2
Info	04:15:59	61	00:00:00	00:00:00	0
Tech	08:51:38	37	00:00:00	00:00:00	0
<b>Sum</b>	<b>14:18:35</b>	<b>663</b>	<b>00:02:07</b>	<b>00:02:07</b>	<b>2</b>

Next we group our stations:

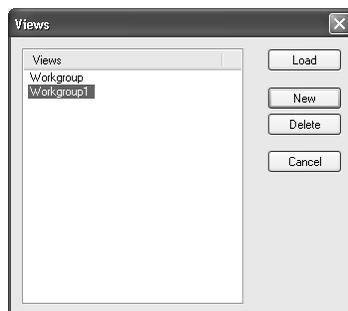
Stations	Use	Calls	Wait time before canc.	Total wait time	Cancellations
Stations Section 1	16:31:17	505	00:02:07	00:02:07	2
Stations Section 4	1:28:43	61	00:00:00	00:00:00	0
Education PCs	02:38:35	77	00:00:00	00:00:00	0
<b>Sum</b>	<b>14:18:35</b>	<b>663</b>	<b>00:02:07</b>	<b>00:02:07</b>	<b>2</b>

**Tip**

*To calculate the total usage of your NetMan system, group all applications in a Meta-ID and calculate the concurrent use spreadsheet for this Meta-ID.*

*If you have common licenses for multiple applications, you can group these applications in a Meta-ID to calculate the concurrent use of these licenses.*

We want to document statistical analyses for the station aggregates every month from now on, so we save the definition created in the “Selection” window as a **View**



When you save a **View**, the currently selected element is saved in the **View** definition. This has the following advantages:

- Complex combinations of “Selections” can be re-created by loading the corresponding **View**.
- Periods that were already calculated and stored in a **View** are loaded when a later calculation includes the same periods, which means the calculation is that much faster.
- Before the data in a log file is deleted, any periods in a **View** that had not been processed up to that point are calculated.
- Data in **Views** is still available for later processing even after the original log file has been deleted.

As a final evaluation, we now create a concurrent use table to obtain additional information about the use of licenses. Again, we have selected a limited number of record IDs to reduce the amount of data processed:

Record ID	Licenses	Max	Days	Duration	Max - 1	Days	Duration	Max - 2	Da
Hoppenstedt	9	9	1	00:02:54	7	1	00:01:12	6	2
Binary Alloy Phase Diagrams	3	3	2	00:07:03	2	33	00:21:11	1	271
Proceedings in Ecology	3	3	2	00:04:27	2	27	00:18:41	1	133
Vademecum	3	3	1	00:08:39	2	1	00:09:38	1	12
Juris	3	3	1	00:03:01	2	28	00:14:31	1	232
NW	3	3	1	00:02:00	2	12	00:28:58	1	196
MPC Yearbook	3	3	1	00:01:11	2	3	00:04:42	1	96
OECD Statistical Compendium	3	3	1	00:00:50	1	147	03:07:09	0	0
MS TechNet	2	19	00:31:45	1	251	01:37:39	0	0	0
ct Archiv	2	12	00:07:46	1	172	01:16:28	0	0	0
WIKI-Katalog	2	9	00:07:52	1	106	01:11:30	0	0	0
Dubbel	2	7	00:06:32	1	151	01:33:01	0	0	0
Stiddeutsche Zeitung	2	6	00:06:32	1	106	00:52:32	0	0	0
European Communities	2	3	00:06:44	1	46	01:28:36	0	0	0
IX Archiv	2	3	00:03:38	1	67	00:17:20	0	0	0
McCarty on Trademarks	2	3	00:03:22	1	46	03:47:10	0	0	0
The Sun	2	3	00:02:20	1	106	00:53:05	0	0	0
F.A.Z.	2	3	00:01:43	1	219	01:45:28	0	0	0
SoFID	2	2	00:19:54	1	49	01:31:19	0	0	0
Astronomy	2	2	00:04:49	1	155	01:12:02	0	0	0
Treaties with Canada	2	2	00:03:41	1	23	00:57:31	0	0	0
Neue Ztircher Zeitung	2	2	00:01:52	1	44	00:44:22	0	0	0
Periodicals - Books	2	2	00:01:32	1	175	00:57:26	0	0	0
Gazette du Palais	2	2	00:00:46	1	34	01:28:23	0	0	0
Statis Bund 2000	2	1	00:12:35	1	27	00:57:07	0	0	0
World Issue - 2	2	1	00:06:30	1	13	00:09:14	0	0	0
NW 1998 II	2	1	00:06:04	1	134	01:36:59	0	0	0
Springer Katalog	2	1	00:04:20	1	12	00:21:23	0	0	0
Current Protocols Series	2	1	00:03:29	1	128	01:19:13	0	0	0
CASSI	2	1	00:03:14	1	28	00:32:08	0	0	0
ABC	2	1	00:02:46	1	175	01:06:57	0	0	0
Foreign Periodicals	2	1	00:01:36	1	96	00:51:15	0	0	0
White Book	2	1	00:01:16	1	22	00:23:12	0	0	0
LexiROM	2	1	00:00:56	1	176	01:30:45	0	0	0
Financial Times	2	1	00:00:34	1	129	03:51:40	0	0	0
WMI	2	1	00:00:20	1	30	00:33:41	0	0	0

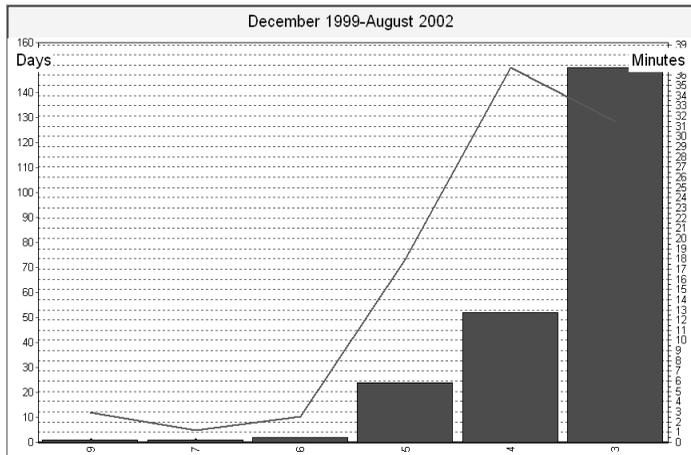
The LICENSES column shows the number of licenses currently configured for the application. This usually defines the limits for parallel use.

The MAX, DAYS and DURATION columns belong together as a block: MAX shows the highest number of parallel users, DAYS the number of days on which this level was reached, and DURATION the longest period during which this number of licenses was used simultaneously.

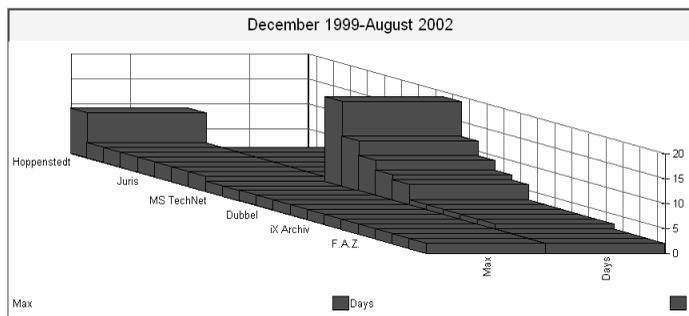
The subsequent columns show the same data for each of the next 5 lower simultaneous-use values.

As the table above shows, the number of licenses available for "SoFID" was always sufficient. For the "Hoppenstedt" company database application, however, all available licenses were in use on one particular day; if a tenth user had attempted to launch this application at that time, they might have had to wait almost 3 minutes for a license to become available. Under Max-1, however, we see that the application was called by multiple users concurrently only 7 times on one day and, as shown under Max-2, only 6 times on 2 days. Additional user licenses for this application might be handy, but are not urgently required.

When we select one line of the spreadsheet and generate a bar graph based on these values, the height of the bars shows the number of days on which the value occurred. The superimposed curve gives the duration in hours.



The following graph gives an overview of all applications that were used by more than one user simultaneously at least once:



This graph was created by selecting the applications and then activating **EDIT / GRAPH / MAXIMUM PARALLEL USE (FOR ALL IDS)**.

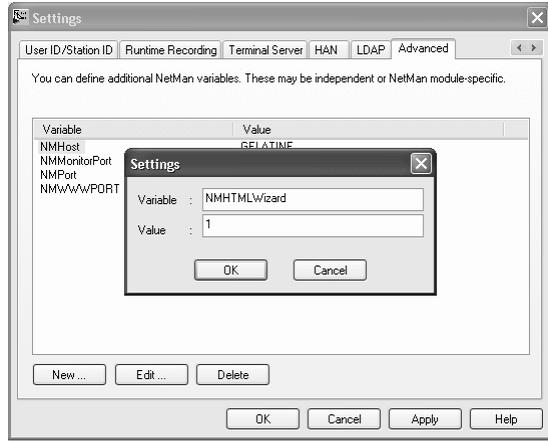


## 8. HTML Documents as User Interfaces

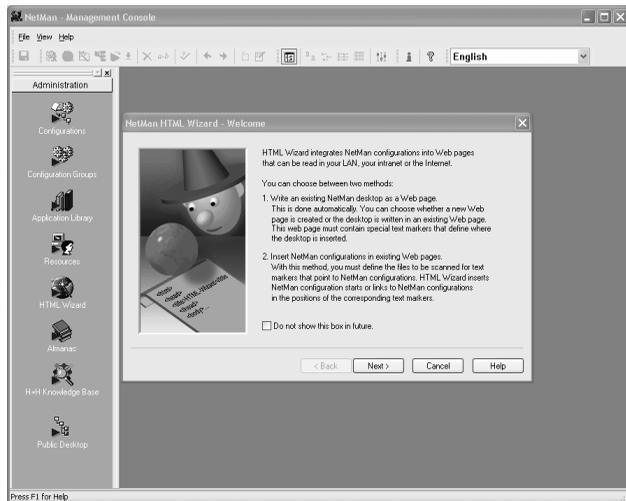
### Introduction

The *NetMan HTML Wizard* lets you create HTML documents containing NetMan desktops and configurations, and is an HTML-based alternative to the NetMan Client as user interface.

The HTML Wizard is not active immediately following installation of your NetMan system, since it might not be needed or wanted in all cases. If you have the HTML View Module, for example, you do not need HTML Wizard at all. To activate HTML Wizard, run the NetMan Settings program and configure the following:



Once it has been activated, HTML Wizard is available the next time you run the Management Console:





The HTML Wizard is designed to automate the procedure for creating Web-based access to NetMan application calls in intranets and the Internet. The following components are required for the production of a functional Web-based solution:

1. The application calls to be presented in Web pages must be defined in NetMan databases.
2. The HTML layout must be defined (either a custom layout template or the sample templates included with HTML Wizard).

With these components, the HTML Wizard can generate pages containing your Web-based services fully automatically. This is particularly useful for:

- NetMan administrators who do not wish to get involved in designing Web pages, but do want to offer applications through the Web
- NetMan administrators who wish to reduce the amount of work involved in creating Web pages

In comparison with NetMan HTML View and the NetMan (Windows-based) Client, HTML Wizard does have a few disadvantages:

- The pages created are static; in other words, if data is modified in the NetMan databases, new pages must be generated to incorporate the changes
- The Web page presents all NetMan configurations released for Web-based access to all users; user or station permissions are not evaluated until a user attempts to launch an application call. In some cases, permissions that restrict user access can be implemented only by configuring separate desktops and presenting these in different HTML pages.
- When the Citrix WebClient is used, the number of application licenses available cannot be determined until the Citrix session has been started.

The first step is to decide how you want to use HTML Wizard. If you do not wish to design your own Web pages, but do want to make your NetMan application calls available to your users for access through Web browsers – whether in the LAN, over an intranet or through the Internet – HTML Wizard can create the Web pages you need fully automatically by converting your NetMan Client desktop to HTML. It even includes the interface required to start the applications from a browser, whether you use ICA or NM link files, plug-ins, ActiveX controls, JavaScripts, or others. This is the first of the two methods described below. The second method, described subsequently, is to have the HTML Wizard insert individual NetMan configurations rather than entire desktops.

**Method 1: Presenting Complete NetMan Desktops in Web Pages**

You can create a NetMan Client Web page anywhere on your Web server, or have it written into an existing Web page. In the latter case, you must mark the position in the existing page where you want the HTML Wizard to insert the NetMan desktop:

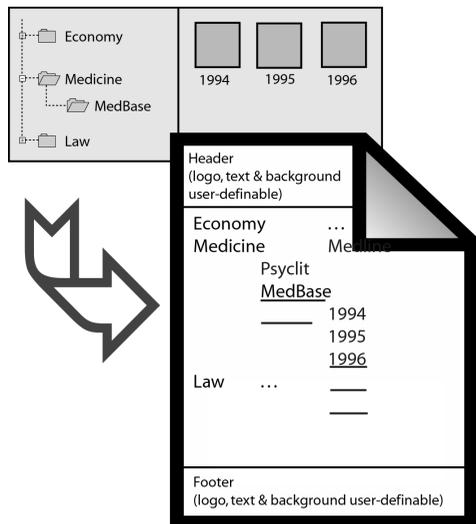
HTML-Wizard inserts desktop here...

```

<html>
...
any HTML code
<!--@nm_desktop_complete-->
<!--@nm_desktop_end-->
...
</html>

```

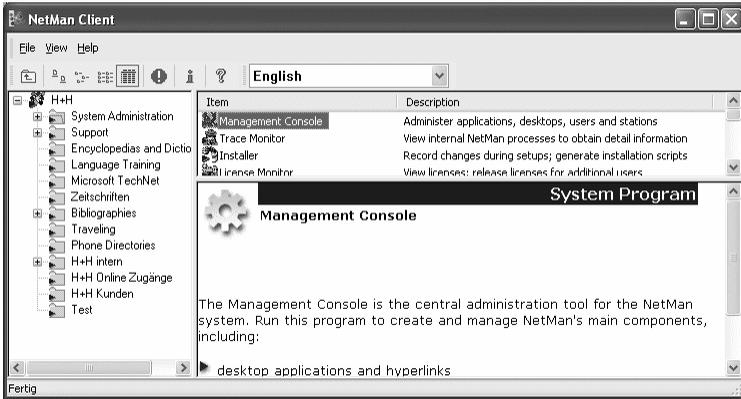
In both cases, a complete NetMan desktop is inserted in the HTML file. The diagram below shows how the desktop structure is displayed when it has been converted to HTML text:



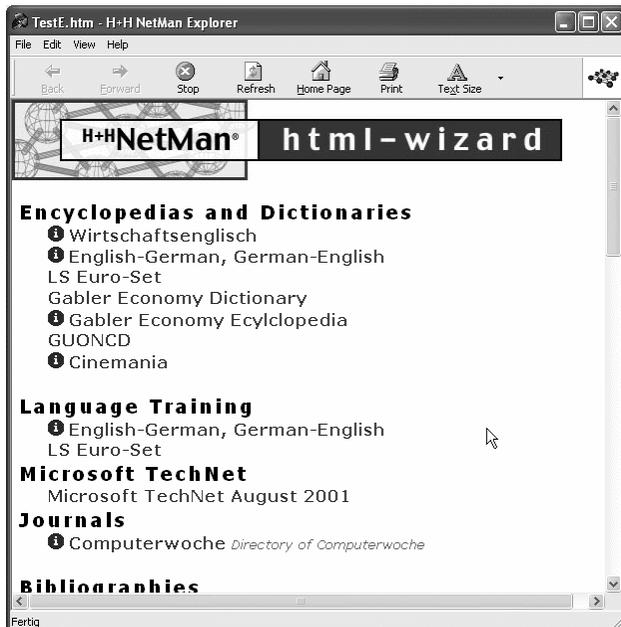
When a user activates the link, an application or a hyperlink is launched.

## Examples

If your desktop structure in the NetMan Client looks like this:

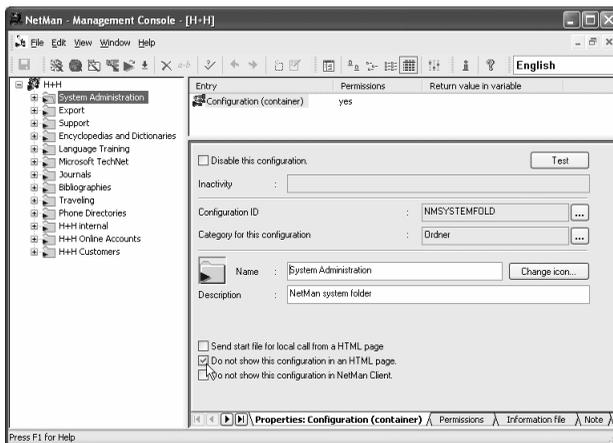


then the HTML output will look like this:



As you can see in this example, not all desktop entries shown in the NetMan Client are

included in the HTML version. Application calls are included in the Web page only if the corresponding option (DISPLAY THIS CONFIGURATION IN AN HTML PAGE) is selected on the PROPERTIES page of the NetMan configuration:



You can also write in an existing file, to insert your NetMan desktop structure into any HTML text. In the example below, the desktop is preceded by a Citrix WebClient download link and a link that starts the NetMan Client:



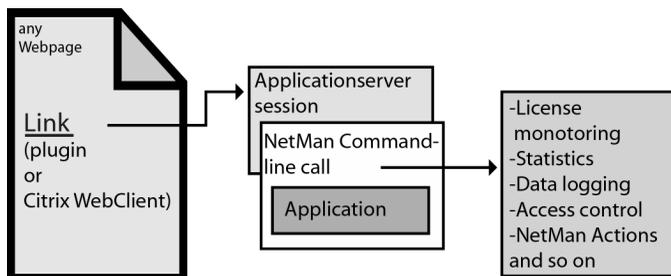
In a terminal server environment, it is generally preferable to start the NetMan Client first, with access to all of its applications; this way only one Citrix license is used even if multiple applications are started. The applications also start more quickly, because a server session does not have to be started every time an application call is launched.

### **Method 2: Inserting NetMan Configurations in Web Pages**

If you want to offer Web pages with individual links to your NetMan application calls, or if the pages you wish to use already exist, you can use the second of the two methods for working with the HTML Wizard. Rather than inserting individual applications in Web pages “by hand”, simply insert a comment with the following format:

```
<!--@nm_configuration = "YourApp"-->
```

and the HTML Wizard automatically creates a link that includes all of the required components.

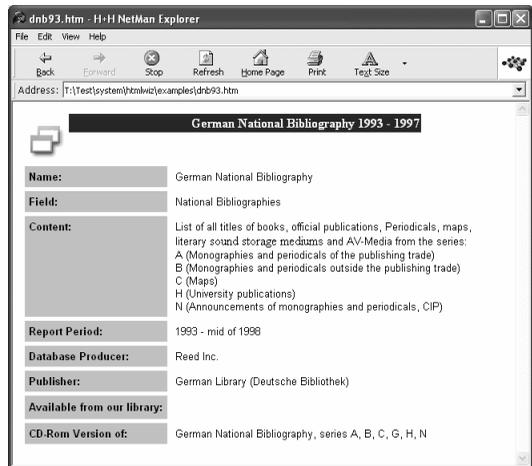


This allows you to group links according to your own preferences, which is ideal for creating customized information systems in the Web. You can group your NetMan configurations with links to other applications; according to subject matter, for example. NetMan is active only in the background, monitoring the application licenses, data logging functions, and access permissions. You can even have these functions run parallel to NetMan operation in your LAN, if desired.

**Example**

In the Web page below, a company had already presented a description on their Web server of the databases they have available. The source code looks like this:

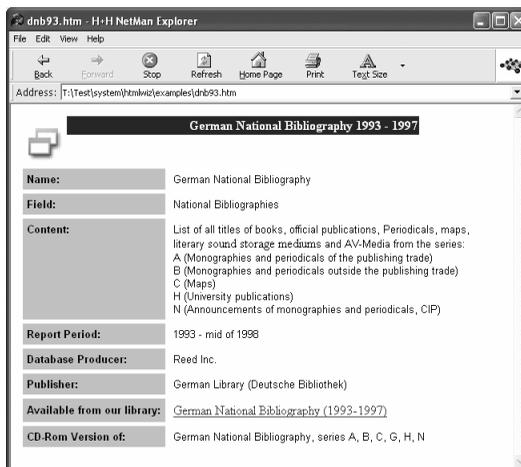
```
<tr>
  <td width="30%" bgcolor="#C0C0C0">
    <font face="Arial" size="2" color="#000000"><strong>
      Available from our library:</strong></font>
    </td>
  <td width="70%" bgcolor="#FFFFFF">
    <font face="Arial" size="2">
    </td>
</tr>
```



We add a comment to the line that contains the text „Available from our library“; the HTML Wizard recognizes this comment and inserts a link to the desired NetMan application call accordingly:

```
<tr>
  <td width="30%" bgcolor="#C0C0C0">
    <font face="Arial" size="2" color="#000000"><strong>
      Available from our library: </strong></font>
    </td>
  <td width="70%" bgcolor="#FFFFFF">
    <font face="Arial" size="2">
      <!--@NM_CONFIGURATION="DNB93"></font>
    </td>
</tr>
```

Once the HTML Wizard has processed the file, the “dnb93” application can be called from this Web page:



The inserted comment caused the HTML Wizard to alter the HTML code as follows:

```
<td width="70%" bgcolor="#FFFFFF">
  <font face="Arial" size="2"><!--@NM_CONFIGURATION="DNB93"-->
  <!--Inserted configuration by NetMan HTML Wizard-->
  <a href="dnb93.ica">German National Bibliography (1993-1997)</a>
  <!--End of inserted configuration = "DNB93"-->
</font>
</td>
```

The HTML Wizard automatically creates the *dnb93.ica* file; it also generates a *dnb93.htm* file, if an ActiveX control, a plug-in or JavaScript is used.

The HTML Wizard also allows you to modify the form of terminal server access after Web pages are completed. This can significantly reduce your workload, especially in view of the continuous development of Citrix Web clients. You can reconfigure this aspect of all your Web pages with a single mouse-click.

## Starting a NetMan Configuration from a Web Page

The Web pages created by your HTML Wizard can function properly only if the browser used to open the page “understands” the links that the page contains. There are basically two different types of links for starting NetMan configurations:

- Links to Citrix ICA files
- Links to NetMan start files

### Using the Citrix Client

If you are using a terminal server with Citrix MetaFrame, you can use ICA files to start server sessions. In this case NetMan runs “invisibly” when the NetMan application call is launched, while all of the functions defined in NetMan are still active, including the data logging and licensing functions as well as the NetMan actions defined in the configuration. The NetMan call generated by the HTML Wizard in this case has the following format:

```
NMcmd32.EXE /id:<configuration ID>
```

(See “NetMan Command Line Call” in chapter 0 for details.)



#### Note

*The NetMan command line program is the only application published for this purpose, as it can be used to launch any NetMan configuration.*

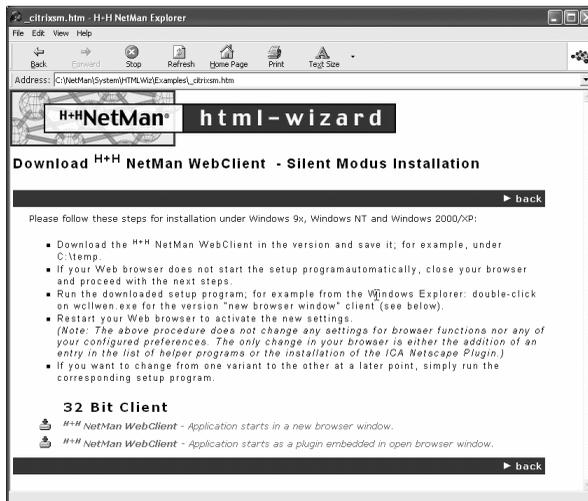
This form of access entails the following prerequisites:

- The ICA WebClient must be installed so the that browser can process ICA files
- The NetMan Terminal Server Module must be installed and registered

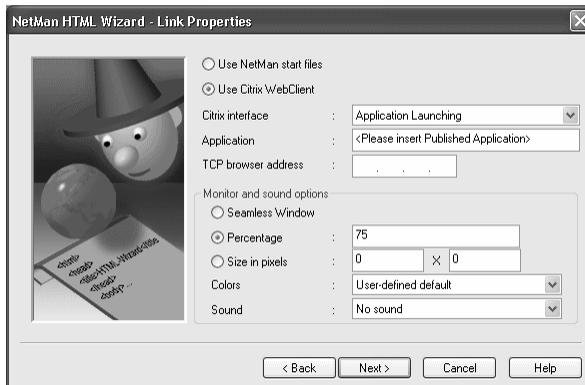


## Tip

Use the NetMan WebClient to install the Citrix Client on workstations. For sample of pages that include this client, see the Examples and %NMHome%\System\HTMLWiz directories. The NetMan WebClient installation avoids known errors in the Citrix Setup and includes among other features, a “silent mode”. You can download the latest version of the NetMan WebClient from the NetMan Web server.



The various options available with the Citrix Client can be configured on the LINK PROPERTIES page:



CITRIX INTERFACES for the browser include:

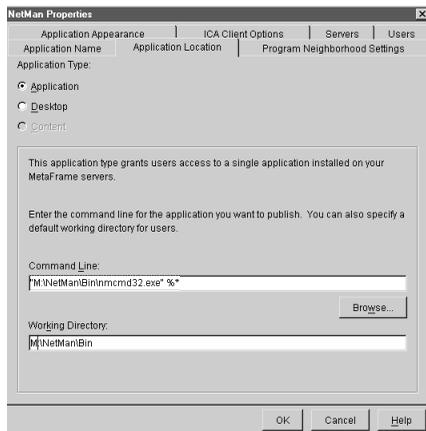
- Application launching
- Embedded ActiveX Control (MS Internet Explorer)
- Embedded Netscape Plug-in
- Browser detection (JavaScript with autostart)
- Browser detection (JavaScript without autostart)
- Embedded Java Applet

Each of these interfaces has advantages and disadvantages. For more information, refer to chapter 3 of the “Terminal Server Module” manual.

The interfaces are located under `%NMHome%\System\HTMLWiz\Launch:`

Citrix interface	Definition file
Application launching	Launch.txt
Embedded ActiveX Control (MS Internet Explorer)	ActiveX.txt
Embedded Netscape plug-in	Plugin.txt
Browser detection (JavaScript with autostart)	JavaAuto.txt
Browser detection (JavaScript without autostart)	JavaMan.txt
Embedded Java Applet	Java.txt

Under START PROGRAM, enter the name of the *published application* defined in the Citrix Management Console to start the NetMan command line program. It is essential that you add a percentage sign followed by an asterisk (“%\*”) at the end of the command line, as a placeholder:





To start the terminal server access control program prior to launching the selected application, for example, enter the following for the command line:

```
"HHASAcc.exe" NMCmd32.exe %*
```

### **Using NetMan Start Files**

You can use the NetMan command line program to start NetMan configurations from a browser interface without a terminal server. In this case, the links created by the HTML Wizard are not links to ICA files, but to *NM files*, also called 'NetMan start files', which are passed to the program as command line arguments. The command line program makes the NetMan working directory its current directory and implements the configuration launch. To register the command line program with its MIME type in the browser, call it once without any additional command line input. Once this has been done, the browser passes NM files to the *NMCmd32.exe* program.

Please note that when you use links to NM files, all the requirements for "normal" NetMan operation in a LAN must be met, including the following two conditions in particular:

- All resources requested by the NetMan configurations must be accessible over the network
- The NetMan Client must be accessible over the network

This may seem obvious, but we point it out here because the use of a browser interface is often associated with an intranet or the Internet, which may seem to imply unlimited access to components that are displayed in the browser. Or it may be associated with the use of a terminal server, for which the only requirement on the local machine is one-time installation of the ICA components.

The advantage of access using NM files does not lie in any simplified configuration of the client PC; the only benefit is the fact that the browser is the interface for all users.

In certain situations, a combination of both options may be useful. For example, a university library might want to use the ICA protocol to control external access to the campus network, while offering Web pages in the campus library PCs that have links to NM files. Users are thus presented with the same interface both on and off campus.

In this case, the HTML Wizard has to generate two copies of each desktop and configuration; one for use with ICA files, and one for NM start files.



#### **Note**

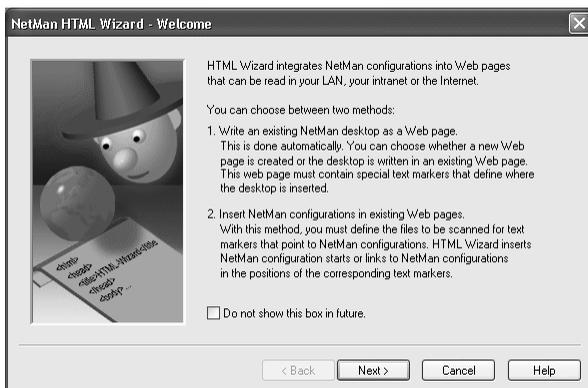
*Unlike the HTML Wizard, NetMan's HTML View creates the HTML documents for users at run time, which lets you configure different responses based on client IP address or host name.*

## Practical Examples

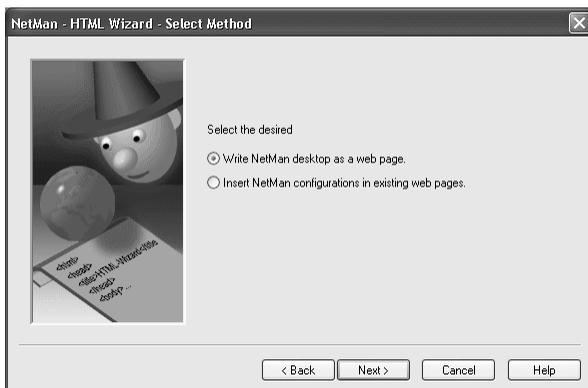
This chapter presents three examples that demonstrate both of the basic methods for using the HTML Wizard. The data used in these examples is included in the initial NetMan installation.

### Converting a NetMan Desktop to HTML

When you start the HTML Wizard, the 'Welcome' window describes the two basic methods briefly. If you no longer need this information, you can select the **DO NOT SHOW THIS BOX IN FUTURE** option.



In the next window, select the option to **WRITE A NETMAN DESKTOP AS A WEB PAGE**.

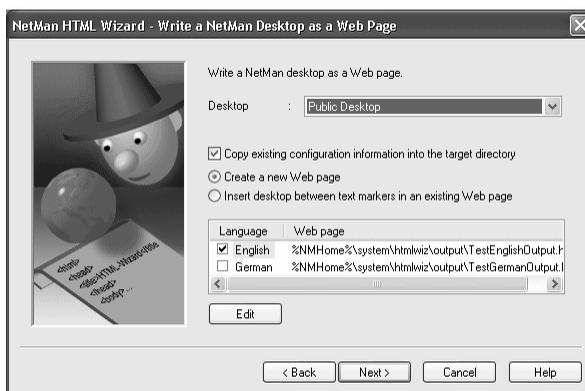


You can now define the following options:

- Which desktop will be converted to HTML

- Whether the information files configured in the NetMan Client are included (in which case the files are copied to the *Info* subdirectory of the HTML output directory)
- Whether the HTML desktop is generated as a new file, or inserted between markers in an existing file
- Which file the data is written in
- If you have the NetMan Language Module, you can also define the language version of the output

For this example, we shall assume you have selected the options as shown here:

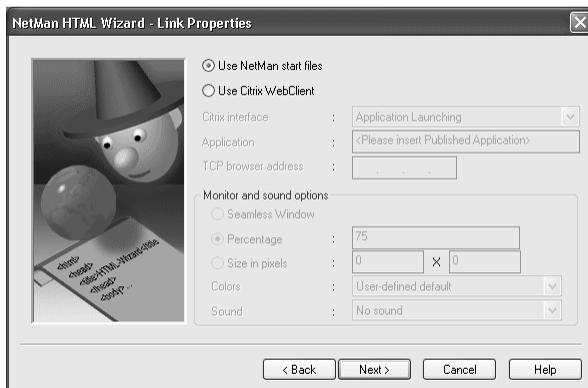


For test purposes, have the files written in the 'Output' directory provided. Once testing has been completed successfully, write the files in a network path on your Web server.

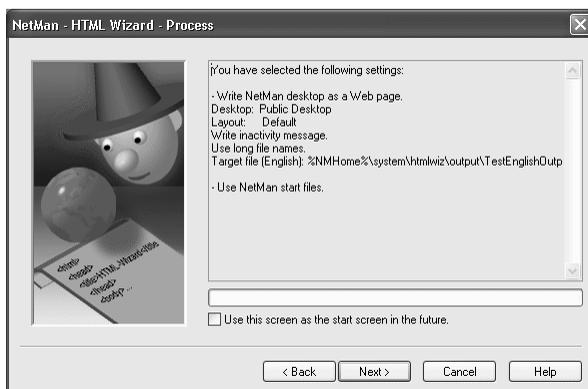
In the next window, select the layout to be used and define whether the message defined for a deactivated NetMan configuration should be output (WRITE INACTIVITY MESSAGE). Whether you select this option or not, links are never included for deactivated configurations.



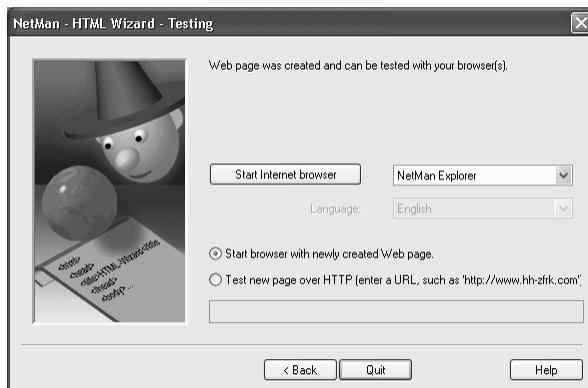
Do not disable the `USE LONG FILE NAMES` option unless your file will be output to a data medium that does not support long names. If this option is deselected, the start file and HTML file for the NetMan configurations are not saved under the configuration ID name, but under an 8.3-compliant file name (a name that is 8 characters long followed by a period and a 3-letter extension).



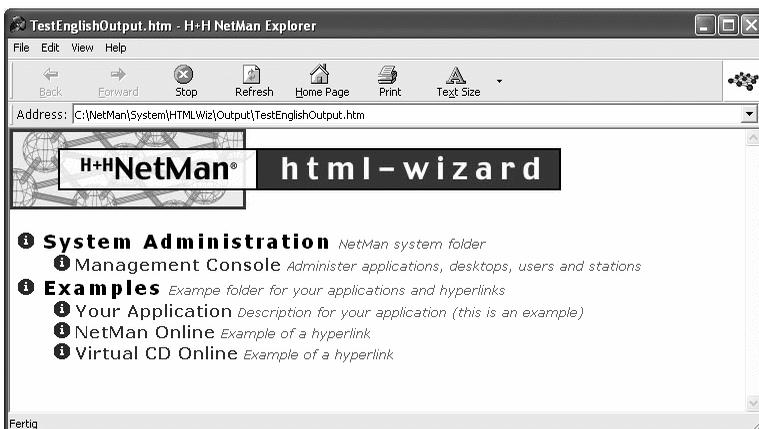
In the next window, you can define whether NetMan configurations are launched by NetMan start files or by an ICA client. We shall assume you have selected `USE NETMAN START FILES`.



The options you have selected are summarized in the next window for you to double-check.



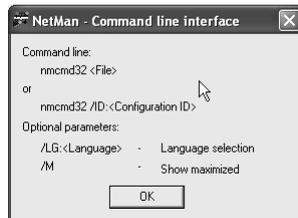
Once the files have been processed, you can test them in a local browser. Since we did not write the files to a Web server, we now select **START BROWSER WITH NEWLY CREATED WEB PAGE**.



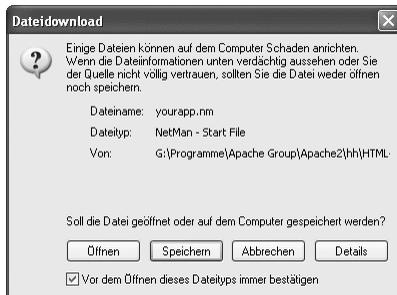
You can now start „Your Application“ by activating the link to ‘YourApp.nm’. This file contains the following:

```
[Config]
ConfigurationID=YOURAPP
```

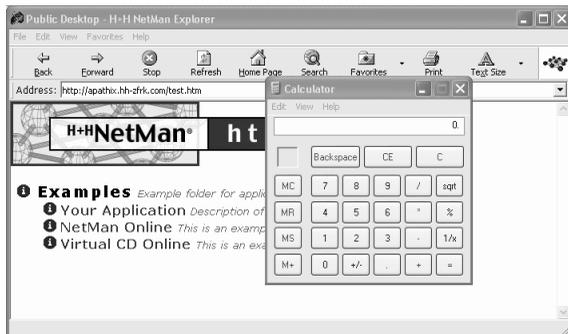
Your browser must be ‘told,’ however, what to do with NetMan start (NM) files; otherwise this file is shown as HTML text. Simply call *NMcmd32.exe* from your workstation once, and the program registers itself in your browser.



From now on, NM files created by the HTML Wizard and received by the browser are passed to the *NMcmd32.exe* program—after you are warned against opening the file. You might want to deselect the option at the bottom of the warning window, so that you are *not* asked every time before a file of this type is opened.



Once you have confirmed the warning message, the application with the configuration ID “YOURAPP” is started:



## Note

The information symbol is created for headings and links if there is an information file specified for the NetMan configuration in question and if you selected COPY EXISTING CONFIGURATION INFORMATION INTO THE TARGET DIRECTORY when configuring the HTML Wizard (see above). When this link is activated, the information file contents are available in the browser.



```

Examples <span ID="descr">
  Sample folder for your applications and hyperlinks</
span>
  </p>
</td>
</tr>
</table>
..
etc.

```

The 'td' formatting (<td width="1%"></td>) that precedes each desktop element indents the left margin for entries in the list according to desktop level. The page is concluded after all desktop entries are written:

```

</body>
</html>

```

If you wish to change the way the page is formatted, change the corresponding template files and generate a new page, rather than modifying individual pages. For example, if you want to present only the configuration name and leave out the description, simply delete the *@NM\_Description* variable from the template.

```

<table border="0" width="100%" cellspacing="0">
<tr>
<td width="@NM_DESKTOPLEVEL%">&nbsp;   </td>
<td><p ID="link">@NM_INFO_LINK<a href="@NM_APP">@NM_PROMPT
</a>
  <span ID="descr">@NM_DESCRIPTION</span>
  </p>
</td>
</tr>
</table>

```

We make this change in the *Link.htm* file stored in the *MyFormat.htf* folder, to leave the templates in the *Default.htf* folder unchanged.



#### Note

When you first install NetMan, the content of the *MyFormat.htf* and *Default.htf* folders is identical.

Then repeat the steps above for generating a page, but select *MyFormat* rather than *Default* for the layout. Now the page looks like this:



Here is an example of the same page in German, using the NetMan Explorer browser:



### *Integrating a NetMan Desktop in a Web Page*

The next example is based on the following preferences:

- We want the application to run on a terminal server over the Citrix Client rather than in the local network
- We want to integrate the NetMan desktop in an existing HTML file, and offer a Citrix Client download and the NetMan Client on the same page

NetMan comes with a sample file, called *example.htm*, that meets these requirements. Copy this file from the `%NMHome%\System\HTMLWiz\Examples` to the root directory of the Web server for this demonstration:



The HTML code in the position where the desktop is inserted looks like this:

...

The following applications are available:

```
</font>
```

```
<br><br>
```

```
<!--@nm_desktop_complete-->
```

```
<!--@nm_desktop_end-->
```

```
<br><br>
```

You can add more information to this page if desired, after the marker that indicates the end of the desktop...

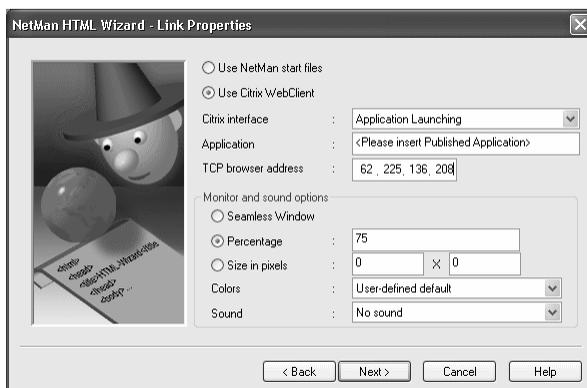
...

The configuration settings in the HTML Wizard differ from those in the first example in the following points:

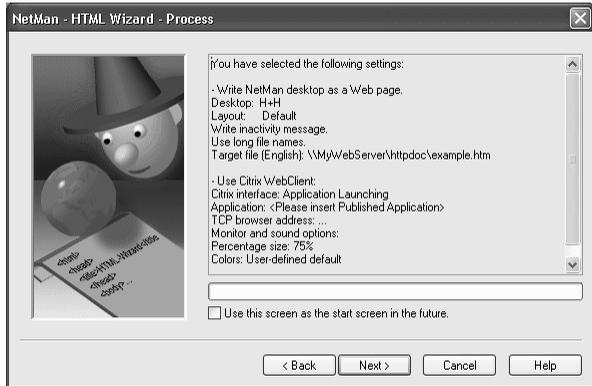
We want to write the desktop in the *Example.htm* file, we do not want to include the information files, and we shall not generate a German version.



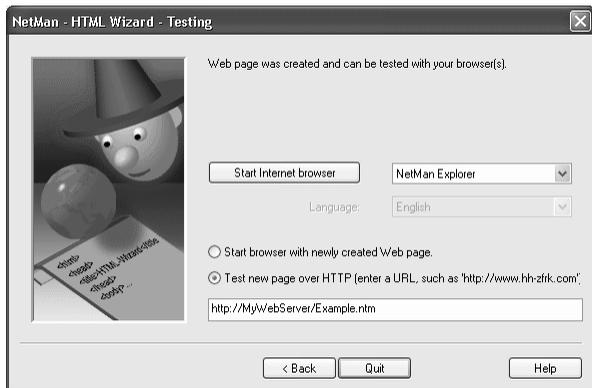
Furthermore, we want the application to be launched by the Citrix Client. To enable this function, we enter the IP address of the MetaFrame server and the name of the published application, which includes the terminal server access control, the NetMan command line program and placeholders for parameters to be loaded from an ICA file:



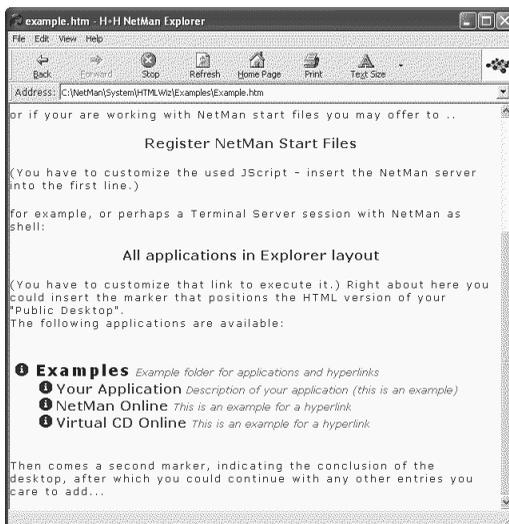
Say we have already made several test runs with these settings and have found the best way to create the page we want; now we can activate the USE THIS SCREEN AS THE START SCREEN IN THE FUTURE. This means we can update the contents of the Web page in future with a single mouse click:



In the next step, the HTML Wizard writes the desktop between the markers. The resulting page is tested over an HTTP connection:

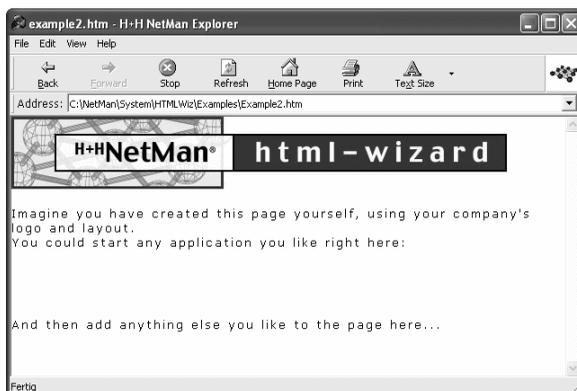


The results are shown in the next illustration. When the link to YOUR APPLICATION is activated, the application is opened within a terminal server session.



## Inserting NetMan Configurations in Web Pages

The following examples demonstrate the second method for working with the HTML Wizard: inserting NetMan configurations in existing HTML pages.



A marker for the desired NetMan configuration has already been inserted. The HTML code for this page is as follows:

```
<html>
```

```
<head><title>NetMan HTML Wizard</title>
<link rel="STYLESHEET" type="text/css" href="_nm.css">
</head>
<body BGcolor="#FFFFFF">
<p></p>
<p ID="text">
Imagine you have created this page yourself, using your
company's logo and layout.
</p>
<p ID="text">
<br>
```

You could start any application you like right here:

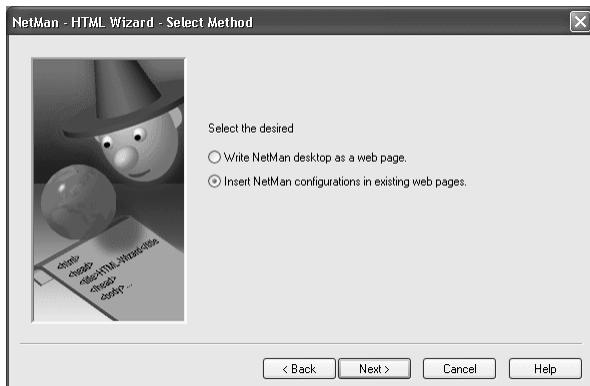
```
</p>
!-@nm_configuration = "YourApp"-
<br>
```

```
<p ID="text">
And then add anything else you like to the page here...
...for example, perhaps a hyperlink to Virtual CD Online
</p>
!-@nm_configuration = "VirtualCD"-
```

```
<p ID="text">
<br>
etc.
```

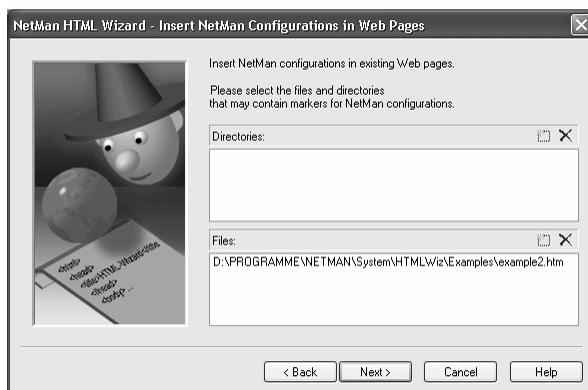
```
</body>
</html>
```

We select the option to INSERT NETMAN CONFIGURATIONS IN EXISTING WEB PAGES:



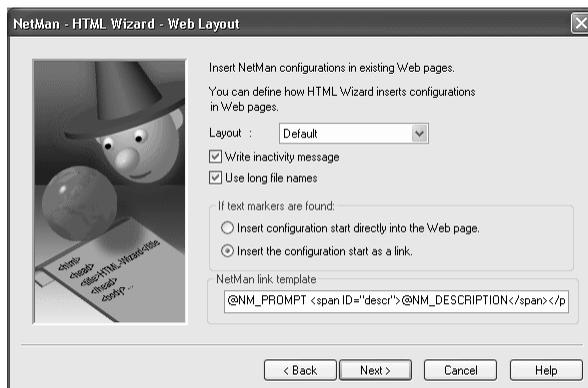


In the next window, we enter the directories and files that the HTML Wizard should scan for markers that point to NetMan configurations:

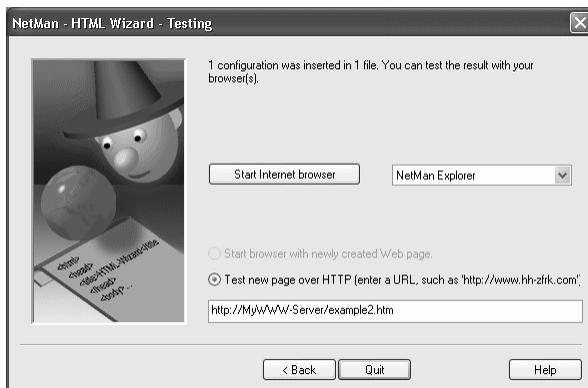


With some Citrix interfaces, you can have the application launch inserted directly into the Web page (known as “direct embedding”). Here we select the option to INSERT THE CONFIGURATION START AS A LINK and accept the default link definition. This has the following two effects: first, the marker is replaced by a link that opens a new Web page (i.e., the application starts in the new page when this link is activated). Second, the format of the link is “Name (Description)” (of the configuration inserted).

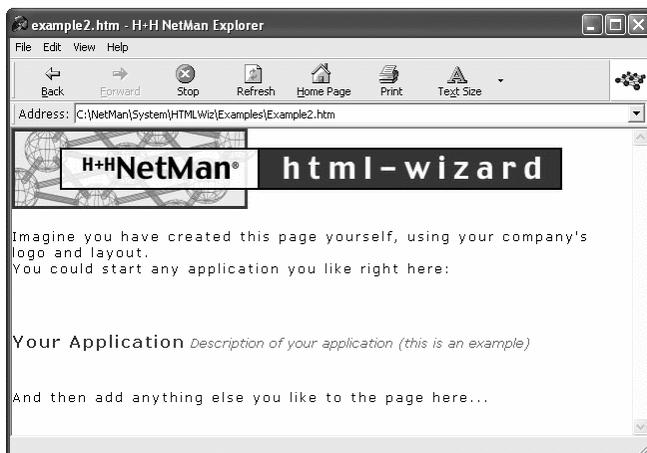
The HTML Wizard ignores the INSERT THE CONFIGURATION START AS A LINK setting if it is incompatible with the selected browser interface; for example, if *Application Launching* or *NM files* is selected.



The last window shows the numbers of files processed and configurations inserted:



The resulting Web page looks like this:



You can modify link formats as follows:

### Example 1

```
@CDM_PROMPT: @CDM_DESCRIPTION
```

Effect: The application description is separated from the name by a colon, and is no longer enclosed in parentheses.

## Example 2

@CDM\_DESCRIPTION

Effect: The text in the link consists of the description, rather than the name, of the NetMan configuration.

Another option is to select INSERT CONFIGURATION START DIRECTLY INTO THE WEB PAGE in combination with BROWSER DETECTION (JAVASCRIPT WITHOUT AUTOSTART) as the Citrix interface. In this case, you need to add the name and description of the application to be launched when you write the Web page.



## HTML Format Templates

### Designing Your Own Layout

The HTML Wizard allows you to select the layout for the Web pages it generates. The term *layout* refers to a number of templates used for HTML formatting of the individual components of a page (title, subtitle, links, page header, and so on). These must be saved in a directory that has the extension *.htf*. The HTML Wizard comes with a *Default* layout, located in the `%NMHome%\system\htmlwiz\default.htf` directory. These templates generate a sample page that has the NetMan logo; you can modify this sample according to your own requirements and preferences.

All subdirectories in the `%NMHome%\system\htmlwiz` directory that have the extension *.htf* are recognized by the HTML Wizard as format templates. To create your own format, modify the templates in the *MyFormat.htf* directory, which is a copy of the *default.htf* directory.

The HTML Wizard looks for format templates that apply to desktop folders, desktop applications, and the basic page format. These are defined in the following template files:

- *BodyInit.htm* contains the basic format of the Web page to be created.
- *Title.htm* contains the header for the entire desktop saved as HTML text.
- *SubTitle.htm* contains the format for a folder.
- *Link.htm* contains the format for an application.
- *Info.htm* contains the format for an information file.

When a desktop object is formatted, the HTML Wizard inserts the name and description defined for the NetMan configuration in question at the positions of the `@NM_PROMPT` and `@NM_DESCRIPTION` markers.

Here is an example for the *Link.htm* template:

```
<table border="0" width="100%" cellspacing="0">
  <tr>
    <td width="@NM_DESKTOPLEVEL%">&nbsp;&nbsp;&nbsp;</td>
    <td><p ID="link">@NM_INFO_LINK<a href="@NM_APP">@NM_PROMPT
  </a>
    <span ID="descr">@NM_DESCRIPTION</span>
  </p>
  </td>
</tr>
</table>
```

Now you can use an HTML editor to modify the format (font size, etc.), or make the changes “by hand” in the HTML code. You can even leave out one of these two markers, if desired. The marker that refers to the NetMan configuration ID (@NM\_APP), however, must not be deleted.



### Tip

If you use an HTML editor, make sure it does not make modifications on its own that render the file invalid as a template; for example, some HTML editors automatically add `</body></html>` at the end of the text.

The markers used in the templates specify (or are replaced by) the following objects:

Marker	Object
@NM_PROMPT	Name of the configuration
@NM_DESCRIPTION	Description of the configuration
@NM_DESKTOPLEVEL	Margin indentation according to folder level
@NM_INFO_LINK	Content of 'Info.htm'
@NM_INFO	Information file
@NM_LAUNCH	Start file (ICA or NM)

The HTML Wizard also scans format directories for files in a subdirectory called *File-Copy*, which are copied to the output directory. These may be, for example, images (*GIF* or *JPG* files) or Citrix Client components.

### Formatting Embedded Browser Objects

Another option is to create a file called *Global.cfg* in a format directory, to define screen coordinates for embedded browser objects and indenting the left margin for desktop entries according to folder level. The following configurations, for example, would correspond to the default formatting (applied when no *Global.cfg* exists):

```
[HTML Wizard]
DesiredHRES =480
DesiredVRES=640
LeftMarginPercent=5
```

If such a file was configured to set *DesiredHRES* to 800 and *DesiredVRES* to 600, then the Netscape plug-in, for example, would be displayed with 800 x 600 pixels when this format template is used. The HTML Wizard replaces the markers in the plug-in definition, contained in `%NMHome%\www\client\launch\plugin.txt`.

```
<EMBED SRC="@NM_LAUNCH" HEIGHT=@NM_DESIREDHRES
WIDTH=@NM_DESIREDVRES START=auto BORDER=on></EMBED>
```

### Formatting the Left Margin Indentation for Desktop Entries

The folders in a desktop structure can be nested up to 9 deep. The default format indents the left margin at each level in 5% increments.

To change this setting, define the desired value in a *Global.cfg* file in your template directory, under *LeftMarginPercent* (see the example above). If you set the indentation to “0”, the margin is not indented for any entries; you could achieve the same effect, however, by removing the corresponding HTML instruction:

```
<td width="@NM_DESKTOPLEVEL" ></td>
```

from the *SubTitle.htm* and *Link.htm* files.

### Using Style Sheets

Because page formatting with the HTML View is distributed over a number of separate HTML text components, it can be difficult to change a format. This is why the default formatting files in *Default.htm* use style sheets.

The following is an excerpt from the *\_nm.css* style sheet, which you can use to change fonts and font sizes.

```
#title {
    font-size: 16pt;
    font-weight: bold;
    letter-spacing: 2px;
    font-style : oblique;
}
```

```
#subtitle {
    font-size: 12pt;
    font-weight: bold;
    letter-spacing: 2px;
}
```

```
#descr{
    font-size : 10pt;
    font-weight: 500;
    letter-spacing: 0px;
    font-style: italic;
    color: #606060;
}
```

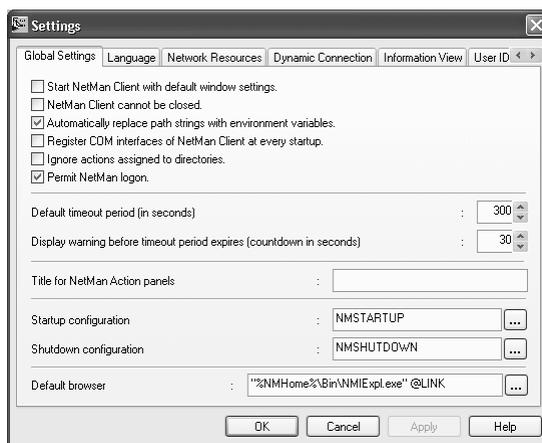
```
#link{
    font-size: 12pt;
    font-weight: bold;
    letter-spacing: 2px;
}
```



## 9. NetMan Explorer

The default browser for Hyperlink configurations started in NetMan is the NetMan Explorer, which both requires the Microsoft Internet Explorer (version 5.0 or later) and replaces it as the user interface. The URL defined in a Hyperlink configuration is passed as a command line argument to the NetMan Explorer, which loads the corresponding page.

This is defined in the NetMan Settings, at the bottom of the “Global” page:

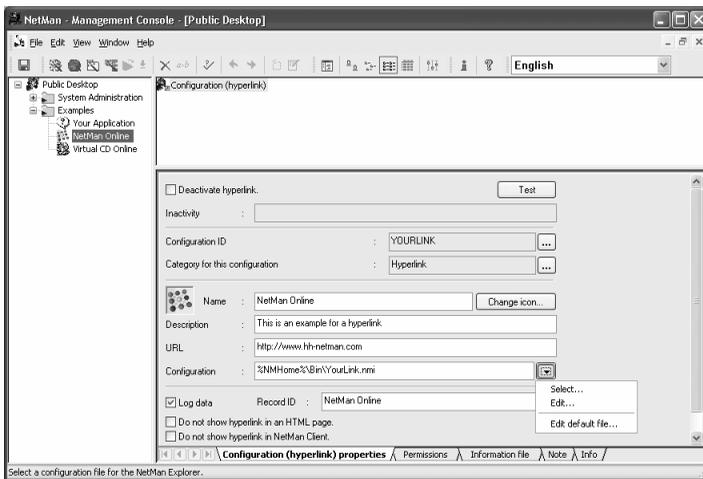


The value in this field is written in the *NMHttpCommandLine* variable. This lets you specify different browsers for different users, stations, or profiles by setting the variable in a Startup configuration.

Leave this field blank to have each workstation use its own default browser to open NetMan links.

With the NetMan Explorer, you can restrict end users' navigation options in a number of ways. In the following section, we will demonstrate some of the possibilities, using the “NetMan Online” and “Virtual CD Online” configurations as examples.

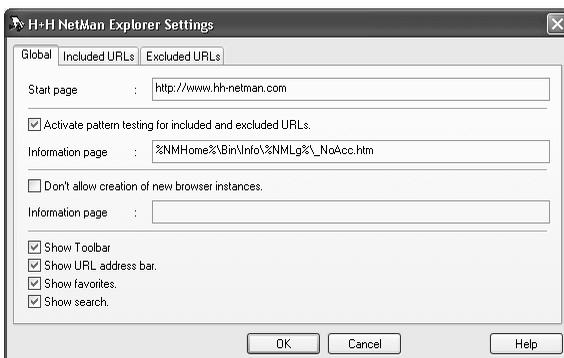
## Examples in the Public Desktop



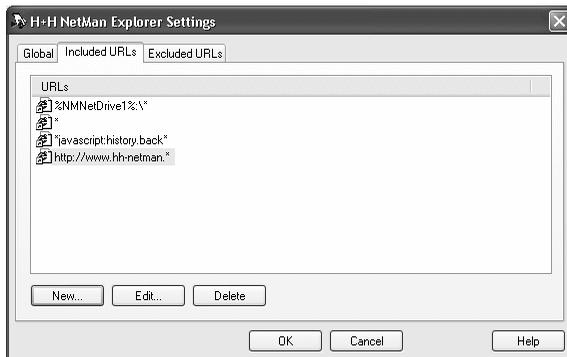
The example configuration called „NetMan Online“ blocks the user from exiting the „WWW.HH-NetMan“ Web page, as well as from using the visiting page. In other words:

- the user sees only the HTML documents that you explicitly permit,
- the user cannot start perform actions that you do not permit (e.g., downloading), and
- the log entry that reflects the launch of the Hyperlink configuration is more precise, in that the usage time it shows refers to the time during which your specified HTML document was open.

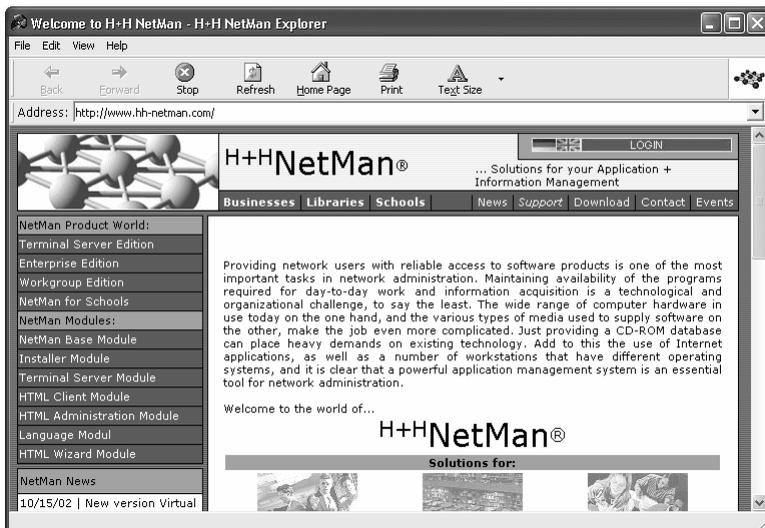
You can edit the `_NMExample.nmi` configuration file in the NetMan Explorer Settings:



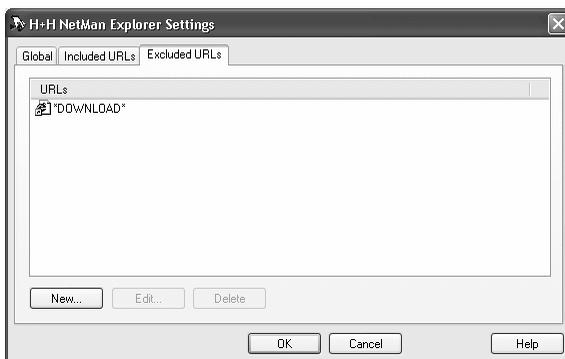
In this example, the operating controls (toolbar, URL address line, Favorites, and Search) are available to the user, as is the option of opening additional browser instances, but the function that restricts access to URLs is active. Permitted URLs are configured as follows:



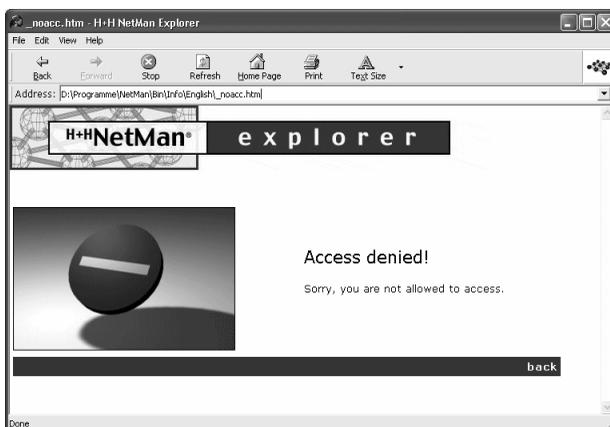
In this case, the wildcard “\*” means the user can also point the browser to “http://WWW.HH-NetMan.com”.



All pages on this server can be accessed, with the exception of the pages specifically excluded:



If the browser is pointed to a URL containing the word “Download,” the NetMan Explorer opens the information page you define for such an event:



To enable the user to return to the page most recently visited, execution of the JavaScript command:

```
javascript:history.back()
```

must be permitted.

There is no specific configuration file configured in the NetMan Explorer for the “Virtual CD Online” sample configuration. In this case, the settings in the NetMan Explorer are applied. These are independent of any settings configured for individual Hyperlink configurations.

## NetMan Explorer Settings

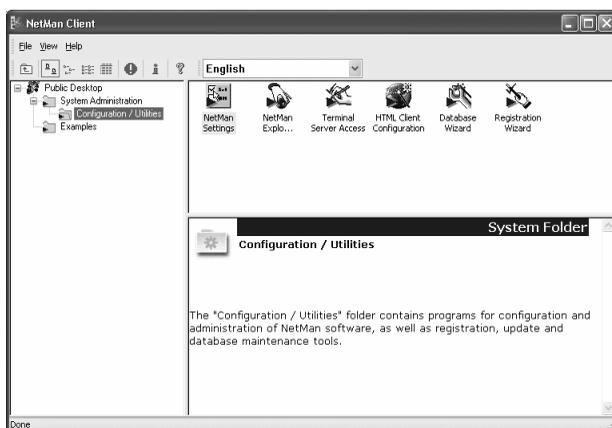
Before you use the NetMan Explorer, you need to adapt its settings for your environment and preferences.

To do this, open the NetMan Explorer Settings program from the CONFIGURATION / UTILITIES folder in the NetMan Client SYSTEM ADMINISTRATION.



### Note

*If you have installed NetMan XP as an upgrade of an earlier NetMan version, you will find the NetMan Explorer settings in the form of a predefined application in the Application Library. For details on integrating the NetMan Explorer Settings from this predefined application, see the example given in “The Application Library”.*



The NetMan Explorer Settings icon looks like this:



There are three dialog pages in the NetMan Explorer Settings:

- GLOBAL: For enabling or disabling the basic NetMan Explorer function.
- INCLUDED URLs: For defining the URLs permitted by the NetMan Explorer.
- EXCLUDED URLs: For defining the URLs to be blocked by the NetMan Explorer.

## Using Wildcards

The following options are available for defining both included and excluded URLs:

- entering specific URLs,
- using wildcards to define URLs, and
- using NetMan variables.

The permitted wildcards are a question mark (“?”) as a place holder for a single letter and an asterisk (“\*”) as a place holder for an unlimited number of letters (or no letters).

### Examples:

- ‘http://www.hh-zfrk.com/’ – Permits only this URL.
- ‘http://www.hh-zfrk.com\*’ – Permits all URLs at this address.
- ‘\*www.hh-\*’ – Permits all URLs that contain these letters (in this order).



### Note

*It is essential that you run the Trace Monitor to check which URLs are actually permitted by your settings, as you might find results other than you expect. For example, when you enter ‘www.hh-zfrk.com’ in a browser’s address line, it is converted in a background process to ‘http://www.hh-zfrk.com/’ (a final forward-slash (“/”) is appended). If you enter ‘www.hh-zfrk.com’ under “Included URLs,” however, none of the other pages at that site are permitted. To permit all pages at that site, you can enter, for example, ‘\*www.hh-zfrk.com\*’.*

Users are permitted to access only those URLs which you define under “Included URLs.” There is no need to specifically exclude URLs unless your definitions under “Included URLs” would otherwise allow access to a URL that you do not wish to permit; for example, when you use wildcards.

### Example:

- Included URL: ‘http://www.hh-zfrk.com\*’
- Excluded URL: ‘http://www.hh-zfrk.com/download\*’

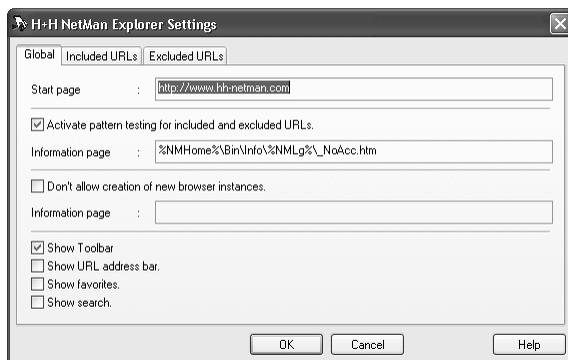
This combination allows access to all pages at the Web site indicated, except pages in the download area.

**Tip**

If you wish to work exclusively with excluded URLs, enter an asterisk (“\*”) as the only entry under “Included URLs.” (This means that all URLs are allowed.) Now you can use the list of excluded URLs to block access to specific pages or Web sites.

**“Global” Dialog Page**

This page lets you define the functions available in the NetMan Explorer. The factory settings are shown here:



**START PAGE:** If no URL is passed to the program from the command line, this setting defines the URL loaded when the NetMan Explorer is started. If this field is left blank, NetMan Explorer opens the page specified as the home page for the Microsoft Internet Explorer.

**ACTIVATE PATTERN TESTING FOR INCLUDED AND EXCLUDED URLs:** Defines whether the NetMan Explorer checks a URL against the URLs entered in the “Included” and “Excluded” lists before opening a Web page.

**Note**

If this setting is disabled, any URL can be loaded.

**INFORMATION PAGE:** If the URL is found in the list of excluded URLs, this page is shown rather than the requested Web page. If this field is left blank, an empty page is shown.

**DON'T ALLOW CREATION OF NEW BROWSER INSTANCES:** Some hyperlinks open a new browser instance rather than loading the linked page in the existing browser window. This option lets you prevent a new browser instance from opening.

**INFORMATION PAGE:** When an attempt to open a new browser instance is blocked by the above setting, the page specified here is opened rather than the linked page.

**SHOW TOOLBAR:** Shows or hides the Toolbar in the NetMan Explorer window.

**SHOW URL ADDRESS BAR:** Shows or hides the URL address bar in the NetMan Explorer window.

**SHOW FAVORITES:** When this option is activated, the NetMan Explorer shows the Favorites configured in the MS Internet Explorer. Otherwise, Favorites are not shown.

**SHOW SEARCH:** When this option is activated, the standard search function can be loaded.



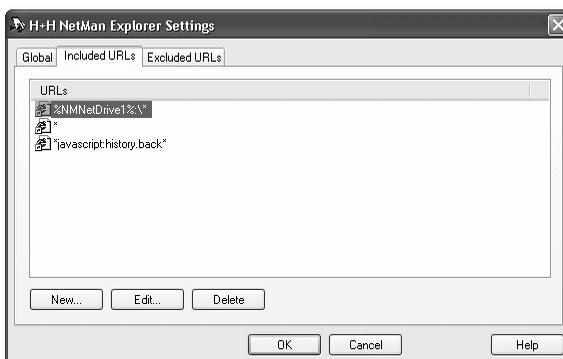
### Note

*If any of the last four options is disabled, the corresponding element is completely hidden in the NetMan Explorer window. For example, if the toolbar is hidden, it cannot be displayed again by selecting the VIEW/TOOLBAR menu item, as the menu item is hidden as well.*

*If you activate "Show Favorites" or "Show Search", make sure the corresponding pages are also included in the list of permitted URLs.*

## "Included URLs" Dialog Page

The PERMITTED URLs dialog page lets you define which URLs can be loaded in the NetMan Explorer.

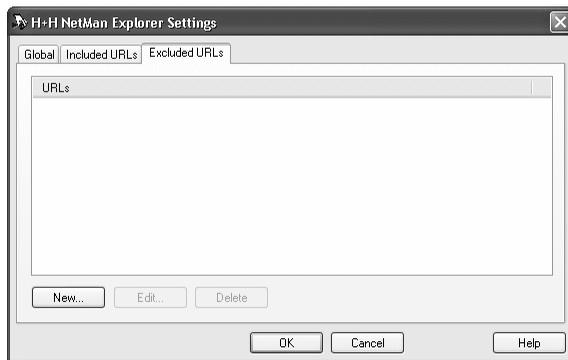


The list shows the permitted URLs. You can edit the list as follows:

- **NEW:** Opens the NEW/EDIT URL dialog, where you can enter a new URL.
- **EDIT:** Opens the NEW/EDIT URL dialog, where you can edit an existing URL.
- **DELETE:** Deletes the selected URL(s) from the list.

### ***“Excluded URLs” Dialog Page***

THE EXCLUDED URLs dialog page lets you define which URLs are blocked from being loaded in the NetMan Explorer.



The list shows the excluded URLs. You can edit the list as follows:

- **NEW:** Opens the NEW/EDIT URL dialog, where you can enter a new URL.
- **EDIT:** Opens the NEW/EDIT URL dialog, where you can edit an existing URL.
- **DELETE:** Deletes the selected URL(s) from the list.



### **Note**

*The list of excluded URLs is mainly used to block URLs that would otherwise be permitted by the settings on the “Included URLs” page. Entries in the “Excluded URLs” list should be limited to subsets of the “Included URLs” list.*

## Passing a URL as a Command Line Argument

You can specify a starting page for the NetMan Explorer by passing a URL on the command line:

```
NMIExpl.exe [<URL>]
```



### Note

*A URL passed on the command line is opened as the start page even if a different page was specified on the GLOBAL page of the NetMan Explorer Settings.*

## Notes on the NetMan Explorer Settings

The NetMan Explorer Settings Program writes its configurations in a file called *NMIExpl.nmi* in the NetMan working directory. These settings are used:

1. when the NetMan Explorer is started without specifying a different configuration file; for example, when it is started from the Windows Explorer, or
2. when a Hyperlink configuration is launched for which no specific configuration file is defined.

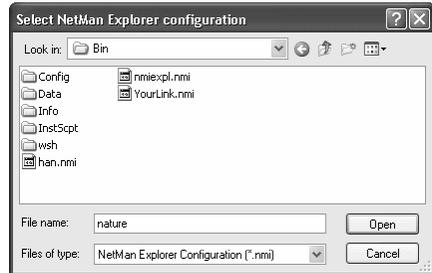
It is easy to configure separate configuration files for individual Hyperlink configurations.



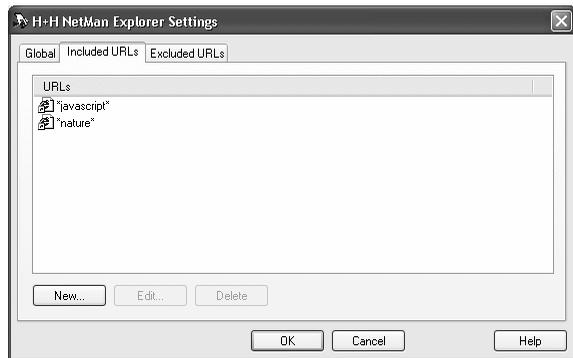
### Tip

*There are two major advantages in defining specific configuration files; on the one hand, it lets you prevent access to pages that are allowed by other configuration files, and on the other hand, it is easier than defining one global configuration file that includes all URLs you may wish to permit or exclude at any given time.*

In the following demonstration, we shall configure a hyperlink to 'www.Nature.com' which does not permit the user to exit the specified Web page. The first step is to create a special configuration file for the NetMan Explorer:



The list of permitted URLs in this configuration file contains the following:

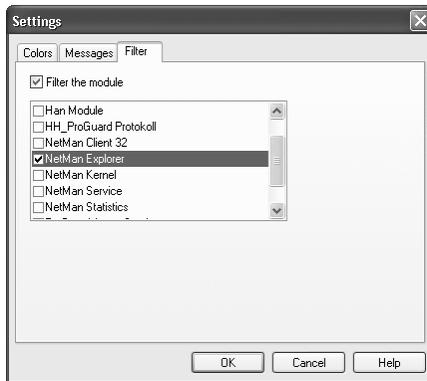


Again, it is essential that the configuration is tested; in this case, we want to check whether any relevant links have been inadvertently excluded. If so, they can be added to the list for inclusion; otherwise, the configuration is finished.

## Testing the NetMan Explorer Configuration

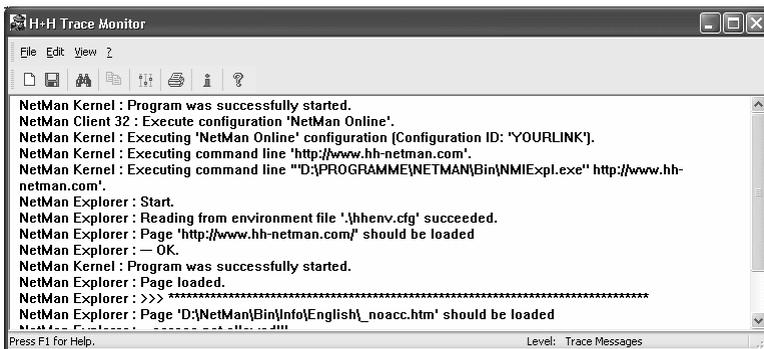
If you find that the NetMan Explorer excludes URLs which you thought you had permitted (or permits pages that you thought were excluded), run the Trace Monitor to follow the process that takes place in your NetMan Explorer.

Start the Trace Monitor and set the output level to “All messages” (by pressing the “plus” key (+) on your numeric keypad, or under VIEW / SETTINGS / MESSAGES) and set the module filter to show output from the NetMan Explorer only.



The Trace Monitor gives you a closer look at the program logic followed by the NetMan Explorer.

In this example, access to “HTTP://WWW.HH-ZFRK.com” was denied—even though you included on your list of permitted URLs—because the Microsoft Internet Explorer added the “/” character to the URL you entered.



### Tip

*It is important to test all of your configurations using the Trace Monitor, because you cannot always tell what URLs need to be loaded, if only temporarily, to access a given page (such as Java scripts, detours to other servers, or “unexpected” URLs).*

## Notes on Using the NetMan Explorer in Protected Environments

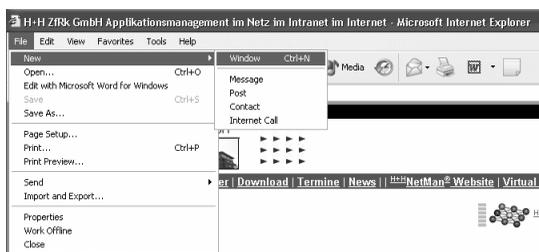
NetMan is often used in protected environments. A “protected environment” is defined here as a workstation on which users are prevented from operating at the “System” level. As a rule, users should be able to access only the information and applications that you provide in NetMan (for example, in terminal server sessions or on public workstations with anonymous users). The NetMan Explorer is designed for use only in this type of environment; otherwise, the default browser (with which the user is familiar) is preferable.



### Note

*Using the NetMan Explorer does not preclude the need for administrators to secure Microsoft Internet Explorer in protected environments!*

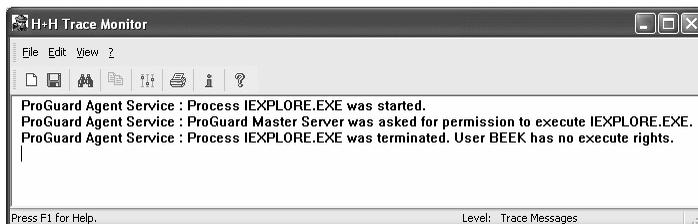
The Microsoft Internet Explorer is almost always accessible, through one method or another, and can be used for gaining access to the entire directory structure on a given workstation. While you can make the NetMan Explorer your default browser (as described in the NetMan knowledge base), there are still a number of other applications that directly or indirectly permit users to call the MS Internet Explorer. Applications which make use of the Internet Explorer internally, as do the NetMan Client and the NetMan Explorer, usually allow the MS IE to be opened by pressing **CTRL-N**, unless this function is explicitly suppressed, as is the case in both of these NetMan programs.



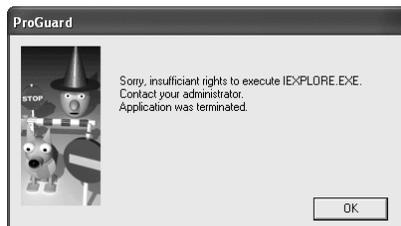
### Tip

*Take the necessary precautions to secure your Microsoft Internet Explorer in protected environments. Depending on your environment, these measures may be implemented through either Group or System policies.*

The NetMan ProGuard Module gives you effective security. The following example should give you an idea of how ProGuard works. The ProGuard Module is described in detail in its own manual. In the first step, we restrict access to the Microsoft Internet Explorer (for a given user group, or for specific workstations). Messages output to the user can be configured by the administrator. As with other NetMan modules, ProGuard processes can be viewed in the Trace Monitor:



With ProGuard, you can block users from starting a given program at any time, but permit the same program when it is started through the NetMan system. With the configurations entered for our „Internet Explorer“ example, ProGuard outputs the following message regardless of the method used in attempting to start the Internet Explorer: whether by double-clicking on a file of a type registered in the Internet Explorer, or through another application program:



## 10. NetMan Helper Programs

The helper programs included with NetMan are described in the following.

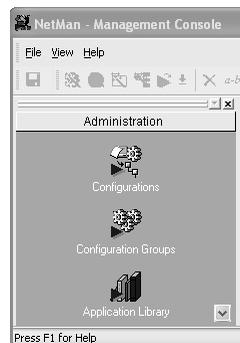
### *The Application Library*

The Application Library is a collection of “ready-made” NetMan configurations, created at H+H and available from our knowledge base or included with updates or service packs. Each *pre-defined application* includes NetMan database entries with its Program action, as well as suggestions for installation and use. Other components can be added as needed, including:

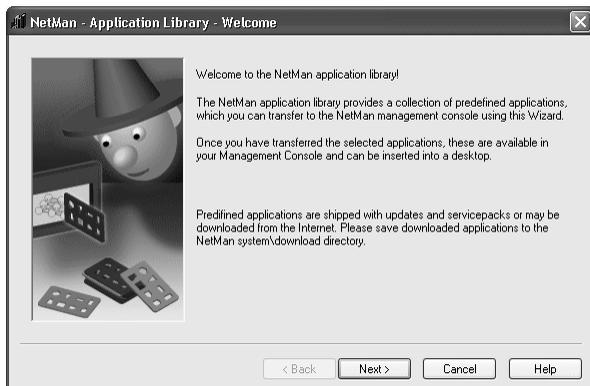
- Actions—these can be inserted before or after the Program action, as may be required or useful, such as conditions, scripts, copy or delete actions, or actions that modify INI files or the Windows Registry.
- HTML-based information files
- Other files of various types

Procedures for obtaining pre-defined applications and integrating them in your NetMan system are described below.

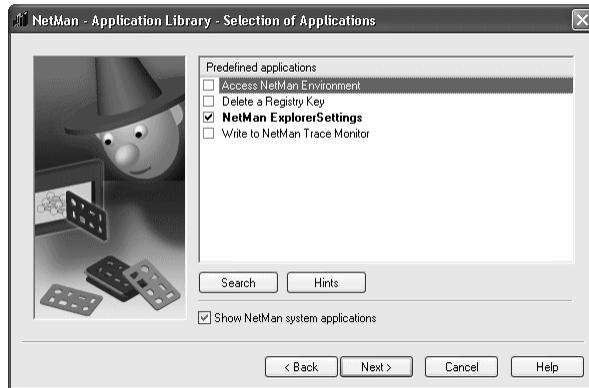
Activate the Application Library from the selection bar in your Management Console:



Click once to open Library.

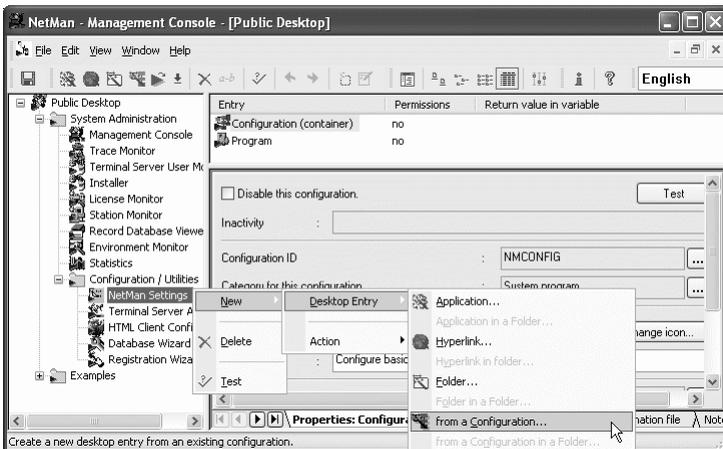


Click on NEXT and enable the SHOW NETMAN SYSTEM APPLICATIONS option on the next dialog page.

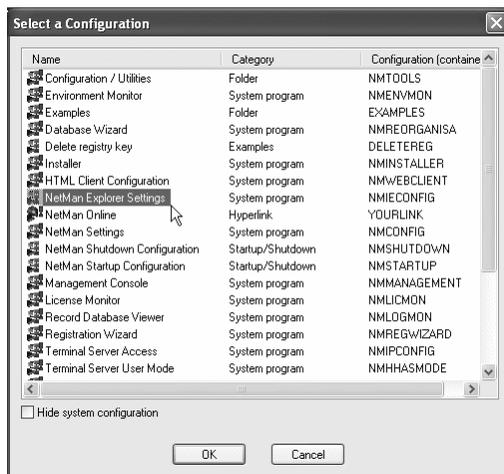


The pre-defined configuration offered here installs the NetMan Explorer Settings program in your system. If you have updated an earlier NetMan version to NetMan XP, the NetMan Explorer Settings program is not located in your system administration folder and must be installed as described here. Click on NEXT to add this configuration (configuration ID: NMIECONFIG) to your NetMan databases.

You can now integrate this configuration in your system administration as follows: First, create a desktop entry based on the new configuration.



The NMIECONFIG configuration is now integrated as a desktop entry, as a Container or Hyperlink configuration:



Use the same mechanism to integrate configurations downloaded from the Internet. The Application Library automatically looks for the compressed configurations (APS files) in the `%NMHome%\System\Download` directory.



## Helper Programs for the 'Execute' Action

The smaller helper programs described below expand your options for the use of NetMan configurations.



### Note

*In most of the examples presented in the following, the `HOLD SUBSEQUENT ACTION(S) UNTIL THIS PROGRAM IS CLOSED` property of the Execute action must be enabled so that the overall configuration can be processed successfully.*

All helper programs are stored in NetMan's working directory.

The helper programs include:

- HHCopy.exe
- HHMkDir.exe
- HHDelete.exe
- HHCmd.exe
- HHDummy.exe
- HHSetAtr.exe
- NMNCon32.exe
- NMLogin.exe

With the exception of HHDummy.exe, all of these helper programs expect command line arguments. To view a list of the arguments valid for a given program, call the program either with no arguments or with «/?» after the program call.

The dialogs generated by the helper programs can be suppressed by entering /q (for “quiet”) when the program is called.

These programs generally output messages to the Trace Monitor only when SHOW ALL MESSAGES is enabled.

Each of the sections below describes a helper program and its arguments, and includes an example illustrating the particular function of the program.



#### Note

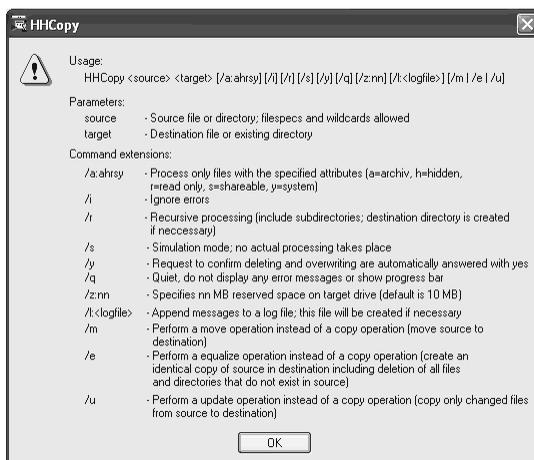
*All of these programs can be used by NetMan customers outside the NetMan directory structure. Some of these helper programs, however, require additional NetMan DLLs in order to run.*

### ***HHCopy – Copying Files and Directories***

HHCopy lets you customize copy routines as follows:

- These actions can be designed by—or completely hidden from—your end users.
- Use of these actions can be recorded in log files.
- These actions can be expanded to include subdirectories, and made dependent on file attributes.
- The “Update” and “Equalize” actions provide powerful tools for expanded functionality.

The following panel is opened when you enter `HHCOPY .exe` or `HHCOPY.exe /?` on the command line:



This gives you the following options:

- In the simplest case, this program copies a file:  
`HHCOPY.exe c:\myfiles\myfile.txt c:\temp`  
 -> A copy of "myfile.txt" is created in c:\temp.
- You can also use wildcards:  
`HHCOPY.exe c:\myfiles\*.txt c:\temp`  
 -> All files with the ".txt" extension are copied to c:\temp.
- With the "/m" option, files or directories are moved rather than copied:  
`HHCOPY.exe c:\myfiles\*.txt c:\temp /m`  
 -> All files with the ".txt" extension are moved to c:\temp.
- You can also have files/directories copied "as needed":  
`HHCOPY.exe c:\myfiles\*.txt c:\temp /u`  
 -> Files with the ".txt" extension that do not already exist in c:\temp, or for which a different version exists in c:\temp, are copied to that directory. The attributes of the file in the target directory are changed as needed to match those of the corresponding file in the source directory.
- You can compare entire directories as well:  
`HHCOPY.exe c:\myfiles c:\temp /e`  
 The content of the target directory, c:\temp, is made to match that of the source directory, c:\myfiles. Files with identical names are overwritten only if

the file in the source directory has a different size or date than that of the corresponding file in the target directory. Differences in file attributes are adapted to match the files in the source directory. Files that do not exist in the source directory are deleted from the target directory.

The following options are available for the “Copy,” “Move,” “Update” and “Equalize” operations:

<code>/q</code>	Suppress standard screen output
<code>/y</code>	Answer all prompts to confirm overwriting or deleting with “yes”
<code>/i</code>	Ignore all errors
<code>/s</code>	Simulation mode: Operations are not actually performed. Use this function in combination with log file data to test operations.
<code>/r</code>	Perform the operation for all subdirectories of the target directory as well
<code>/a:&lt;attribute(s)&gt;</code>	Process only files that have the specified attribute(s)
<code>/l:&lt;log file&gt;</code>	Record operations in the specified log file. This log file is created automatically if it does not already exist.



#### Note

*The HHCOPY helper program assumes that the target directory exists. The target directory is created automatically only when you call the equalize operation. For all other operations, create the target directory (if needed) using the HHMkDir command before starting the Copy operation. Exercise caution when performing an Equalize operation with the “/r” option, as incorrect usage can cause significant damage. For example, if your source directory is empty, all files and subdirectories in the target directory are deleted.*



#### Tip

*Use the “/s” option to test operations before performing them.*

**Example**

Say you have an application that could be made non-functioning through certain user input. At the same time, you cannot prohibit users from saving input, as 'write' permission in the application data is required for use of the application. If you have a reference installation of this application, you can insert an Update action to provide a functioning installation for users:

```
HHCopy.exe \\<Server>\NMSysProg\Normdat.cdw
\\%NMAppUNC%\Normdat.cdw /r /u /q /i /y
```

The original installation is located in the \\<Server>\NMSysProg share. Since the Update function overwrites only changed files in the target directory, the operation is completed so quickly that the user does not even notice that saving data actually triggers an application update.

**HHMKDir – Creating Directories**

The following panel is opened when you enter HHMKDir.exe or HHMKDir.exe /? on the command line:



Calling the HHMKDir.exe program is equivalent to entering the "MD" command. The only difference between the two following examples is that the first opens a command line window:

```
MD %NMHome%\User\Tmp\%ComputerName% > NUL
```

```
HHMKDir %NMHome%\User\Tmp\%ComputerName% /q
```

The /q suppresses standard screen output.

**Example**

Say a given application requires a path for temporary files. When the program is run on a terminal server by multiple users in parallel, you need a separate "temp" path for each user. You can implement this by inserting two 'execute' actions. The first creates a workstation-specific directory, and the second uses a 'Subst' command to map it as a drive. The command lines for these two 'execute' actions are as follows:

```
HHMKdir.exe %NMAppDrive%\Application\TMP\%ClientName% /q
```



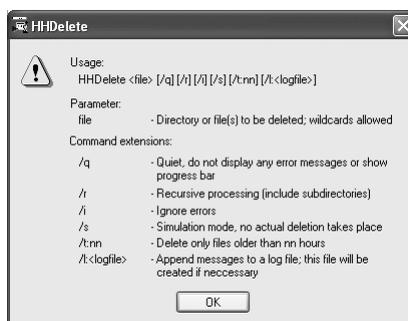
```
HHCmd.exe Subst Y: %NMHome%\User\TMP\%ClientName%
```

HHCmd.exe is another NetMan helper program and executes the ‘Subst’ command in the background (see below for details).

### HHDelete – Deleting Files and Directories

HHDelete gives you the functions of the “DEL” and “RD” commands in a Windows program, the execution of which can be logged if desired. The HHDelete program makes it easy to delete subdirectories with names that are unknown and cannot be logically deduced. The /t option lets you delete files of a specified minimum age.

The following panel is opened when you enter HHDelete.exe or HHDelete.exe /? on the command line:



The following options are available:

<b>/q</b>	Suppress standard screen output
<b>/r</b>	Perform the operation for all subdirectories of the target directory as well; emptied directories are deleted
<b>/i</b>	Ignore all errors
<b>/s</b>	Simulation mode: Operations are not actually performed. Use this function in combination with log file data to test operations.
<b>/t:nn</b>	Directories and files are deleted only if they are at least ‘nn’ hours old; for directories, the age is calculated based on the latest file in the directory.
<b>/l:&lt;log file&gt;</b>	Record operations in the specified log file. This log file is created automatically if it does not already exist.

### Example

Say a given application creates a large number of temporary subdirectories, named %NMAppDrive%\ZDB\OS-xxxxx.TMP. Generation of that portion of the directory name represented here by “xxxxx” is either random or time-dependent. With extensive use,

the application creates multiple megabytes of “waste data.” Because you do not know the names of all these subdirectories, nor whether a particular subdirectory is in use at any given time, you can use HHDelete, with a wildcard, to delete data at specific intervals (for example, every 6 hours) to help minimize the accumulation of waste data. To do this, insert the following ‘execute’ action after the Program action in your NetMan configuration:

```
HHDelete %NMAAppDrive%\ZDB\OS-*.tmp /q /t:6 /r /i
```

### ***HHCmd.exe – Hiding Command Execution***

The following panel is opened when you enter HHCmd.exe or HHCmd.exe /? on the command line:



HHCmd.exe lets you execute commands entered on the command line, batch files and scripts in hidden processes. The default command processor used is configured in *HHCmd.cfg* as Cmd.exe and can be changed as needed.

### ***Examples***

One example for the use of HHCmd.exe is given above, in the section describing HHMkDir.exe (hidden execution of the ‘Subst’ command).

For another example, we will return to the problem mentioned in the description of HHCOPY.exe, in which an application directory is restored from a protected reference installation before the application is launched (using the Update function). Simply restoring the entire directory is convenient when you do not know exactly which components need to be restored, and the process required for finding out would be too time consuming. If you do know which components might need to be restored, however, you can perform the “Update” more efficiently. In this case, the critical data is deleted, or restored from another directory. The path to the “Dataware” application is passed to the script as a command line argument, so that the script is applicable for all “Dataware” applications. A preceding Execute action calls the script for hidden processing, with the working directory for the “Dataware” application as a parameter:

```
HHCmd.exe %NMAAppDrive%\DWW_Init.cmd Normdat.cdw
```

The script is as follows:

```
: first Parameter: Directory of Dataware application
copy \\<Server>\NMSysprog\%1\_dw_.cfg \%1\_dw_.cfg
del \%1\dww.ini
```

## HHDDummy.exe – “Do Nothing” or “Wait”

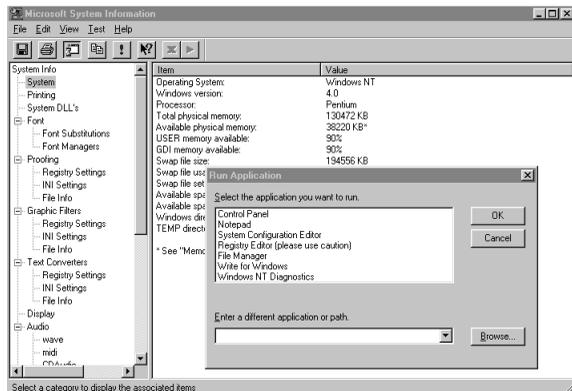
The following panel is opened when you enter HHDDummy.exe /? on the command line:



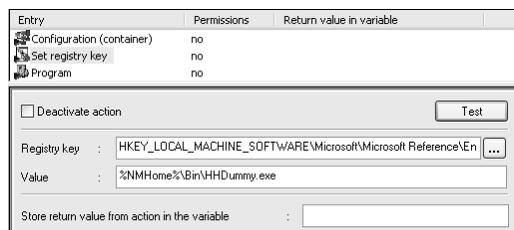
/t:nn HHDDummy.exe closes after ‘nn’ seconds.

### Example

Microsoft Encarta permits users to call MSInfo.exe, which provides extensive information on the workstation. If the user selects „System Info,“ they suddenly have access to functions that should not be permitted in a protected environment:



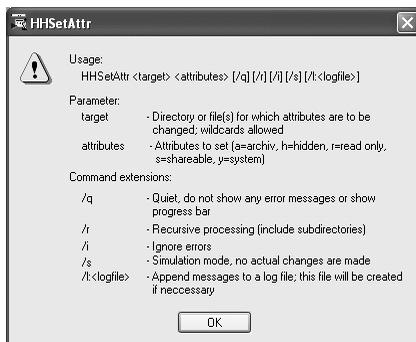
You could prevent this access by deleting the corresponding Registry entries, but this would result in the appearance of an annoying error message any time the button in question is activated. A better alternative is to insert „HHDDummy.exe“ as a ‘pseudo-entry’ in the Registry; now, no error message is shown and no function is activated.



## HHSetAttr.exe – Setting File Attributes

HHSetAttr.exe is a Windows program that lets you set file attributes, similar to command line program attrib.exe. Additionally, it lets you set the NetWare file attribute “Shareable” and record the operations executed.

The following panel is opened when you enter HHSetAttr.exe or HHSetAttr.exe /? on the command line:



The following options are available:

- /q** Suppress standard screen output
- /r** Perform the operation for all subdirectories of the target directory as well.
- /i** Ignore all errors
- /s** Simulation mode: Operations are not actually performed. Use this function in combination with log file data to test operations.
- /l:<log file>** Record operations in the specified log file. This log file is created automatically if it does not already exist.

### Examples

In the first example, all file attributes are cleared from all files in the NetMan application drive so that you can tell from the 'Archive' attribute which files are updated by end users through the use of applications.

```
HHSetAttr.exe %NMAAppDrive%\*.* /q /r /i
```

In the second example, all executable files are set to 'read only' mode to protect them from user manipulation, whether intentional or inadvertent.

```
HHSetAttr.exe %NMAAppDrive%\*.exe r /q /r /i
```

## NMLogin.exe – Executing a Server Login

The following panel is opened when you enter NMLogin.exe or NMLogin.exe /? on the command line:



/d      Execute logoff



### Note

*The “Network login” action is preferable for use in NetMan configurations, as it is easier to configure and can be tracked in the Trace Monitor. NMLogin.exe, on the other hand, can also be used outside the NetMan system; for example, in a script. The NMLogin.exe program does not require any other NetMan DLLs, and can be copied to other directories.*

### Example

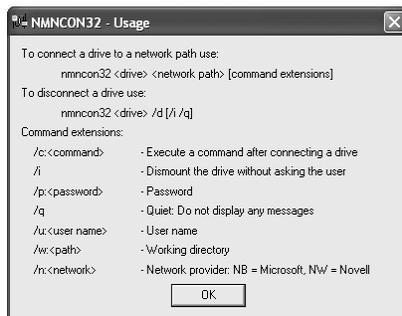
Say you want to execute a login at the beginning of a MetaFrame session for anonymous users who have no rights elsewhere in the network. You can use the NMLogin program outside the NetMan system for this purpose; for example, if NetMan is not installed on the server in question. In our example, we assume you have already set up a “Gateway user” called NMAnon for this purpose, with the password “\*\*\*\*\*”. Now all you need to do is have the following command processed in the login script for anonymous users:

```
NMLogin.exe Server nmanon xxxxxxxxx
```

The anonymous user in question can now access resources on the server within the scope provided by the NMAnon account.

## NMCon32.exe – Connecting a Drive

The following panel is opened when you enter NMCon32.exe or NMCon32.exe /? on the command line:



The following command maps the “NetMan” resource from the “Server” server to the “P:” drive:

```
NMCon32.exe P: \\Server\Netman
```

The following options are available:

<b>/c:&lt;command&gt;</b>	Execute the program after mapping the drive
<b>/i</b>	Ignore all errors
<b>/p:&lt;password&gt;</b>	Enter the user password
<b>/q</b>	Suppress standard screen output
<b>/u:&lt;user&gt;</b>	User account under which the drive is connected
<b>/w:&lt;path&gt;</b>	Working directory for executing the program
<b>/n:&lt;network&gt;</b>	Network provider; this option speeds up execution, as the program would otherwise have to find the provider

You can implement drive mapping within NetMan configurations by configuring the corresponding setting in the Program action or by adding a *Connect Drive* action (e.g., in a startup configuration). NMCon32.exe, on the other hand, can be used outside the NetMan system; for example, by adding it to a login script. The NMCon32.exe program does not require any other NetMan DLLs, and can be copied to other directories.

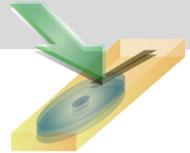
### *Example*

Say you want to map a volume or share on a different server to a local drive designation for anonymous users who have no rights elsewhere in the network at the beginning of a MetaFrame session.

```
NMNCon32.exe P: \\Server\Netman /q /i /u:nmanon /p:xxx /c:  
p:\bin \nm32.exe /w: p:\bin /n:nb
```

The options entered here have the effect that the drive is mapped under the NMAnon account, with password “xxx” with no input dialogs opened and no error messages output. Furthermore, once “P:” is mapped successfully, the “P:\Bin\NM32.exe” is started in the “P:\Bin” directory.

## H+H NetMan® XP Installer Module





# Table of Contents

---

1. Introduction .....	1
Contents of This Manual .....	1
Performance Features .....	1
Requirements .....	2
Recommended Reading on the Windows Registry .....	2
Installation .....	3
Basics .....	3
Using an Installer in a Network – the H+H Installer Services .....	4
Using an Installer in a Terminal Server Environment .....	6
How an Installer Works .....	7
2. NetMan Installer in Three Easy Steps .....	9
SnapShot Definitions .....	9
Selecting Files .....	10
Monitoring the Registry .....	12
Results of Comparison .....	13
Script Files .....	14
3. From SnapShot to Script: 3 Sessions with Bob .....	17
SnapShot of the Workstation .....	17
Comparing System States After Installation .....	19
Generating a Script from Results .....	20
4. Installation with the NetMan Installer – a Demonstration .....	23
The SnapShot .....	23
The Application Setup .....	24
From Comparison to Results .....	25
The Script .....	27
Editing the Script .....	28
5. Integrating Scripts in the NetMan Client .....	31



# 1. Introduction

## *Contents of This Manual*

- Chapter 1, “**Introduction**”, gives you an overview of the performance features of the NetMan Installer module. It also describes installation programs in general and tells you how the NetMan Installer works, with tips on how best to use it in LANs and in terminal server environments.
- Chapter 2, “**NetMan Installer in Three Easy Steps**”, describes the basics of the NetMan Installer.
- Chapter 3, “**From SnapShot to Script: 3 Sessions with Bob**”, describes in detail how to monitor installation routines and use the resulting file to create a script you can use to distribute application components over a network.
- Chapter 4, “**Installation with the NetMan Installer – a Demonstration**”, presents an example using all of the techniques learned in the previous chapters.
- Chapter 5, “**Integrating Scripts in the NetMan Client**”, tells you how to use NetMan Installer scripts in your NetMan application calls.

## *Performance Features*

As a network administrator, or as the one responsible in your network for the deployment of applications, you are frequently confronted with the challenge of making new and hitherto unknown applications available to your network users. The hurdles you are faced with in accomplishing these tasks can be high or very high, depending on the applications you install. An *Installer* is the ideal aid in clearing these hurdles. The emphasis here is on “aid”, because the program has not been written that can automatically do the entire job for you, from start to finish.

NetMan Installer is the ideal tool for installing network-capable applications. The greatest advantage of the NetMan Installer over other installers is that it fits seamlessly into the overall NetMan concept. It simplifies the integration new applications into your NetMan system, thus reducing your workload considerably.

## Requirements

To get the most out of your Installer, you need to have a basic understanding of the following:

- The directory structure of the Windows operating system version you use
- The structure and function of the Windows Registry
- The significance of subkeys in the Registry



### Note

*If the Installer is not used correctly, your PC may not operate correctly when you use an installation packet created by the NetMan Installer.*

A working knowledge of the software you use, combined with the experience gained from your first few installation projects, are all you need to get optimal results with the NetMan Installer.

## Recommended Reading on the Windows Registry

*Born, Günter: Inside the Microsoft Windows 98 Registry.*  
August 1998. 494 pages + 1 CD-ROM. Microsoft Press, \$39.99.  
(ISBN 1-57231-824-4)

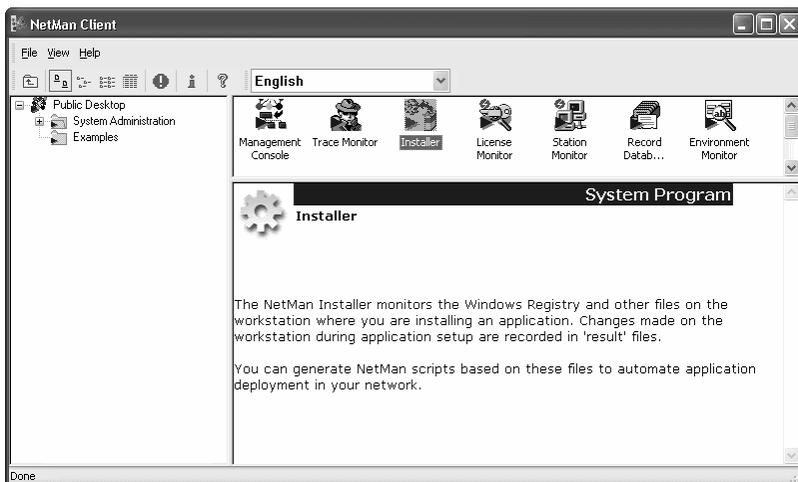
*Frank R. Walther: Registry Guide. Windows 2000, Windows NT4.*  
2001 496 pages, Markt+Technik, •59.95  
(ISBN: 3-82726-042-6)

*John Woram: Windows 98 Registry.*  
1999. 464 pages, MIS Press, \$24.99  
(ISBN: 1-55828-591-1)

## Installation

The procedure for installing the NetMan Installer module is described in detail in Chapter 2 of the “Base Module” manual, under “Subsequent Installation/Registering Additional Modules”. Remember that the licensing for this module must be activated when you register your NetMan software.

Once the module has been installed and registered, you can call the NetMan Installer from the SYSTEM ADMINISTRATION folder in the NetMan Client:



The Installer icon looks like this:



## Basics

The task of an Installer is to keep track of all the modifications and additions made on a given workstation during application setup. This can serve one or both of the following purposes:

- The record of modifications provided by the Installer can help network administrators determine whether and how applications influence one another.
- The Installer can put together an “installation package” that executes the required workstation modifications, for transparent installation of the application on other workstations. A great advantage here is the ability to edit this installation package, which lets you optimize the installation process still further.

Not only the setup program itself, but also internal settings in a newly installed program can modify configuration files or the Registry. You might find that you want to make some of these new application settings the defaults for your users, for example by distributing them in a NetMan Script action. It is not always evident at first glance, however, where the new settings are stored, especially if the application in question makes a lot of changes in INI files or the Registry. This is where NetMan Installer functions come in handy, keeping track of all modifications and writing them in an editable script. For some practical examples of Installer scripts, just point your browser to our NetMan Knowledge Base (<http://www.hh-netman.de/Misc/knowledgebase.cfm?lg=0>).

### ***Using an Installer in a Network – the H+H Installer Services***

One of the routine tasks in network administration is providing user applications in the network. As the number of workstations in a network increases, so does the amount of work involved in performing the application setup on each individual station. An Installer program simplifies these tasks, reduces the administrative and helps ensure reliable deployment of applications.

When you have NT-based clients in a LAN, you need to be able to install software on workstations without using an administrative account. In the NetMan system, this is taken care of by the two services, *InstallerMaster* and *InstallerAgent*. The latter is installed as a service on client stations and runs under the local system account.

### ***Installing the H+H Installer Services***

The setup program for the *InstallerMaster Service* is stored under %NMHome%\System\Install\MasterSetup. You are prompted to enter the installation path, the TCP port (for communication with clients) and the directory in which the private signature key (*key.pm*) shall be stored.



#### **Note**

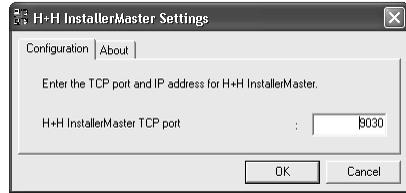
*We strongly recommend keeping a backup of this key; for example, on diskette or CD. All scripts are signed (certified) for installation in a LAN. A private key and a public key (the latter is stored on the clients; see below) comprise the encryption. If the private key is inadvertently deleted or overwritten by an update or new installation, existing scripts can no longer be executed.*

Once installed, the new service is listed under “Services” (*H+H Installer*) and a new option, *H+H InstallerMaster Service*, is added to the Control Panel with the following icon:



H+H InstallerMaster  
Service

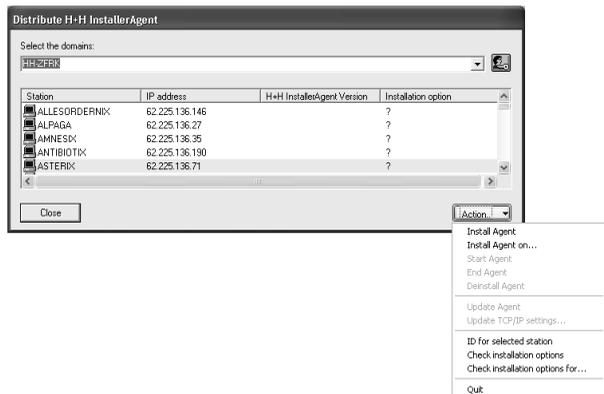
Double-click on this icon any time you need to change the TCP port:



There are two methods for installing the *InstallerAgent Service on the clients machines*:

- You can use the *insdistr.exe* distribution program, stored in the *InstallerMaster's Bin subdirectory*, for remote installation of the Client Service. When you start this program you are prompted to enter a user name and a domain. If you are already logged on with administrative privileges, you can skip this dialog. The next dialog, DISTRIBUTE H+H INSTALLERAGENT, gives you a choice of the following ACTIONS:
  - Install the Agent on client stations
  - Start or stop the Agent
  - Deinstall the Agent
  - Update the Agent (with a newer version)
  - Update the Agent's TCP/IP settings
  - Log on to a given client station
  - Check the available options for installing the Agent on a given client station

A green workstation icon in the DISTRIBUTE H+H INSTALLERAGENT dialog indicates that the Agent is installed on that station.



- You can use the *setup* program stored in the InstallerMaster's Bin\AgentSetup subdirectory for local client installation. This setup just

prompts you to enter the IP address and the TCP port of the InstallerMaster Service.

A public signature key, *cert.pem*, is copied to the client station under %NMWinDir%\InstallerAgent, so that the scripts can be decrypted and executed (see above). Another key stored in that directory, *hhcert.pem*, enables execution of scripts created by H+H on this machine. You can delete this key if you do not need it. Once the Agent has been installed, the new service is listed under “Services” and the *H+H InstallerAgent* is added to the Control Panel options with the following icon:



H+H InstallerAgent

Double-click on this icon to modify the IP address and TCP port of the InstallerMaster service:



### ***Using an Installer in a Terminal Server Environment***

In a terminal server environment, i.e. with Microsoft Windows NT/2000 Terminal Server Edition or Citrix MetaFrame, installing applications on individual workstations is no longer an issue, because applications are installed centrally, on the terminal server. But an Installer program has many uses in terminal server environments all the same; for example:

- If you have a server farm, using multiple MetaFrame servers, your application software has to be distributed among the servers.
- An Installer makes it much easier to reinstall applications, for example when the server is updated or reinstalled.
- It is of primary importance that a network server have as little downtime as possible. This is a good reason to run application setup on a separate workstation first, and then use the resulting installation script as the basis for installation on the server.
- Installing an application sometimes causes the target station to crash, which can have serious consequences when the target is a terminal server—all the more reason to create an installation script on a separate station first.

## ***How an Installer Works***

There are basically two different procedures that an Installer can use to monitor application installation procedures:

### ***Direct Monitoring***

One way to trace modifications made in a PC is through **System Monitoring**. With this method, the Installer “taps in” to the operating system and “listens in” on the commands called by the application setup program. This is a very effective method on the one hand, but on the other hand it is not particularly efficient, since *all* system calls are monitored. Furthermore, setup programs do not always follow general operating system conventions, and may make modifications that the Installer cannot register. Moreover, this type of Installer is operating-system dependent, which means that different versions are required for each of the Windows versions.

### ***Comparative Monitoring***

Another method of installation monitoring consists in comparing the system state of the workstation before the installation with the state after the installation. This method has the decided advantage that the same Installer program can be used with all 32-bit Windows systems.



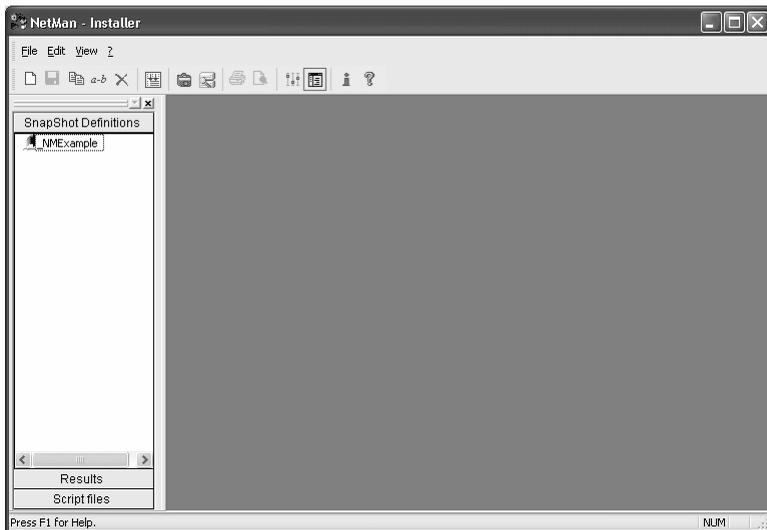
#### **Note**

*The NetMan Installer uses the “comparative” method. The records made of the system states are referred to in this manual*



## 2. NetMan Installer in Three Easy Steps

The first time you start the NetMan Installer, the following program window opens:

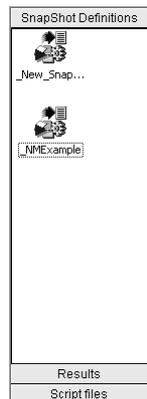


The selection bar on the left-hand side shows the headers *SnapShot Definition*, *Results* and *Script Files*. These elements are described in detail below.

### *SnapShot Definitions*

When you make a SnapShot of a workstation state prior to installing an application, the SnapShot does not have to include the entire workstation, nor the entire Windows directory, nor even the entire Registry. To optimize the entire procedure, from SnapShot to setup to comparison to script, you can simply select the relevant elements (directories, files, Registry branches, etc.) for monitoring, and save this selection as a *SnapShot Definition* which you can load at any time.

When you select the *SnapShot Definitions* element in the selection bar, you will find a predefined SnapShot Definition called *\_NME:example*, which you can use for all supported Windows operating systems. When you open this SnapShot Definition, the selected elements are shown in the window pane on the right. These include certain drives, directories and Registry entries. You can either use this Definition, copy the Definition and modify it, or create your own Definition.



The procedure and the options available for creating SnapShot Definitions are described in detail below, in sections 2.1.1, “Selecting Files”, and 2.1.2, “Monitoring the Registry”. Section 2.2, “Results of Comparison” gives you a good overall view of NetMan Installer features.

## Selecting Files

In a SnapShot Definition you specify the directories and files to be monitored during application setup. The Installer detects any files that are added, deleted or changed. There are two ways the Installer can detect modified files:

- By comparing the date and time given for the file
- By comparing file versions: many file types, such as DLL and EXE files, register an internal version number

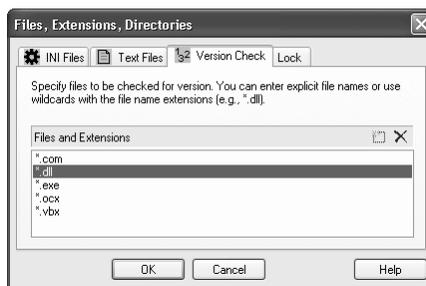
With some types of file, such as INI files (which contain Windows application settings), it is important to know not only whether they have been modified but also **what** modifications were made. If the Installer detects changes in the ‘system.ini’ file, for example, this does not mean the entire file should be copied over to another machine just to install the application—in fact, that might cause a fatal error on the target machine. It is far more important to find out exactly which of the changes made in a given file are required for installation. The NetMan Installer can detect these modifications within files. For this function, the Installer supports two formats:

- Files with the INI structure
- Common text files

You can use the buttons in the toolbar in the individual SnapShot Definition window to activate and deactivate the functions for *Checking INI Files*, *Checking Text Files* and *Checking file versions*, and to specify which files are checked.



The following is a brief description of the **Files, Extensions, Directories** dialog, which is opened by the last of these 4 buttons. For more detailed information, please refer to the on-line Help.



Here you can define which types of file are checked for new versions, and which INI or

text files are checked for changes in content.



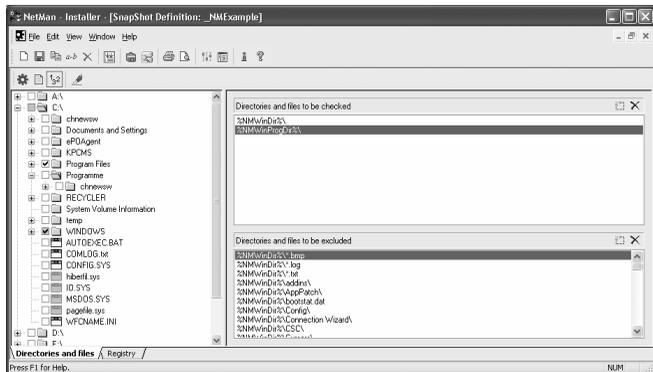
## Tip

Click on the *Lock* tab to enter the names of files, directories and drives that you wish to exclude from monitoring. For example, it is a good idea to exclude your CD-ROM drive(s).

The next two subsections describe the file selection process in detail. When you first open a new SnapShot Definition, no elements are selected. There are two ways to select the desired elements:

- by selecting them in the display
- by entering file and directory names

### Selecting Elements in the Display



To select a file or directory, click in the checkbox next to it. Click again to de-select the element. The checkboxes are shown with or without a checkmark, and with or without background shading, to indicate one of the following 4 states:

- The file or directory is not selected and thus will not be monitored.
- The file or directory is selected and will be monitored. In the case of a directory, a checkmark without shading indicates that the selection includes all files within the directory, as well as all subdirectories and their files.
- The directory is selected, but contains elements that are excluded from monitoring.
- In this case, the directory itself is not selected but contains elements that are selected for monitoring.

The  and  checkboxes result from a direct mouse-click, but the shaded checkboxes,  and , occur only when other elements are selected—not the ones adjacent to the shaded box.



### Note

The Installer uses two different symbols to represent files:



Visible file



Hidden file

## Entering File and Directory Names

In working through the previous section you will have observed that every time an element was selected or de-selected in the left-hand side of the SnapShot Definition window, an entry was added to or deleted from one of the right-hand window panes, DIRECTORIES AND FILES TO BE CHECKED OR DIRECTORIES AND FILES TO BE EXCLUDED. You also have the option of entering directory and file names in these panes manually. The \* and ? wildcards are permitted.



### Note

NetMan automatically replaces files and directory names it recognizes with environment variables. For example, C:|Program Files is automatically replaced by %NMWinProgDir%. This helps make the configuration system-independent.

## Monitoring the Registry

To configure settings with respect to the Registry, click on the REGISTRY tab in the SnapShot Definition window. The procedure for selecting keys, data types and values from the Registry is analogous to the selection of directories, subdirectories and files from the directory tree. One difference, however, is in the functions available in the right-hand panes; you can delete entries here, but not add or edit entries through manual input. Since the entry names in the Registry are generally so long, we figured

it was unlikely anyone would wish to enter them by hand.

Thus Registry entries are selected by marking them in the display on the left.



#### Note

We strongly advise *against* selecting the entire Registry. Some keys exist in more than one main branch, or 'hive', in the Registry. For example, it is important to include the HKEY\_CURRENT\_USER hive, but then you do not need to select HKEY\_USERS, primarily because the former contains latter, and also because setup routines do not usually modify HKEY\_USERS values, but rather store values only for the current user. We would like to emphasize at this point how helpful it can be to read the reference literature listed in section 1.3.1, "Recommended Reading on the Windows Registry". A good working knowledge of the Registry makes it much easier to decide which keys you should select for monitoring. Although there are a number of pre-defined selections for SnapShot Definitions, we do recommend that you familiarize yourself with the structure and functions of the Windows Registry.



#### Tip

Because the Windows Registry is in part made up of Windows system files, you should exclude these files from monitoring.

## Results of Comparison

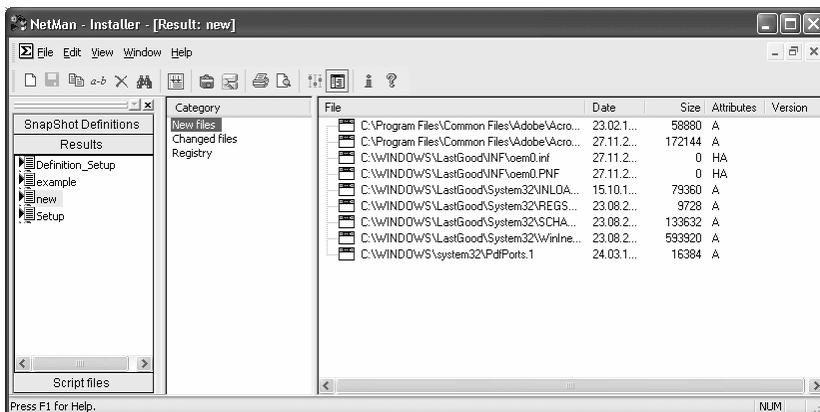
The Installer begins by taking a SnapShot of a computer state, including the areas specified in the SnapShot Definition. This is the "Before" picture. After the application has been installed, another SnapShot is taken, using the same definition. This is the "After" picture. The next step is a comparison of Before and After; the difference between the two is the *Result*. This is what tells you how the workstation was altered by the application setup routine.



#### Note

How you get from the SnapShot Definition to the Result is described step by step in Chapter 3, "From SnapShot to Script: 3 Sessions with Bob".

The *Result of the comparison* is displayed as follows:



The Result is divided into the following *Categories*:

- New files
- Deleted files
- Changed files
- INI files
- Registry
- Text files

These categories are listed on the left, with the corresponding results displayed on the right. The example above shows the new files that the setup routine copied to the Windows directory. In the Results window, you can copy, rename and delete an entire Results file. The individual elements within a Results file, however, can be edited only in the script file (see below).

### Script Files

A script file is generated from the results of the comparison, and can be edited manually in the NetMan Installer program. You can make changes in the script file right down to the element level, and define whether individual files should be copied or deleted or, for example, specify whether certain Registry key values are changed or not.

You can integrate the resulting script in a NetMan configuration by simply adding a Script action and entering the file name (see Chapter 5, "Integrating Scripts in the NetMan Client"). This mechanism allows you to distribute applications over the network with transparent, automatic installation on workstations. You can also distribute applications within MetaFrame server farms in the same manner.

**Discussion**

Script files contain the source text of a script language which is interpreted by NetMan at runtime.

```
//Installer script

#Esc
Declare
Number nIncluded = 1
Number nActType = 1
Number nSoftMount = 1
Number nTest = 0
Number nCont = 1
Number nReturn = 0
Number nTypeCount = 0
String cParam = ""
String cText = ""
String cTitle = ""
String cAppID = ""
EndDeclare

Main
HHWSetShowProgress(TRUE)
//ATTRIBUTES
//COPYFILES
HHWAddCopyFile("%NMHome%\BIN\INSTSCPT\Financial
Times\Ctl3d.dll", "%NMWinSysDir%\Ctl3d.dll", 2)
nIncluded =1
HHWCopyFiles("")
//DELETE FILES
//CONFIGFILES
//REGISTRY
//ActRun(34, "K:HKEY_LOCAL_MACHINE\SOFTWARE\ProQuest
Information and Learning\The Financial Times\1.00.000\ /
V: /T:16", "", "", 1, cAppID, 0)
nTypeCount =0
nIncluded =1
//ActRun(34, "
K:HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App
Paths\chnews.exe\Path /V:C:\chnews /T:8", "", "", 1,
cAppID, 0)
nTypeCount =0
```

```
nIncluded =1
//ActRun(34, "/
K:HKKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\App
Paths\\chnews.exe\\ /V:c:\\chnewsw\\chnews.exe /T:8", "",
"", 1, cAppID, 0)
nTypeCount =0
nIncluded =1
ActRun(15, "regedit.exe /s
\\%NMHome%\\BIN\\INSTSCPT\\Financial Times\\Financial
Times.nmr\\", "", "", 1, cAppID, 0)
EndMain
```

NetMan Installer script files can also contain NetMan environment variables, such as the *%NMWinSysDir%* variable in the above example. *This feature makes it even easier to use these scripts for installing applications on other computers.* In this example, it does not matter whether the operating system on a given machine is installed on the C:, D:, or M: drive, or whether the directory in question is called *WINDOWS\* or *WINNT\*. The script language also supports system variables, as well as control structures such as if.

---

### 3. From SnapShot to Script: 3 Sessions with Bob

Now that you have been introduced to the selection bar on the left-hand side of the Installer program window, with its *SnapShot Definitions*, *Results* and *Script Files*, we will show you exactly how you get from the SnapShot Definition to the Result file and from there to a Script file. These tasks are carried out by the NetMan Wizard, Bob.

#### *SnapShot of the Workstation*

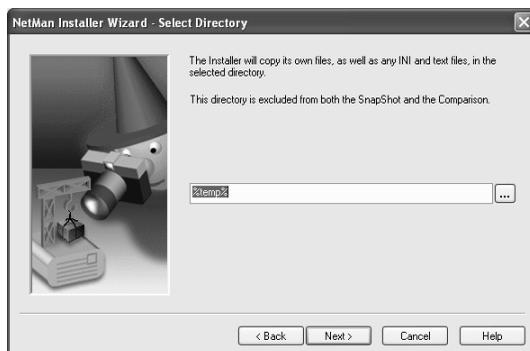
Once you have defined the range of the SnapShot, you are ready to produce a picture of the current state of the workstation. To do this, select FILE | INSTALLER WIZARD | SNAPSHOT... from the menu bar, or click on the camera icon, shown here, in the toolbar:



This opens a window with Bob the Wizard:



The next window prompts you for a directory where Bob can save the Installer log files. Generally, you can simply accept the default, *%temp%*.

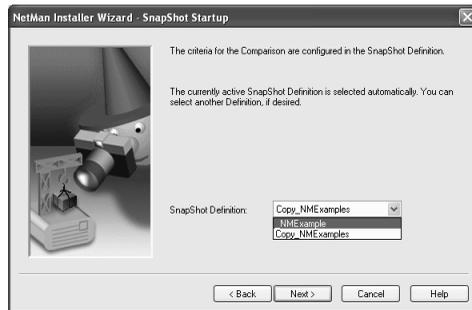




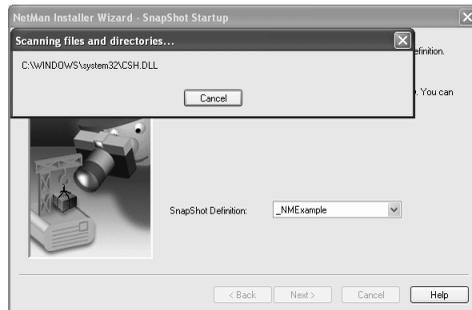
## Note

*Under Windows XP, using the %temp% variable can lead to problems when the next user, after reboot, logs on under a different name, as the 'temp' directory is generally stored in a user profile. For this reason, you need to enter a specific path name if you are running Windows XP.*

The next window prompts you for the name of the SnapShot Definition you wish to use for the subsequent application installation.



Now the SnapShot is taken:



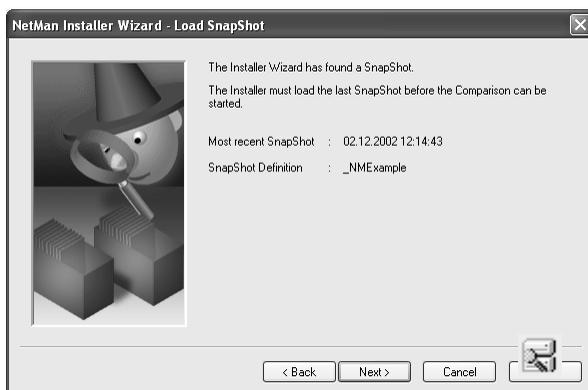
When the SnapShot is completed, you can run the setup program for the application you wish to install.



## Comparing System States After Installation

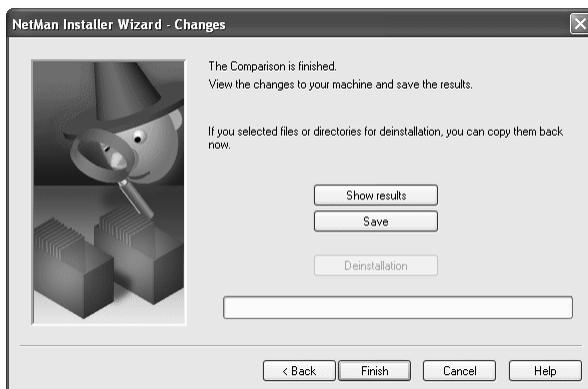
The second step, comparison of the “before” and “after” states, is performed directly following the application installation. To do this, select FILE | INSTALLER WIZARD | COMPARE from the menu or click on the Compare icon, shown here, in the toolbar:

This opens a window that displays the name, date and time of the last SnapShot created, as well as the Definition used:



Click on 'Next' to confirm this data and start the comparison.

The next dialog box lets you view the results and save the results file, if desired:



The Results file is shown in the *Results* section of the selection bar.

## Generating a Script from Results

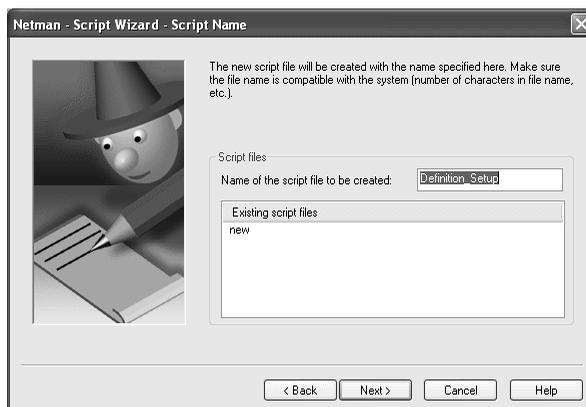
You can use the results of the comparison between “before” and “after” states on the workstation to generate an installation script, which you can integrate in a NetMan configuration for seamless transparent installation of the required components on a client workstation. To make a script, select **FILE / SCRIPT WIZARD** FROM THE MENU BAR OR CLICK ON THE **SCRIPT** BUTTON, SHOWN HERE, IN THE TOOLBAR.



Bob the Wizard is back again, this time prompting you for the name of the Results file that you would like to make a script from:

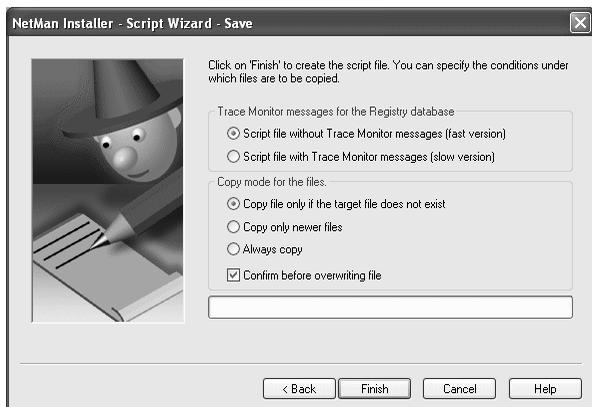
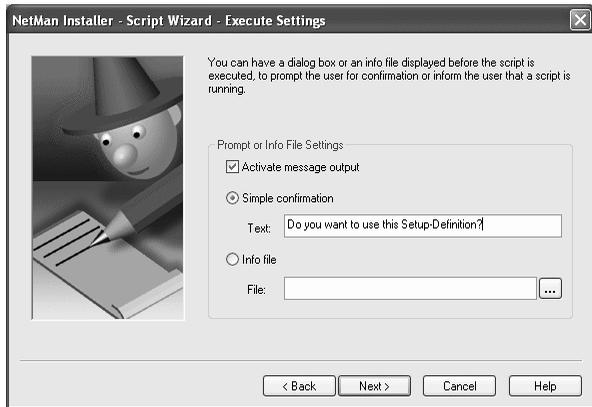


In the next window, you can enter a name for the script file.

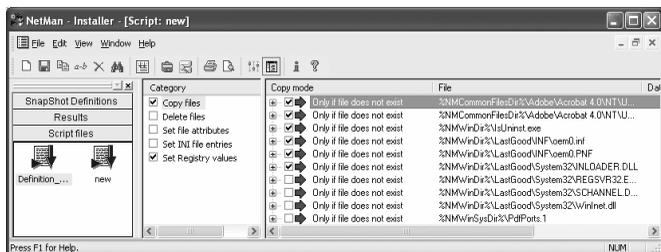


## Generating a Script from Results

The next two windows offer you a number of options for generating the script (refer to the on-line help for details):



Once the script has been generated, it is shown in the SCRIPT FILES section of the selection bar.



You can edit these script files manually, if desired. For example, you may want to exclude some of the files from being copied into the client Windows directory, or there may be some changes to the Registry that are not necessary.



#### Note

*The automatically generated scripts can be significantly improved by subsequent manual editing. It is essential, however, that you check very carefully to determine whether a given element can be deleted or modified.*

The types of modification possible in a script file can be summed up as follows:

- Deactivate, delete or add linked files and/or Registry entries
- Enable/disable a prompt for confirmation before copying files
- Change copy mode for individual files and Registry entries
- Configure file attributes for individual files

For detailed information on editing script files, please refer to the on-line Help.



#### Tip

*In the Script Editor you can use Drag&Drop to copy files from other windows (such as the Windows Explorer) to the desired category (such as Copy/Delete files or Set file attributes/Set INI file entries). When you add a file to be copied, the file in question is copied to the script directory and is thus available any time the script is executed. When you Drag&Drop INI files (including any files with the INI-structure, such as CFG files), you can get a quick overview of the INI file structure. You can also use Drag&Drop to copy files from one Installer script to another, but only when files are copied from a given category in one script to **the same category** in another script.*

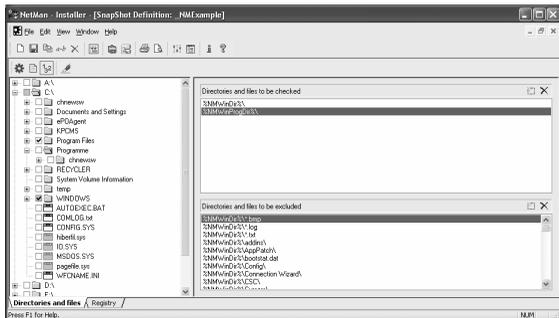
## 4. Installation with the NetMan Installer – a Demonstration

In this chapter, we will carry out an installation step by step using the NetMan Installer. For this demonstration we will install the same application used for demonstrations in the Base Module manual, the “Financial Times” application (from 2002), which is driven by the “Caravan” retrieval software from Chadwyck-Healey.

“Financial Times” is a CD-ROM database and a 32-bit program.

### The SnapShot

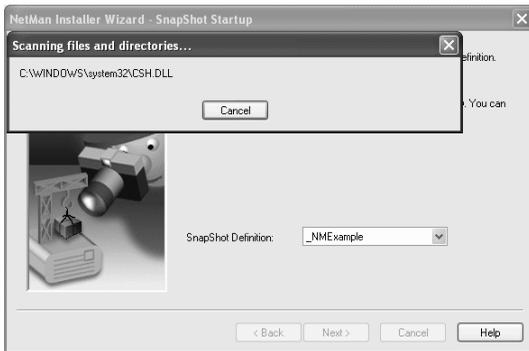
We want to install “Financial Times” on a computer running Windows XP Professional. For the SnapShot Definition, we select a version of the sample definition, “\_XP” which has been modified for use with the computer in question. These modifications include, for example, the exclusion of several subdirectories under %SystemRoot%, as well as drives A: (floppy drive), E: and F: (CD drives). The “Check File Version” function is switched on.



We activate the SnapShot, ..



and the Installer Wizard reads the data in directories, files and the Registry as specified in the SnapShot definition.

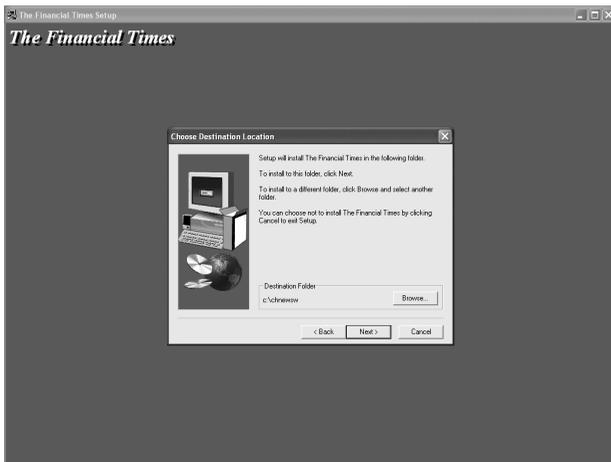


When the SnapShot is completed, the Wizard outputs a message that the resulting SnapShot will be the basis for all subsequent comparisons.



## The Application Setup

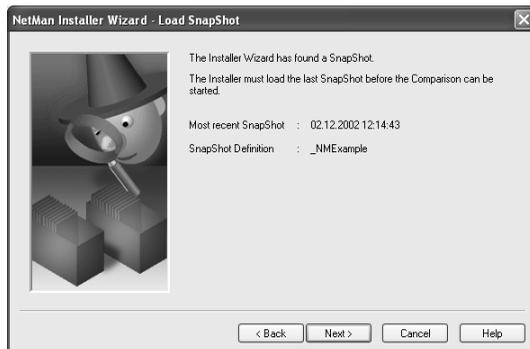
Now we install the “Financial Times” application.



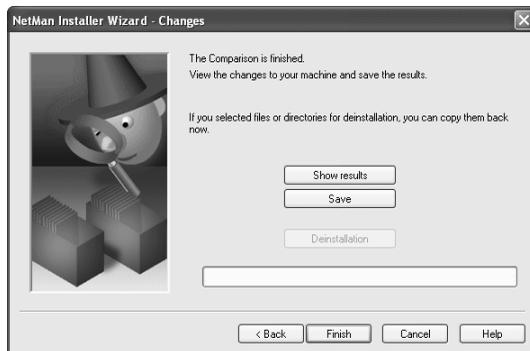
## From Comparison to Results

We initiate the comparison immediately following the installation of the application so we can find out what modifications are necessary before the application can be started for the first time on a client station. If we later find that other settings are configured following the initial startup, such as program options, disabling certain operating functions so users cannot access them, start options, etc., we can run the comparison again after making these changes.

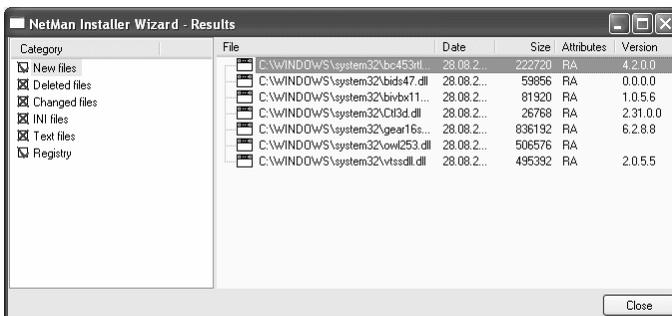
To run the comparison, select FILE | INSTALLER WIZARD | COMPARE from the menu, or click on the Compare button in the toolbar.



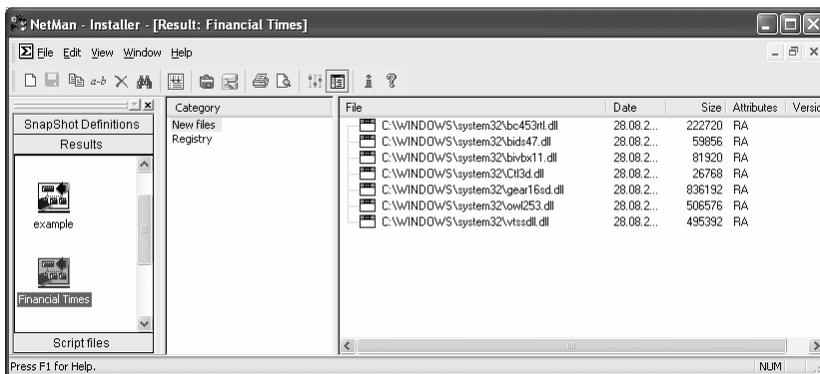
The Snapshot is loaded and the comparison made.



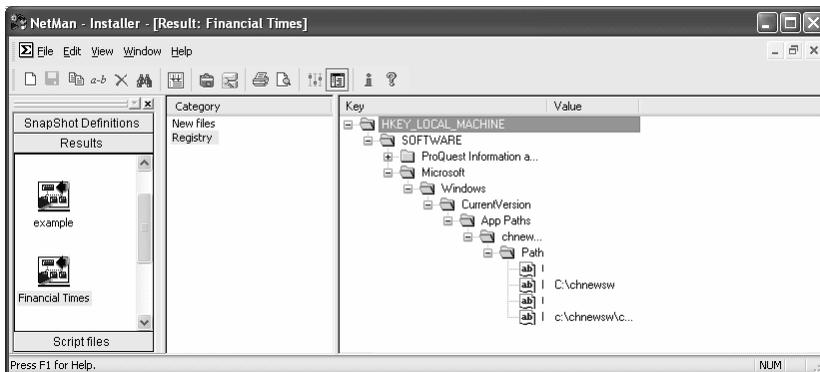
We save the results and then view them.



We find that some new DLL files have been installed...



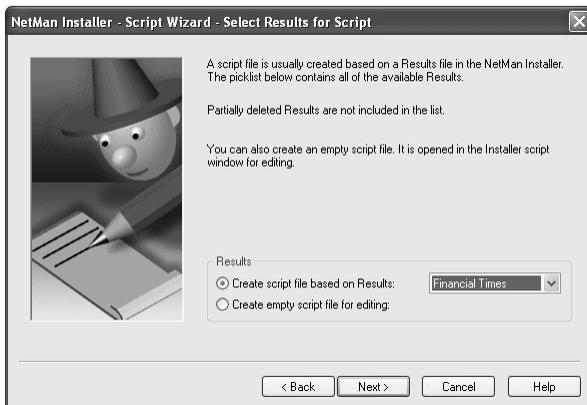
... and we find a number of changes in the Registry:



## The Script

We now select FILE | SCRIPT WIZARD to create a script for “Financial Times” from the results.

We select the Results that will form the basis of the script...



and enter a name for the script file:



## Editing the Script

We have already mentioned how important it is to edit the Installer script so that it contains only the necessary changes. In many cases, it is clear at a glance that some of the changes recorded have nothing to do with the installation of the application; sometimes there are changes listed which need not—or should not—be distributed to other workstations. You can delete or disable these entries when you edit the script. A easy way to check whether there is any need to edit the script is to try calling the application on a workstation on which it has not been installed.



### Note

*If you do not know what a given file or Registry entry does or whether it is necessary, make sure the script does not distribute it on your workstations.*

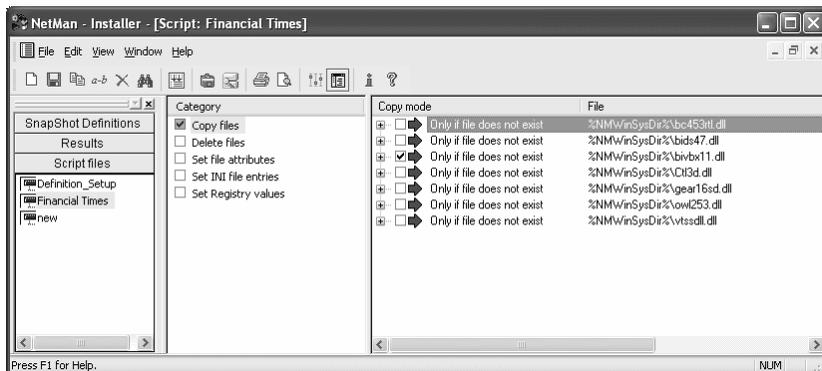
The installation of “Financial Times” has been with us since the NetMan “Base Module” manual. You may recall that we discovered in Chapter 4, under “Your Second Application: Financial Times” we discovered that the application runs without having its Registry entries copied to the individual workstation. Furthermore, all program-specific DLL files can be moved to the program directory; only the ‘Ctl3d.dll’ system file has to be copied, if it is not already on the target workstation. We now select the SCRIPT FILES section of the selection bar, open the new “Financial Times” script, and edit it so that only the required files are copied.



### Note

*The Script Wizard has already replaced all path entries that are entered in the NetMan Environment with NetMan environment variables. This helps makes the script system-independent.*

Now we can edit the script. We disable the copying of Registry entries and the program-specific DLLs:



## Tip

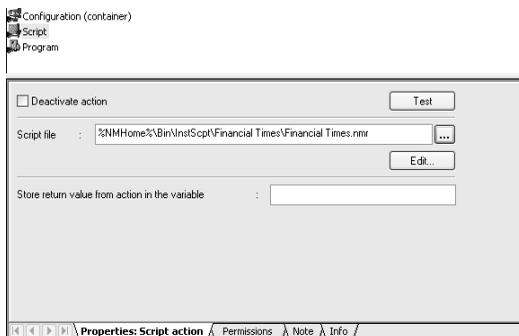
As you gain experience in using the Installer and editing scripts, it will become easier to evaluate the Results and decide how best to modify the script. Script entries for copying font files to the installation machine can usually be disabled, for example, because applications generally have no problem using default fonts if their own fonts are not installed. Moreover, the EXE and DLL files installed during setup can often be distributed by installing them in the new application's own working directory.



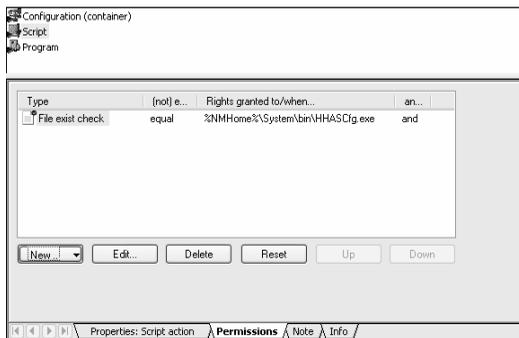
## 5. Integrating Scripts in the NetMan Client

To integrate an installation script in the NetMan Client, run the Management Console and add a *Script action* to the NetMan configuration for the application that the script installs. Position the Script action so that it is executed before the Program action.

In the following example, we will integrate the installation script created for the “Financial Times” application. Start the Management Console and select the NetMan configuration for the “Financial Times” application. Move the focus to the *Program action* and add a preceding *Script action*. In the **SCRIPT FILE** field, enter the full name of the “Financial Times” installation file:



As discussed in the “Base Module” manual, we recommend assigning “execute” rights to the Script action so that the DLL file is copied only if it does not already exist:



“Financial Times” can now be launched on any workstation in the network by simply clicking on this configuration in the NetMan Client.

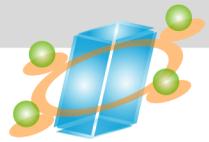


### Tip

*If you only use the script to install the application on a Terminal Server, you do not need to add a Script action to the configuration. Since the script only needs to be executed once, you can run it by simply right-clicking on the script in the NetMan Installer program.*



**H+H NetMan® XP Terminalserver Module**





# Table of Contents

---

1. Introduction .....	1
Contents of This Manual .....	1
Prerequisites .....	1
Performance Features .....	1
Discussion: NetMan and Terminal Servers in Intranets and the Internet .....	2
Station Names under Terminal Servers .....	5
2. NetMan Settings for Terminal Servers .....	7
Examples of Startup Configurations in Terminal Server Sessions .....	7
Anonymous User Logon in the Domain .....	7
Mapping Client Drives .....	9
Defining the Maximum Number of Parallel Sessions .....	10
3. Starting NetMan in the Terminal Server Environment .....	13
NetMan as Shell .....	13
Launching NetMan Configurations Without a NetMan Shell .....	13
Using Citrix ICA Files to Start NetMan .....	14
Starting NetMan from HTML Pages with Citrix Web Clients .....	17
Application Launching .....	18
Embedded ActiveX Controls (MS Internet Explorer) .....	19
Embedded Netscape Plug-ins .....	20
Browser Detection (JavaScript with Auto-start) .....	20
Browser Detection (JavaScript without Auto-start) .....	24
Embedded Java Applets .....	24
4. Special NetMan Features for Terminal and MetaFrame Servers .....	27
Terminal Server Access Control .....	27
Program Function .....	28
Notes on the Interaction Between HHASAcc and the NetMan Client .....	28
Access Control Functions in the Trace Monitor; Testing .....	31
Changing the Operating Mode: HHASMode.exe .....	31
5. Notes on Troubleshooting in the Terminal Server Environment .....	35
Problems Starting NetMan .....	35
Problems with an Application .....	36



## 1. Introduction

For instructions on how to install and activate this module, see Chapter 2 of the “*Base Module*” manual.

### *Contents of This Manual*

This manual is divided into the following chapters:

- Chapter 1, “**Introduction**”, describes the system requirements and performance features of this module and contains a Discussion of the interaction between NetMan and terminal servers.
- Chapter 2, “**NetMan Settings for Terminal Servers**”, describes the NetMan settings available for use with terminal servers, and helps you determine which of these settings are best for your system.
- Chapter 3, “**Starting NetMan in the Terminal Server Environment**”, describes different ways of calling NetMan during terminal server sessions and discusses the advantages and disadvantages of the various interfaces available for these purposes.
- Chapter 4, “**Special NetMan Features for Terminal and MetaFrame Servers**”, describes how NetMan expands your terminal server functions.
- Chapter 5, “**Notes on Troubleshooting in the Terminal Server Environment**”, offers you practical examples in the form of NetMan configurations that can be particularly useful in terminal server environments

### *Prerequisites*

The NetMan Terminal Server module supports Microsoft Windows NT 4.0 Terminal Server (Service Pack 3 and later) and Citrix MetaFrame versions 1.0 and later.

### *Performance Features*

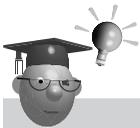
The Terminal Server module offers the following features:

- It allows you to run NetMan in terminal server environments
- It adds an access control function for terminal servers
- It reduces the work involved in managing virtually published applications in your Citrix MetaFrame server (when you use the HTML View module). If you choose to run NetMan both in your LAN and on terminal servers, no additional program configurations are necessary to have NetMan serve any application through the terminal server while it is already running in the LAN.
- When your terminal server is accessed through the NetMan HTML View module, the NetMan Terminal Server module improves security by providing increased desktop control and *secure distribution* of ICA files.
- You can define the number of parallel sessions allowed for each user, for better control and distribution of terminal server resources.



- It makes it easy for you to observe the terms of your license agreements with manufacturers of the applications started through NetMan.
- It monitors the frequency and duration of application use in intranets and the Internet.
- It allows you to restrict the use of anonymous published applications to specified users and stations. In conjunction with the NetMan HTML View module, you can also specify which applications are presented to which users.
- Used in conjunction with the HTML View module, the Terminal Server module allows you to permit or deny access based on the operating system that the client is running.

If the advantages of these features are immediately clear to you, then you can skip the following Discussion of NetMan's performance in terminal server environments. If, on the other hand, you think that a terminal server is like a single workstation that happens to be used by several people at once, and that software such as NetMan (=Network Management) is therefore unnecessary—or if you would like to read a detailed description of the performance features listed above—then we recommend reading the following Discussion.



## Discussion

### ***NetMan and Terminal Servers in Intranets and the Internet***

The advantages of a terminal server in a low-bandwidth system are well-known; the benefits of terminal server technology are often discussed using buzzwords such as “thin clients” and “total costs of operation” (TCO). When we designed NetMan, we knew that the use of terminal servers would become more widespread as time passed, and we implemented functionalities that will help you optimize your total costs of terminal server operation, while eliminating or reducing the difficulties and flaws often encountered with terminal servers.

In an article from iX magazine, Rainer Ganser describes some of the difficulties involved with security:

*“Questions of security naturally play a greater role in a multi-user environment than on a stand-alone PC. Because the Hydra user is not by default restricted in his rights, he can call and modify almost anything, just as though he was sitting at the server; this makes security measures even more important [for terminal servers] than for file servers or print servers.[...] Microsoft has set itself a lofty goal where security is concerned: their Security Manager is intended to protect the Hydra in conformity with C2 security specifications. One should be warned, however, against the highest level of security: if you choose this level, you will find that access rights are so restrictive that even a logon is impossible—because*

*the Windows NT Explorer, which functions as the desktop, is not permitted to start..."*

*(Rainer Ganser, Frank Mühlenbrock: HydraNT, iX 1998, 4, p. 57)*

You can eliminate this problem by using the NetMan Client as your shell, because each user is automatically restricted to those applications to which they have explicit permissions. This is what we referred to above as „increased desktop control“.

The skeptical view of security in the article above applies primarily to the MS Terminal Server, not to the Citrix MetaFrame add-on. With Citrix, you can publish applications for remote user access. You can also limit client access so that only published applications are available; this greatly increases your terminal server security. Another effective method for lowering TCO is the use of anonymous published applications. With this mechanism, you can make these applications available to an unlimited number of users without additional administrative work. The disadvantage, however, is that you no longer have control over server access: anyone who can reach the terminal server through a network can also access it. Here, too, NetMan gives you additional control features. For example, you can permit or deny server access, or limit the client to certain applications, entirely on the basis of the client's IP address.

If users access your server through the HTML View module, NetMan offers even more advantages. Your terminal server is protected by NetMan's *secure distribution of ICA files*. The only access to your terminal server permitted is through the links that you offer in your Web server. Even an expert user with an ICA client and the required technical knowledge cannot use the ICA files in your Web server links to access your terminal server. This increased security goes hand-in-hand with a reduced administrative workload: rather than publishing each of your applications, which can be a lot of work if you have a large number of applications (or perhaps load-balancing terminal servers), you can publish a single application, NetMan, as a gateway to all the NetMan configurations on your terminal server. In other words, the NetMan configurations are virtually published.

Another problem with terminal servers is the lack of control over the number of user sessions: Once users have access to your terminal server, they can open as many sessions as they like. Especially if your applications are published individually, you can easily see how users may wind up starting a new server session every time they want to open another application—which in turn uses up a lot of server resources. When you use NetMan as the shell for terminal server sessions, however, users can call as many applications as they like during a single session. When NetMan configurations are activated individually without starting the NetMan Client (for example, through the HTML View), you can prevent or limit multiple server sessions by defining the number of multiple sessions allowed. In NetMan this definition can be global or user-specific.

In conclusion, we would like to mention the NetMan Base Module performance features (see “Additional Program Properties” in Chapter 5) that play a role in the terminal server environment. The typical structure in such an environment is characterized



by a lack of information about the stations and users that access the terminal server; the licensing and event log functions in the NetMan Base Module bridge this gap:

*“... With regard to application licensing, on the other hand, multi-user operation should not present a problem—at least it has not thus far. As a rule, application licensing is based on the number of CPUs that run the application. And since the Windows Terminal Server runs programs on the server rather than, say, copying them to client memory and letting them run on the client machine, the terms of a single-license agreement are being observed, at least according to the wording of the contract. Some network software licenses, on the contrary, assume that local copies are created on the client machine—either directly (if partially) on the hard disk, or in working memory. It is safe to assume that program manufacturers, should the modern terminals become popular, will soon be changing their licensing conditions and count licenses per user rather than per CPU.” (Eric Tierling: *Gezähmtes Monster. Terminals und schmale Clients mit Microsofts Terminalserver und Picasso von Citrix.**

*[Taming the Monster: Terminals and Thin Clients with the Microsoft Terminal Server and Picasso from Citrix.] c't 1998, 10, S. 232).*

You may rest assured that the authors' assumption is correct. In this respect, NetMan makes it *easy to keep application use within vendor's licensing limitations*: you can simply store the number of licenses as an additional program property of a NetMan configuration. These properties are valid every time the application call is launched, whether the application runs on the machine where it is installed or is called from the Internet using the HTML View module. If no licenses are available for a given application, NetMan informs the user and assigns them a position in a waiting list.

The Statistics program in the NetMan Base Module provides functions for statistical evaluation of application use data, giving you as administrator comprehensive information concerning your application licenses. You can see when your licenses were all in use (maximum value recorded for parallel use), how often users were put on a waiting list for an application, and how often users canceled their application call rather than wait for a license. The application use data lets you draw important conclusions which can form the basis of organizational decisions, by answering questions such as:

- Do you have more licenses than you need for a given application? Do you have too few licenses for another?
- Which stations and which users call which applications?
- Does the use of a given application justify the cost of its acquisition? How can you best distribute operating costs for the application over your budget?

## ***Station Names under Terminal Servers***

In most aspects, the operation of NetMan in a terminal server environment is no different from its operation in a LAN. One difference, however, is the way station names are assigned in the terminal server environment. As you read in the “Base Module” manual, NetMan obtains a unique ID for each station from the network operating system. Depending on your selection on the USER ID/STATION ID page of the NetMan Settings, the station ID is either the user-defined computer name assigned under Windows, the network card address, the IP address or the full name stored in the DNS server. In a terminal server session, the station ID is obtained from the “Client Network” on the local client machine. Station IDs are recorded for a number of purposes in the NetMan program, including:

- listing currently active stations
- monitoring license use
- assigning access permissions to applications and NetMan configurations
- optional inclusion in the event log
- calculating application-use statistics according to station

Since NetMan obtains the station ID from the network operating system, and the network requires a unique station designation, the uniqueness of the NetMan station ID is always assured by the network operating system in “normal” network operation. Only a single instance of NetMan can run in a LAN. With terminal servers, however, a single LAN station may open multiple terminal server sessions and may start more than one instance of NetMan.

There may be situations in which it makes sense to allow multiple parallel NetMan sessions; for example, if two NetMan configurations are activated through the HTML View module. You can define whether parallel NetMan sessions are permitted, and how many parallel instances are allowed at one time.

In order to distinguish station IDs in multiple parallel sessions, a session number is appended to the station ID with the format “#n”. For example, if NetMan establishes “MyComputer” as the station ID, then the station ID in the first terminal server session is “MyComputer#1”, in the second session “MyComputer#2”, and so on.



## 2. NetMan Settings for Terminal Servers

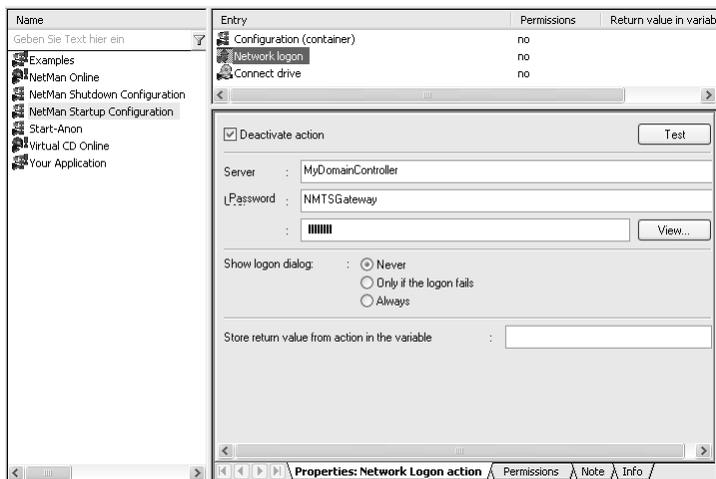
Although NetMan operation in a terminal server is basically the same as in a LAN, there are a few additional options available in NetMan Settings and, depending on your server configurations, there may be some differences in the program startup routine when using NetMan in combination with a terminal server. The following sections describe these differences and offer some examples of additional NetMan settings.

### *Examples of Startup Configurations in Terminal Server Sessions*

You may wish to add special functions to your NetMan startup configurations for use in a terminal server session. Make sure you define the NetMan permissions for session-specific NetMan actions in such a way that they are processed only during terminal server sessions. The following examples may give you some ideas for your own configurations.

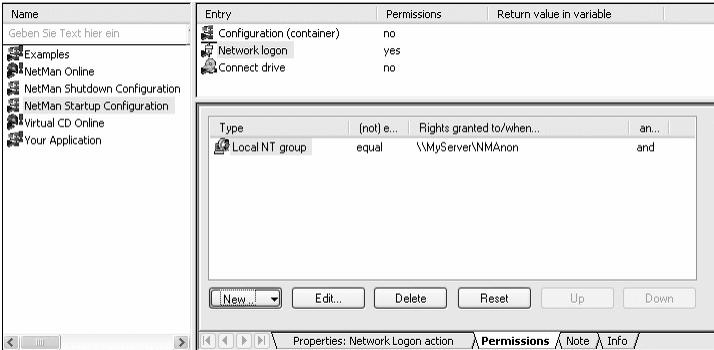
#### *Anonymous User Logon in the Domain*

In MetaFrame (versions 1.8 and later), anonymous users have no permissions to domain resources when the “Guest” account is deactivated. If you wish to allow access to some resources for the anonymous user, you can add a Network Logon action to the NetMan startup configuration (see “Expanded Action Options” in Chapter 5 of the “Base Module” manual):

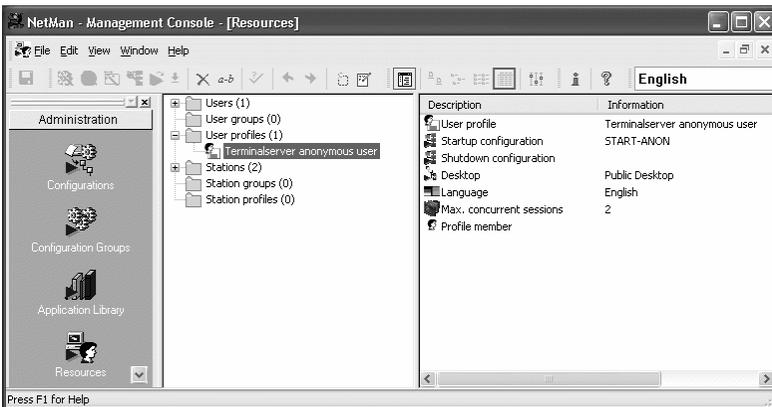


This example is based on an account with the user name “NMTSGateway”. This account was created especially for this purpose, using NetMan’s Terminal Server Access Control program (see Chapter 4 for details). The dialog box above shows a NetMan

configuration that logs the anonymous user onto servers in the “MyDomain” domain. The logon action active in this illustration logs the user onto the server called “CD-Server”. Since anonymous users have temporary passwords, it would be difficult to replicate anonymous users in the domain database; fortunately there is usually no reason for such replication. To make sure this logon is accessed only by anonymous users, the “execute” permissions for this action are bound to a user group on the terminal server created for anonymous users, called “MyTerminalServer”:

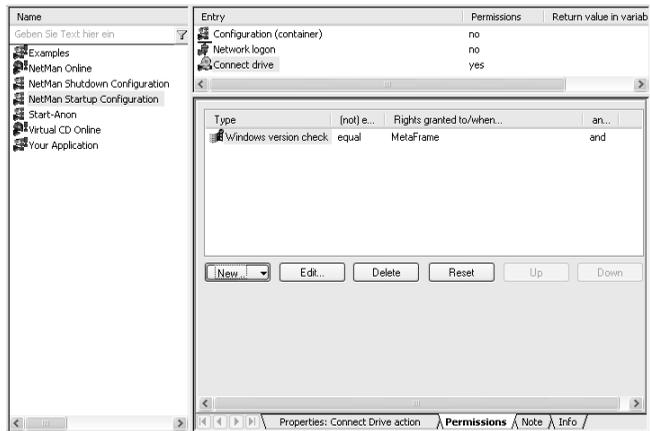


Alternatively, you can store these special startup configurations in a separate startup configuration. In the example below, these configurations are stored in the “Start-Anon” configuration, which is assigned to a certain NetMan user profile. This has the advantage that user privileges do not have to be analyzed for these actions (as shown in the dialog pictured above).

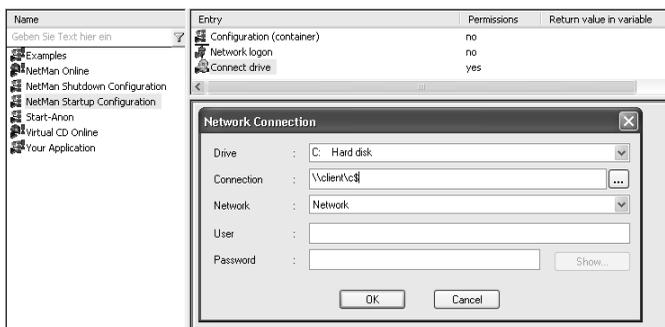


## Mapping Client Drives

Terminal servers offer the option of referring to local resources on the client station during terminal server sessions. For example, you can either activate the “Connect client drives at logon” option in the user configuration, or you can map the required drives in a login script. Another option is to map client drives in a NetMan startup configuration. If you do not want the same drives mapped at every startup and for every user, you can assign “execute” permissions to the action accordingly:



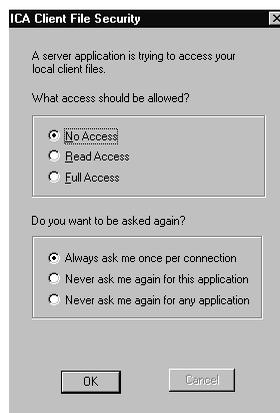
In the window shown below, the “execute” permissions assigned to the drive mapping (“Drive Connect”) action ensure that it will be executed only if the client station is a terminal server.



If NetMan is started on only one terminal server and this action should always be executed, then you do not need to assign “execute” permissions to the action.

When a user starts a terminal server session by selecting a link in an HTML document, the following dialog opens:

If “No Access” or “Read Access”, rather than “Full Access”, is set in the first section, it will not be possible to write data to the local hard disk during this session; in other words, the user cannot save data locally.



## Note

If the user accesses the terminal server from a browser using a Netscape plug-in or ActiveX controls (see “Starting NetMan from an HTML page using the Citrix Web Client” in Chapter 3), and answers the first question shown here with “No Access” and the second with “Never ask me again for any application”, these settings remain valid for all sessions because they are written in the WEBICA.INI file in the Windows directory. The setting can be changed only by modifying or deleting this INI file. For example, you might modify the INI file to read “GlobalSecurityAccess=405”; this is equivalent to “Full Access” and “Never ask me again for any application”.

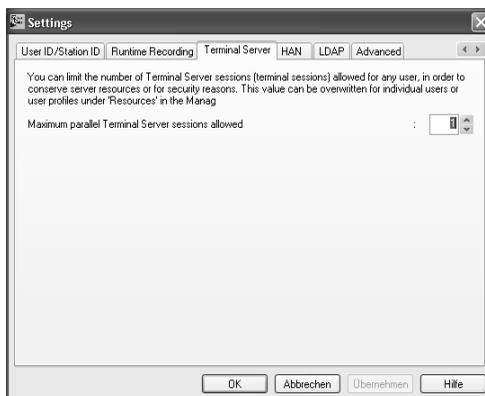
## Defining the Maximum Number of Parallel Sessions

If your NetMan configurations are launched on a terminal server, you may wish to limit the number of parallel sessions allowed for a given workstation. You can define a global limit as well as different limits for individual users and user profiles.

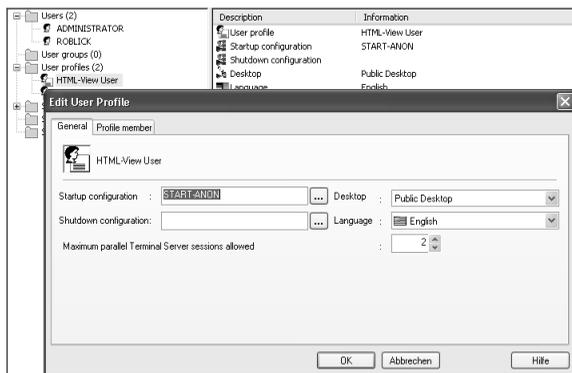
If there are differences between the number of parallel sessions as defined at different levels, the global setting acts as a default value, a user-profile setting overrides the global setting, and the user setting overrides both of these settings.

Example:

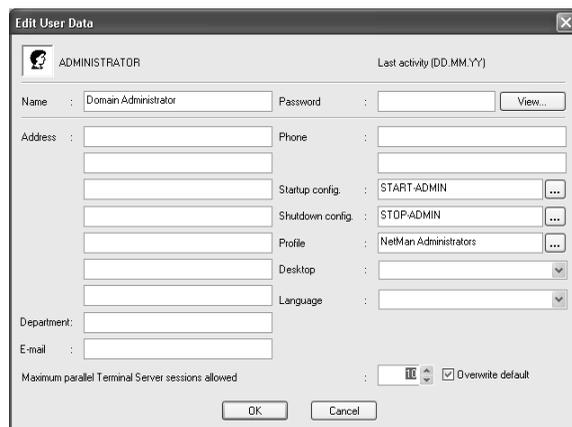
Generally, you want to block multiple parallel terminal server sessions for all users. Thus you have not changed the default setting of “1” on the TERMINAL SERVER page of the NetMan Settings:



You do want to allow parallel sessions, however, for those who use the NetMan HTML View as their interface. For this purpose, you have created a user profile called “HTML View User Group”, and now you set the maximum number of parallel sessions for this group to “2”:



As an administrator, you want the right to open as many parallel sessions as you choose:



Even if you are included in the “HTML View User” profile, the settings configured in your own user account override the value set for the profile here, as well as the default (“global”) value.

If a user tries to start more sessions than they are allowed, an error message is displayed.

If the maximum number of sessions for a user is “0”, this user will not be able to start NetMan in a terminal server session.



### 3. Starting NetMan in the Terminal Server Environment

You can start NetMan during a terminal server session as you would any other program, through a desktop link, through the “Start” menu, or directly from the Explorer. But these options should be reserved for system administrators.

In order to retain the full benefits of the desktop control features NetMan offers, you need to provide your users with a NetMan interface, or “shell”.

#### ***NetMan as Shell***

There are two ways to start NetMan as a shell; both methods provide equally good network security.

- Define the NetMan Client as the start program in the user configuration of the terminal server software.
- If you are using the Citrix Server software, you can publish NetMan as an application.

In both cases, the NetMan desktop replaces the “normal” Windows desktop (based on explorer.exe) as the standard desktop.

#### ***Launching NetMan Configurations Without a NetMan Shell***

For some users, you may wish to present just a few applications rather than the full NetMan desktop. At the same time, you do not want to give up the functions enabled by the NetMan actions that you configured in the corresponding NetMan configurations. In other words, you want a user to launch a NetMan configuration without starting the NetMan Client. All you have to do is call the NetMan command line program (nmcmd32.exe) rather than the NetMan Client (nm32.exe) and enter the NetMan configuration ID as a command line argument:

```
NMCmd32.exe /ID:<Configuration ID>
```

For example, the command line call to launch the NetMan configuration for the “LexiRom” application is:

```
NMCmd32.exe /ID:<LexiRom>
```

With this command, NetMan will execute all of the NetMan actions defined for the LexiRom application call, but the user is not presented with a NetMan interface. All NetMan functions and properties (licensing, runtime recording, etc.) defined for the application are active.

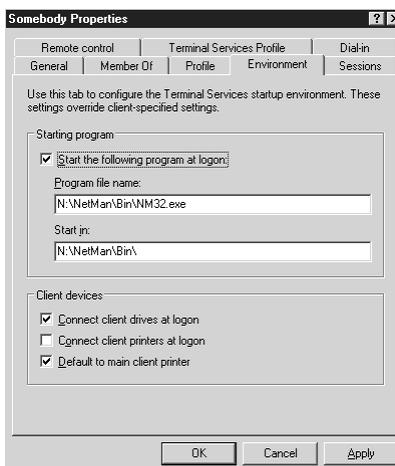


## Using Citrix ICA Files to Start NetMan

Another way to make NetMan available to your LAN or intranet users is to define a program start for the NetMan Client or NetMan command line program in ICA files and then making these files available for user access. Here are four different examples using this method. The first two start the NetMan Client; the second two launch individual NetMan configurations:

**Example 1:** The NetMan Client is defined in the Citrix user database as the start program for a given user.

In the dialog shown below, this user is called “NetManUser”:

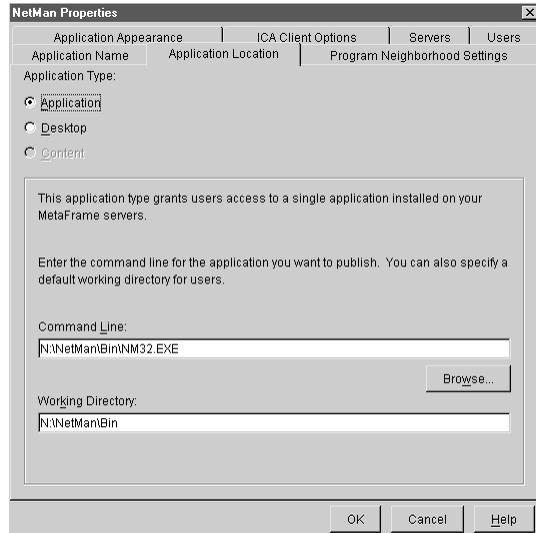


The ICA start file for this user contains the following:

```
[ApplicationServers]
NetMan=
[WFClient]
Version=2
TcpBrowserAddress=192.111.12.1
[NetMan]
WinStationDriver=ICA 3.0
TransportDriver=TCP/IP
Address=192.111.12.1
ClientAudio=On
ScreenPercent=90
Username=NetManUser
Domain=MyDomain
Password=000694vf4eax55a7
```

**Example 2:** The NetMan Client is the shell, defined under Citrix MetaFrame as an anonymous published application.

To start the NetMan Client in a published application (in this case, in “NetMan”) preceded by the Terminal Server Access Control program...



...the ICA file contains the following:

```
[ApplicationServers]
NetMan=
[WFCClient]
Version=2
TcpBrowserAddress=192.168.12.1
[NetMan]
WinStationDriver=ICA 3.0
TransportDriver=TCP/IP
Address=NetMan
TransportDriver=TCP/IP
ClientAudio=On
ScreenPercent=90
InitialProgram=#NetMan
```

**Example 3:** The NetMan “LexiRom” configuration is defined in the Citrix user database as the start program for the “LexiromUser” account.

The configuration is similar to that in Example 1, except that in this case the command line entered under “Initial Program” is:

```
N:\netman\bin\Nmcmd32.exe /id:lexirom
```

The ICA file contains the following:

```
[ApplicationServers]
Lexirom=
[WFClient]
Version=2
TcpBrowserAddress=192.111.12.1
[Lexirom]
WinStationDriver=ICA 3.0
TransportDriver=TCP/IP
Address=192.168.12.1
ClientAudio=On
ScreenPercent=90
Username=lexiromuser
Domain=MyDomain
Password=000694vxxeax55a7
```

**Example 4:** The NetMan “LexiRom” configuration is defined under Citrix MetaFrame” as an anonymously published application.

This configuration is similar to that in Example 2, except that in this case the command line for the published “LexiRom” application is:

```
N:\netman\bin\Nmcmd32.exe /id:lexirom
```

The ICA file contains the following:

```
[ApplicationServers]
Lexirom=
[WFClient]
Version=2
TcpBrowserAddress=192.168.12.1
[Lexirom]
WinStationDriver=ICA 3.0
Address=lexirom
TransportDriver=TCP/IP
ClientAudio=On
ScreenPercent=90
InitialProgram=#lexirom
```

The most commonly followed procedure is to distribute ICA files, the Citrix ICA Web Client and presentations of the available applications through HTML pages. This procedure is described in detail in the following pages.

## Starting NetMan from HTML Pages with Citrix Web Clients

With the ICA Web Client from Citrix you can use a variety of interfaces to access a terminal server from a browser. You can start NetMan on a terminal server using any of these interfaces. In this section, we describe the configuration options you can use with these interfaces.

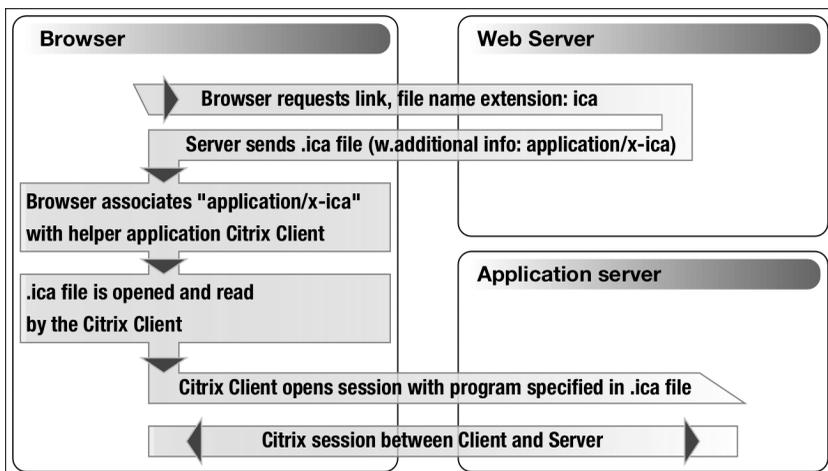
- Application Launching
- Embedded ActiveX Controls (MS Internet Explorer)
- Embedded Netscape Plug-ins
- Browser Detection (JavaScript with auto-start)
- Browser Detection (JavaScript without auto-start)
- Embedded Java Applets



### Note

*The NetMan HTML View module generates these ICA files automatically. The structure of these ICA files can be made dependent on the client browser, host name, or IP address, and the files have an expiry date. All of these features help to improve security on your MetaFrame server.*

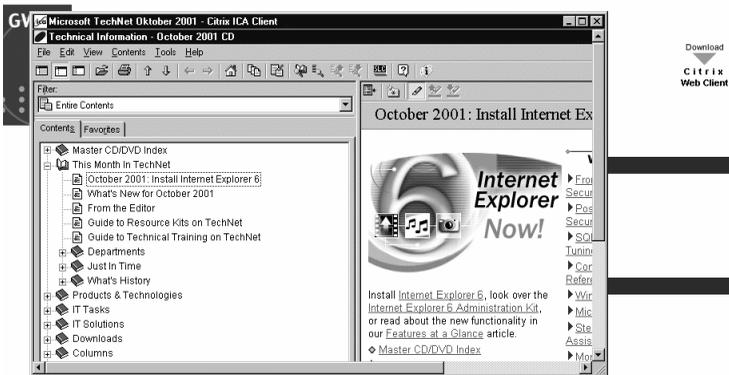
The diagram below shows how ICA files are used to start terminal server sessions:



If any of your users has a Netscape browser, make certain that the "ICA" Mime type is registered in the Web server. The way to configure this may vary from one server to the next; refer to the server manual for instructions.

## Application Launching

This interface presents the simplest method for integrating a terminal server in the Web. The user clicks on a link to start an application; the link, however, does not point to the application itself but to an ICA file, which the Web server then sends to the client's browser. The browser associates the file's MIME type with a helper application called "ICA Web Client", which opens the file. The ICA Web Client opens a terminal server session and requests the application indicated in the InitialProgram line of the ICA file that it received from the Web browser. The user can see the ICA Web Client running in a separate window. (Alternatively, you can select the "Seamless" mode to suppress the ICA window, to create the appearance that the application is running on the local desktop.) The browser passes the focus to the ICA Web Client and runs in the background while the application is active.



Microsoft TechNet October 2001  
International Supplemental October 2001

## Disadvantages

- With this method, is it too easy for users to start multiple server sessions, even unintentionally. A user who wants to open another application while the first one is running might simply minimize the active window and select another application from an HTML page. This starts another terminal server session, which means the user takes up multiple server software licenses, none of which is released until either the user exits an application or a session is automatically closed by a timeout function.
- The terminal server is not "truly" integrated in the Web, because the browser is running in the background.
- The user has to click on the browser's "Back" button to return to the HTML page from which the application was launched.

## Advantages

- This form of Web integration can be used with any Windows Web browser that integrates helper applications. All browsers that we know of have this functionality. In other words, you can configure the browser (usually under Edit/Preferences) to start a specified helper application when a certain type of file is received from a Web server.
- Can be used with both 16-bit and 32-bit Windows operating systems.

## ***Embedded ActiveX Controls (MS Internet Explorer)***

The ActiveX controls provided by Citrix allow a particularly elegant form of Web integration. With ActiveX controls you can embed a link to a terminal server in an HTML page—hence the term “embedded ActiveX control”. The functionality of ActiveX controls in this context is similar to that of the “Application Launching” method described in the previous section—it even uses the same executable file as a helper application. The difference is that the ActiveX control, in this case an OCX control (*WFICA.OCX*), embeds the ICA Web Client in the HTML page. When Microsoft Internet Explorer opens an HTML page that has a reference to this ActiveX control, the OCX control loads the ICA Web Client and embeds it in the HTML page. The ICA file named in the link is opened by the ICA Web Client, which reads the file and creates the connection to the terminal server.

At the time of writing, ActiveX controls are still a proprietary development of the Microsoft company and can be used only with the Microsoft Internet Explorer.

ActiveX controls are programmed to install themselves on the client machine automatically if this is allowed by the browser security settings. The next time this ActiveX control accesses the Web server, it checks whether the server has a more recent version of the control and, if so, asks the user whether the new version should be installed.

## Disadvantages

- Can be used only with the Microsoft Internet Explorer
- The automatic installation of an ActiveX Control includes a large number of modifications to the Registry; this is a potential source of errors
- A minimum screen resolution of 800x600 is recommended
- Can be used only with 32-bit Windows systems, not 16-bit systems
- Security settings in the MS Internet Explorer have to be set low enough to allow the ActiveX control to install itself

## Advantages

- If the minimum screen resolution is available, applications can be presented attractively and elegantly in the Web environment
- The display format does not lead the user to inadvertently start multiple server sessions
- The automatic installation of ActiveX controls makes this method convenient for users.

## ***Embedded Netscape Plug-ins***

The functionality of the Netscape plug-in is similar to that of ActiveX controls. Here, too, the ICA Web Client is invoked (see “Application Launching” above), but in this case it is a special plug-in DLL file that embeds the ICA Web Client in the HTML page. As with the ActiveX control, the function of the plug-in DLL is defined by HTML instructions. Again, the ICA file is opened with the ICA Web Client, which interprets it and creates the connection to the terminal server accordingly.

The setup routine for the Netscape plug-in determines the location of the Netscape directory so that it can copy its plug-in DLL into the browser’s “Plug-in” subdirectory. This could lead to difficulties; for example, if a Netscape browser from another manufacturer is used (such as the “Corel” version), then the plug-in setup routine does not find the Registry entries it is looking for and the plug-in DLL is not copied to the client machine. This type of problem can also occur if there is more than one version of Netscape on the computer.

Plug-in technology is also proprietary; it belongs to Netscape and works only with Netscape Navigator versions 3.0 and later.

Theoretically, the Netscape plug-in could also be used to some extent with the Microsoft Internet Explorer, but we cannot guarantee that it will function correctly. As of version 5.5 SP2, the MS Internet Explorer no longer supports Netscape plug-ins.

### **Disadvantages**

- Can be used only with Netscape Navigator
- Must be set up manually by the user
- Minimum screen resolution of 800x600 recommended

### **Advantages**

- If the minimum screen resolution is available, applications can be presented attractively and elegantly in the Web environment
- The display format does not lead the user to inadvertently start multiple server sessions
- Versions available for both 16-bit and 32-bit Windows systems

## ***Browser Detection (JavaScript with Auto-start)***

The main drawback of the two methods described above is that they rely on proprietary software; in other words, they will not necessarily function in the same way with every type of browser. To use these methods, you must define a different HTML page for each browser type and create a browser-specific structure within your Web server. This is not particularly practical, and entails a great deal of additional administrative work.

If your Web server is accessed by both Microsoft and Netscape browsers, we recom-

mend writing a JavaScript that detects the browser type used by the client. Your HTML page is then generated accordingly. In order for this to work, the client's browser must be configured to allow execution of JavaScript.

Below is an example of a JavaScript. As new browser types are developed, the script required for browser type detection may change; this is simply a sample of one possible script.

### Script

Code meaning

```
<script language="JavaScript">
```

HTML tag that defines the beginning of the JavaScript.

```
var icaFile = "netman.ica";
```

Variable that specifies the ICA file to be opened.

```
var width = "801";
```

Variable for horizontal resolution

```
var height = "601";
```

Variable for vertical resolution

```
var start = "Auto";
```

Variable that defines whether the terminal server is accessed automatically ("Auto") or only when the users clicks in the Plug-in or ActiveX window.

```
var border = "On";
```

Variable that defines whether there is a frame around the Plug-in/ActiveX control

```
var hspace = "0";
```

Variable for width

```
var vspace = "0";
```

Variable for height

```
var cabLoc = "wfica.cab";
```

Variable for the archive file that contains the ActiveX control (valid only for ActiveX access)



```
var activeXHTML = '<OBJECT classid = "clsid:238f6f83-b8b4-11cf-8771-00a024541ee3" data="" + icaFile + ` ` CODEBASE="" + cabLoc + ` ` width=' + width + ` ` height=' + height + ` ` hspace=' + hspace + ` ` vspace=' + vspace + ` ` <param name="Start" value="" + start + ` `><param name="Border" value="" + border + ` `></OBJECT>';
```

String variable containing the HTML code for ActiveX control access. The variables in this string are set by the values defined above.

```
var plugInHTML = '<EMBED SRC="" + icaFile + ` ` width=' + width + ` ` height=' + height + ` ` start=' + start + ` ` border=' + border + ` ` hspace=' + hspace + ` ` vspace=' + vspace + ` `>';
```

String variable containing the HTML code for Netscape Plug-in access. The variables in this string are set by the values defined above.

```
var userAgent = navigator.userAgent;
```

Variable containing the string that enables this script to establish the client's browser type.

```
if (userAgent.indexOf("Mozilla")!=-1) { if
(userAgent.indexOf("MSIE")!=-1) { if
(userAgent.indexOf("Windows 3") > 0) {
document.write(plugInHTML); } else {
document.write(activeXHTML); } } else { if
(userAgent.indexOf("Win16") > 0) {
document.write(plugInHTML); } else {
document.write(plugInHTML); } } }
```

This is where the type of browser is actually detected and the corresponding HTML tag sent to the client. This process is explained in detail below.

```
</script>
```

HTML tag that defines the end of the JavaScript.

The central function of this script is the detection of the user's browser type. To do this, the JavaScript object called "navigator.userAgent" compares the browser's "userAgent" string to known userAgent strings.





## ***Browser Detection (JavaScript without Auto-start)***

This is similar to the procedure described above; the only difference is that the connection to the terminal server is not created until the user explicitly clicks in the ActiveX or Plug-in window. This is defined by the value for the “var start=” variable in the JavaScript shown above, which can be set to “Manual” or “Autostart”.

## ***Embedded Java Applets***

With an ICA Java Applet, you can provide client access to a terminal server from any Java-capable browser running on any operating system. The applet is stored on the Web server and loaded by the browser over HTTP when the client activates your NetMan link.

The distinctive feature of Java applets is the fact that they are platform-independent, which means they can run on any client machine, regardless of the type of operating system on the client.

Unfortunately, Java security specifications are very restrictive. For example, a Java applet cannot create a network connection to any computer other than the host from which it originated. This may mean, for example, that you have to store the Java applet on a Web server that runs on the same computer as your terminal server to enable the applet to create an ICA network connection between the browser and the terminal server. This situation is inconvenient at best, which is what led to the introduction of “signed” Java applets.

A signed applet comes in an archive that contains all Java classes. In the signed applets that create ICA connections, the signature certifies that the applet comes from the Citrix company. When you call this archive, Java security specifications are expanded to allow the applet to create a connection, following user approval, to a computer other than one on which it originated. The format for signed applets, however, has not been standardized. The Microsoft Internet Explorer, version 3.02 or later, uses “.cab” archives; the signed applet in question is stored in an archive called JICAEngM.cab.

In order for the Internet Explorer to accept and execute the file in this archive, the security settings in the Internet Explorer must be set to “Medium”. The Netscape Navigator version 4.0 or later uses “.jar” archives; the signed applet for use with this browser is in an archive called JICAEngN.jar. From version 4.0 onward, the MS Internet Explorer will also be able to work with JAR archives. Again, the HTML definitions are different for each type of browser.

As you see, there are no hard and fast rules for the configuration and integration of Java applets, and the rules that do exist can change relatively quickly. Moreover, the Java applet is integrated in the HTML page by a JavaScript that detects the client’s browser type and either generates HTML code accordingly or sends the corresponding type of Java archive to the client browser; this script may have to be modified as the environment changes.

### Disadvantages

- Problems with the restrictive security specifications in the Java environment can necessitate the use of signed applets, which are not uniformly supported in the different types of browser.
- Java security settings can make it difficult or even impossible to access local resources.
- Because the applet is located on the Web server and re-loaded every time the HTML page is opened (the applet is approximately 150 KB in size), this method is not recommended for systems that have little bandwidth available.

### Advantages

- Java enables platform-independent access to terminal servers from a Web browser.
- Because the applet is located on the Web server and re-loaded by the browser every time the HTML page is opened, there are no files to be updated on the client machine.



## 4. Special NetMan Features for Terminal and MetaFrame Servers

The NetMan Terminal Server module has two features for expanded MS Terminal Server functionality:

- The Access control program lets you restrict terminal server access to individual IP-addresses, address ranges or host names. You can also bind NetMan user names to IP addresses, address ranges or host names for simplified administration.
- NetMan incorporates a Windows dialog you can use to switch to the “Installation” mode, rather than calling *Change.exe* from the command line.

This module also has the following features for expanded Citrix MetaFrame functionality:

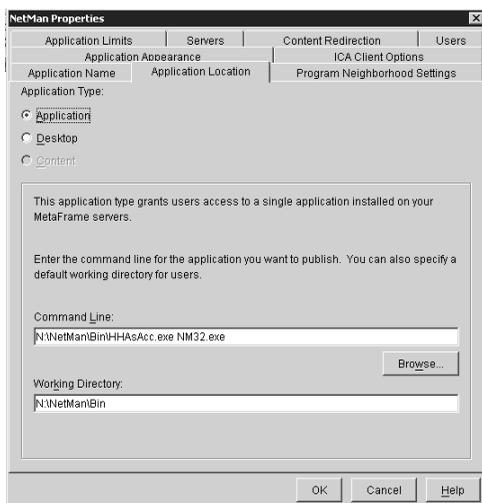
- Virtually published applications are an extension of the Citrix concept of published applications and are designed to reduce the administrative work involved in defining and maintaining large numbers of applications.
- Secure distribution of ICA files increases security in your Citrix server by allowing access only through ICA files that are downloaded at the time of access.

These program extensions are available only in conjunction with the NetMan HTML View module, and are described in detail in the manual for that module.

### Terminal Server Access Control

NetMan’s Terminal Server Access Control program lets you restrict access to your terminal server.

To activate the access control feature, simply enter “HHAASAcc.exe” (= H+H Application Server Access) in the command line that calls NetMan. This applies whether you are calling the NetMan Client or using the NetMan command line program to call an individual NetMan configuration. In the dialog box below, the access control program is added to the command line for the anonymously published “NetMan” application:

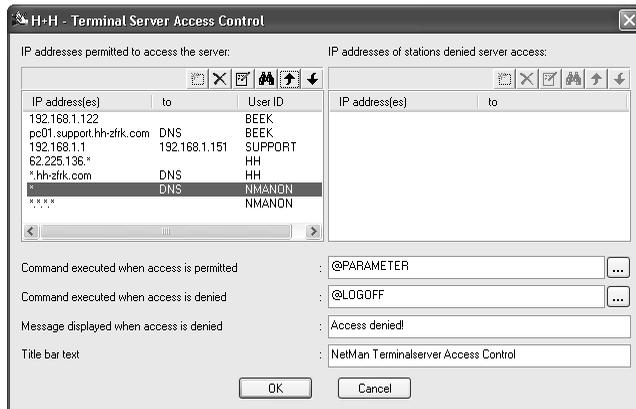


## Program Function

The *HHASAcc* program detects the client's IP address and compares it to

- the list of permitted host names, addresses and address ranges
- the list of excluded host names, addresses and address ranges

These lists, as well as the programs to be called and the message shown when access is denied, are configured in the "Terminal Server Access" configuration in the NetMan System Administration folder:



The order in which addresses and address ranges are listed is important, because the lists are scanned from top to bottom and the first match found is applied.

In the dialog shown above, for example, if the sixth or seventh entry was at the top, all clients stations would be assigned the user name "NMANon".

You can configure the access server control to run one of two programs when a client accesses your terminal server, depending on whether access is permitted (for example, *NM32.exe*) or denied (for example, *Logoff.exe*). You can also define a text to be displayed when access is denied.

## Notes on the Interaction Between *HHASAcc* and the NetMan Client

### Binding NetMan User Names to IP Addresses

You can assign a "typified" NetMan user name to each entry in the lists of permitted address. A typified user name can apply to:

- an individual address or host name (the first and second entries in the list shown above)
- a range of IP addresses (third entry) or a host name with a wildcard ("\*\*")

(fourth entry)

- a segment of an IP address, to define a subnet (e.g., "192.168.1.\*")
- several or all subnets (fifth and sixth entries)

### Example:

When the NetMan Client is called with terminal server access control, using the command

```
HHASAcc.exe NM32.exe
```

by a station with the host name MyComputer.support.hh-zfrk, the client is recognized as a permitted user and the value for the *@IPUSER* variable is set to «SUPPORT», based on the fourth entry in the list above (\*.support.hh-zfrk.com).

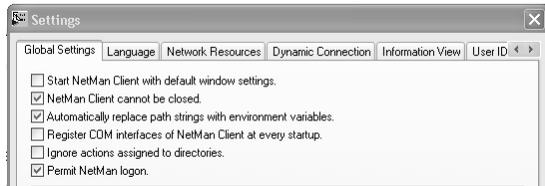
Thus the COMMAND EXECUTED WHEN ACCESS IS PERMITTED is as follows:

```
NM32.exe /logon:Support
```



### Note

*Any user name you bind to a host name, IP address, address range or subnet must already exist in the NetMan database or must be created there manually before it can be effectively assigned in the NetMan terminal server configuration. NetMan does not enter typified user names in its user database automatically. In the NetMan Settings, you need to either assign permission to log on with the 'Logon Parameter', or set the NMAAllowLogon variable to «1» for the Anonymous Users group.*



The following parameters are available for use in the command line fields:

<i>@PARAMETER</i>	Command line argument passed to the HHASAcc.exe program
<i>@ADDRESS</i>	Client IP address
<i>@CLIENTNAME</i>	Name of the client machine
<i>@IPUSER</i>	User name bound to the IP address



@STATIONNAME

Session ID

@USERNAME

Login name of the terminal server user



#### Note

*Not all of these variables are recommended for use with NetMan, but they may be of interest when working with other programs.*

#### Arguments Passed to HHASAcc.exe

For NetMan operation in the terminal server environment, we recommend entering @PARAMETER in the COMMAND EXECUTED WHEN ACCESS IS PERMITTED field. This passes the argument that was entered in the command line after HHASAcc.exe. For example, if your command line was:

```
HHASAcc.exe calc.exe
```

then the calculator program starts. If the command line was:

```
HHASAcc.exe nmcmd32.exe /id:calculator
```

then the NetMan configuration for the calculator program is launched without starting the NetMan client.

#### Examples

A user calls the “LexiRom” program from an HTML page. The NetMan configuration ID for this application call is “LEXIROM”. The command line call for this is:

```
HHASAcc.exe nmcmd32.exe /id:lexirom
```

(See the first section of Chapter 5 in the *NetMan Base Module* manual.)

Or, if you enter the following in the COMMAND EXECUTED WHEN ACCESS IS PERMITTED field:

```
@PARAMETER /logon:@IPUSER
```

Then the command line is:

```
nmcmd32.exe /id:lexirom /logon:<typified_user_name>
```



#### Tip

*Some of the commands used to access a terminal server from an HTML page may involve*

a large number of ICA files. You can simplify the configuration of these links by using NetMan HTML Wizard or the NetMan HTML View to generate them.

### **Access Control Functions in the Trace Monitor; Testing**

You can test the function of the access control features by starting the NetMan Trace Monitor before starting the access control program. Set the Trace Monitor output level to ALL MESSAGES to view all output from the Terminal Server Access Control program.

To simulate the functions for station access and binding to NetMan user names, enter an IP address in the *HHASAcc.cfg* configuration file stored in %NMHome%\Bin.

Example: NetMan is anonymously published under Citrix. You want to permit access by computers in a given Internet domain under a special NetMan user name. For this example, we'll use the well-known "Google.com" domain. To obtain a valid IP address in this domain, enter

```
NSLookup www.google.com
```

on the NT command line. To make your workstation computer (which in this example has the IP address "192.168.10.1") simulate the IP address obtained, enter the following lines in the *HHASAcc.cfg* file:

```
[ReplaceIP]  
192.168.10.1=216.239.55.100
```

Then start the Trace Monitor and enter the command

```
HHASAcc.exe NM32.exe
```

The Trace Monitor shows the following output:



#### **Note**

*Note that hostnames are resolved from IP addresses. If the DNS configuration of a given computer does not allow "reverse NSLookup", NetMan is unable to determine the host name and access is denied.*

### **Changing the Operating Mode: HHASMode.exe**

Most application setup programs make changes in the Windows directory of the workstation they are executed on; for example, they may create or modify DLL files, configuration files, INI files, or Registry entries. Depending on your configurations, clients in a terminal server environment have their own Windows directories. If an application does not find the INI entries it needs, it copies the INI file (if available) from the central Windows directory to the user-specific Windows directory (=INI file mapping



is ON). A similar mechanism provides the user with the required Registry entries. To ensure that these components are available to all users from the time the application is installed, it is important to switch the terminal server operating system to the “Install” mode (=INI file mapping is OFF) when installing an application. This way, the required information is provided by the operating system on a multi-user NT machine, and not only to the administrator account used to install the application.

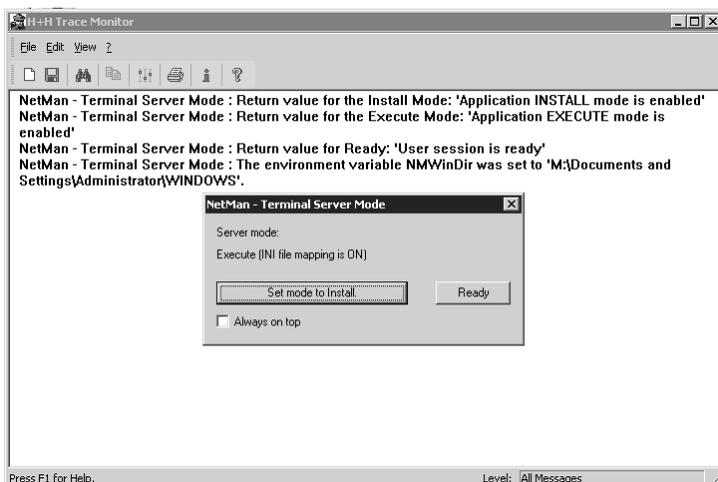


## Tip

*When installing applications in “installation mode”, keep in mind that the current Windows directory during installation is the system Windows directory on the terminal server, and the application installation might impair server functioning. To avoid serious problems, we strongly recommend testing the setup beforehand on another workstation and using the NetMan Installer to monitor the components installed.*

You can use a NetMan helper program, located in the `%NMHome%\Bin` directory, to toggle between the “Install” and “Execute” modes. This is equivalent to calling the “Change” program from the command line:

```
Change user /<install | execute>
```



In the dialog box shown above, the current mode is “Install”. In the “Execute” mode, the Windows directory is in the “Profiles” subdirectory of the current user. When you use *HHASMode.exe* to change to the “Install” mode, the NetMan *NMWinDir* variable is set to `%SystemRoot%`. NetMan system programs only function properly when the

NetMan Windows directory variable is set correctly. For example, this value enables the NetMan Installer Module to monitor the central Windows directories when applications are installed. This is why the NetMan Installer Wizard prompts you to change to the “Install” mode (if this is not the current mode) when it is started in a terminal server environment, and always updates the *NMWinDir* variable.

**Tip**

*The NetMan Installer module can be a tremendous help to you in a terminal server environment. When you install applications, you are not informed of the changes made on the server during setup; if you had to re-install the server at some stage, you would have to re-install each of your applications as well. Scripts created by the NetMan Installer can save you this step. The same applies to load-balancing server farms; the NetMan Installer can save you having to install each application individually on each and every server.*



## 5. Notes on Troubleshooting in the Terminal Server Environment

Terminal server environments are generally characterized by restricted user privileges, which protects server stability. In many cases, users are allowed only to start applications, while access to other system resources (i.e. the Explorer, with the Windows desktop and Start button) is denied. With NetMan this is also the rule in terminal server environments. When problems occur, they can be difficult or impossible to trace, since no resources are available outside the application started. In the following we offer some tips on how NetMan functions can be used in troubleshooting.

### *Problems Starting NetMan*

As administrator, you can start the NetMan Trace Monitor to view the internal processes that run when NetMan is started. If an end user has problems starting NetMan, you can position the Trace Monitor call to precede the NetMan start command. You can do this either in the definition of the published application, or in the definition of the start program in user administration. You can add the following arguments when starting the Trace Monitor:

```
HHTrace.exe [/c:<Program>] [/l:<Output Level>]
```

For <Program>, you can enter one of the following:

- NM32.exe (NetMan Client)
- NMCmd32.exe (command line program)
- HHASAcc.exe (terminal server access control)
- NMHTTP.exe (virtually published application called from NetMan HTML View)

For <Output Level>, you can enter one of the following:

- 1 (error messages only)
- 2 (trace messages (this level is usually sufficient))
- 6 (all messages)

Examples:

```
N:\NetMan\Bin\HHTrace.exe /c:NMHTTP.exe
```

...starts the Trace Monitor (with the default output level), then the virtually published application

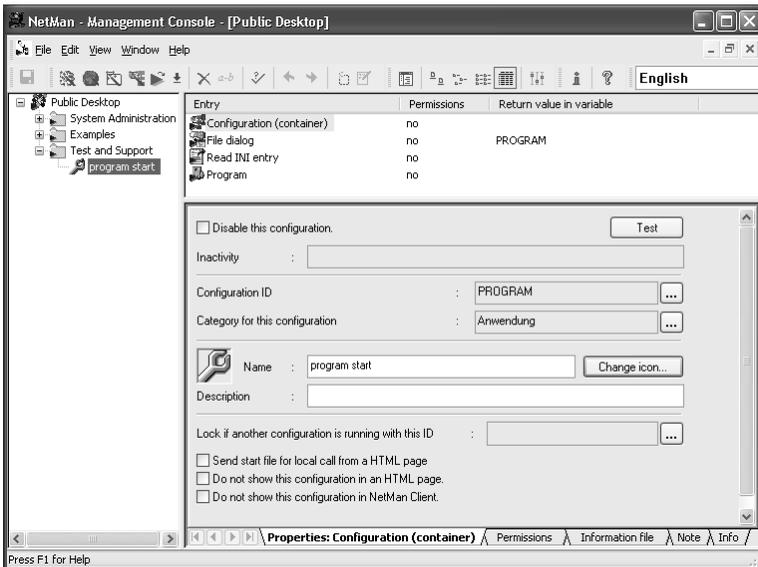
```
N:\NetMan\Bin\HHTrace.exe /c:"HHASAcc.exe NM32.exe" /l:6
```

...starts the Trace Monitor, then the Terminal Server Access Control Program, followed by the NetMan Client, and sets the Trace Monitor output level to 6, "All messages".

## Problems with an Application

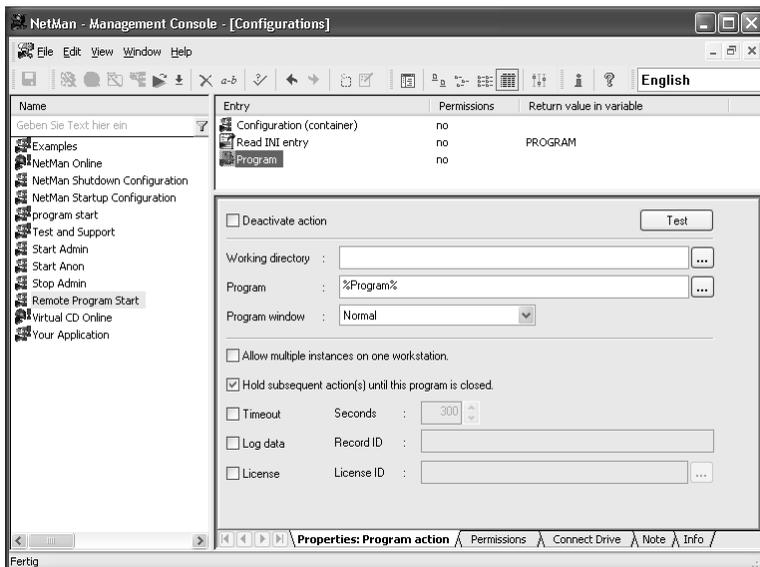
Problems that occur when an application is started by user might not be reproducible when you log on as administrator. As a user, however, the system resources needed for troubleshooting are not available. To get around this limitation, you can use a “Remote Configuration” action to start other programs.

The example below shows how an administrator can start helper programs in the reduced environment of a user.



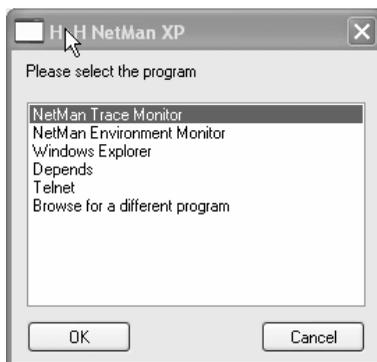
The program is selected from a file dialog. The program name is written to an INI file. Then a Remote Configuration is started for a defined user or station profile, and a station or user designated for this purpose (e.g., “Testuser” or “Testmachine”) is assigned this profile.

The Remote Configuration is question is designed so that the program to be started is read from the INI file and started by a Program action:



In conjunction with the “mirroring” function in terminal server environments, the procedure described above offers a powerful support tool, as you can call as many helper programs as you like in the user environment mirrored. This mechanism can be refined as well; for example:

- Add a command line argument or a “File list” action to specify which station starts the helper programs. You can store the selection in the INI file. In the Remote Configuration definition, you can read this value and have the function cancelled for all profile members that have a different “Station” value.
- Create a list of frequently used helper programs, and select from this list rather than a file dialog.





**H+H NetMan® XP Language Module**





## Table of Contents

---

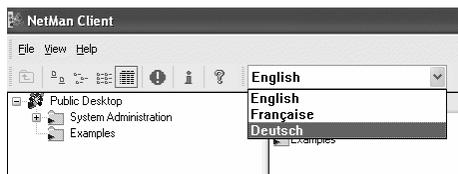
1. Introduction .....	1
Contents of This Manual .....	1
Performance Features .....	1
Languages Available .....	3
2. Creating NetMan Configurations in Multiple Languages .....	5
3. Defining the Default NetMan Client Language .....	7
Control Options Made Possible by the NetMan Language Variable .....	8
4. Language Controls in the HTML Framework .....	11



## 1. Introduction

For instructions on installing and activating the NetMan *Language Module*, see Chapter 2 of the NetMan “Base Module” manual.

When the Language Module is installed, the language can be selected in the NetMan Client toolbar:

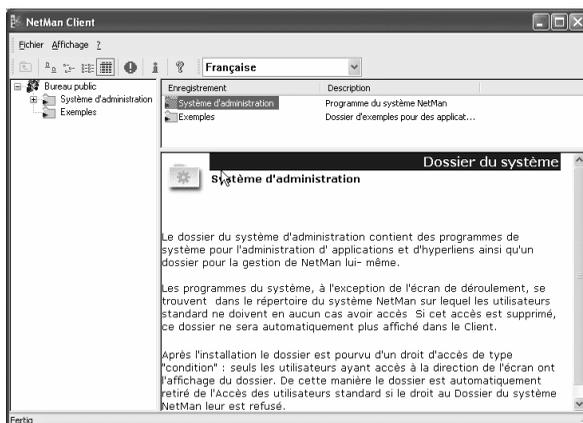


### Contents of This Manual

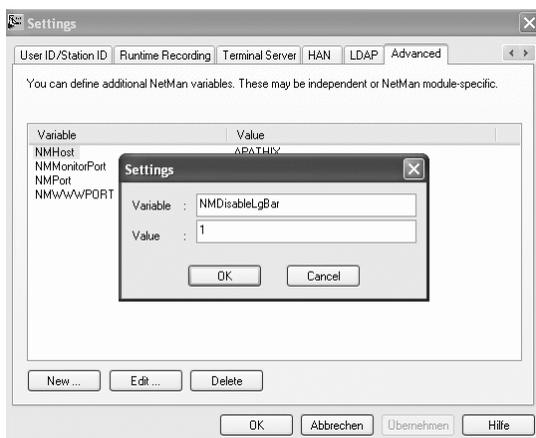
- This chapter, “Introduction”, gives you an overview of the Language Module performance features
- Chapter 2, “Creating NetMan Configurations in Multiple Languages”, describes how to create different language versions of individual NetMan configurations
- Chapter 3, “Defining the Default NetMan Client Language”, shows you how to define a ‘startup’ language for the NetMan Client, including the assignment of different startup languages for specific users or user profiles
- Chapter 4, “Language Controls in the HTML Framework”, illustrates a multilingual implementation in NetMan’s HTML context.

### Performance Features

The NetMan Language Module makes your NetMan Client a multi-lingual user interface. This means you can switch languages while the NetMan Client is running:



You can also hide the toolbar selection list, to disable this option in the NetMan Client. To do this, set the value in the 'NMDisableLgBar' variable to 1.

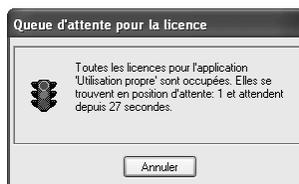


For the administrative interface, you can choose between English and German. This is selected during setup, and can be changed later in the NetMan Settings. Additionally, the Language Module lets you offer different languages for your users as described below.

When you register your NetMan Language Module, the following areas are available in a choice of languages:

- NetMan Client, including the on-line Help
- NetMan system messages to clients
- NetMan Explorer

The “License Queue” message below is an example of a system message in the NetMan Client:



With the Language Module, you can present your NetMan configurations and their HTML-based information files in the language of your choice.

The NetMan system places no limit on the number of languages that can be installed. Additional languages can be obtained only from your software dealer.

You can add the Language Module at any time to make your NetMan system multi-lingual.

### ***Languages Available***

At the time of this printing, the additional languages available are *German, French and Dutch*. Contact H+H for the latest information concerning the availability of other languages, or send an e-mail to the 'Support' address mentioned in the Foreword to let us know what other language(s) you would be interested in.

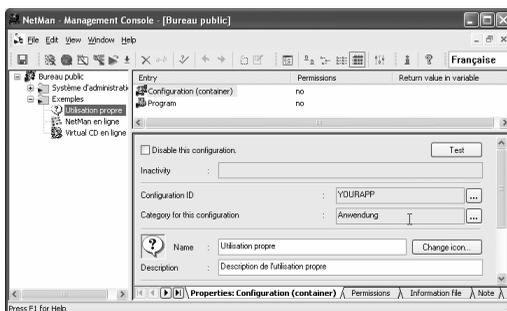


## 2. Creating NetMan Configurations in Multiple Languages

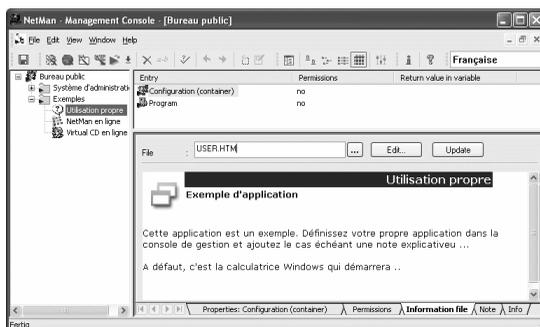
When you switch the NetMan Client from one language to another, you can edit the following elements to match the current language setting:

- the name of configuration
- the description of configuration
- the information file assigned to the configuration
- any texts you might have configured in configuration actions (such as dialog box messages)

Once NetMan and the Language Module have been installed, you can switch the language and edit the language-sensitive elements (name and description) in “Your Application” (“Utilisation propre” in French) as shown here:



Click on the INFORMATION FILE tab to edit the French information file:



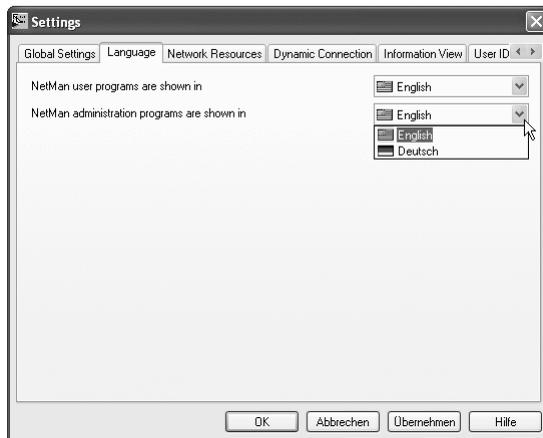
### Note

*If you do not edit the language-sensitive elements in other languages, NetMan shows the base language (in this case, English) texts for these elements when the user changes the NetMan Client language setting. An exception to this rule is the information file, which is not shown at all if it does not exist in the selected language.*

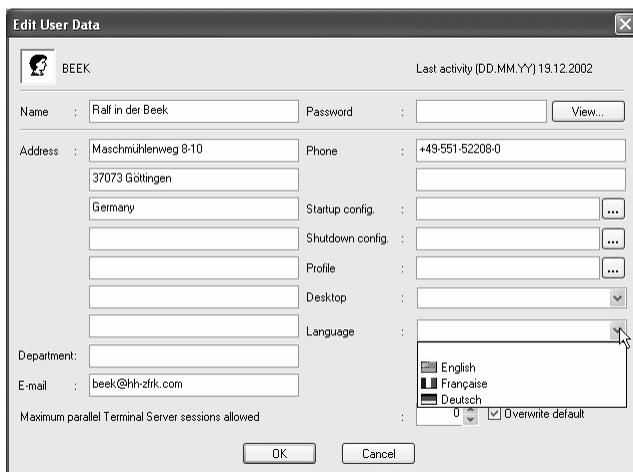


### 3. Defining the Default NetMan Client Language

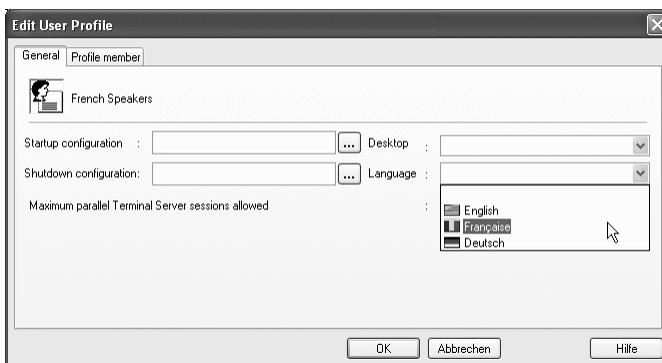
To define the startup language for the NetMan Client, select **SYSTEM ADMINISTRATION / CONFIGURATION / UTILITIES / NETMAN SETTINGS** to start the NetMan Settings program. The default language is defined on the **GLOBAL** dialog page:



This global setting is valid when there is no other setting defined for a given user or user profile. To configure user or user profile preferences, run the **RESOURCES** program in the Management Console. For example, you might define a separate language for a user...



...or perhaps create a user profile called “French Speakers”:



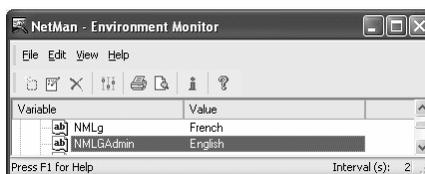
### Control Options Made Possible by the NetMan Language Variable

If you wish to define a language for a specific workstation, regardless of the user or users who might work there, you will find that there is no option for defining the language when you configure stations or station profiles. This is because in designing NetMan, we have defined language as a ‘user’ property, rather than a ‘workstation’ property.

If you do require a station-specific language setting, however, there is a solution. NetMan’s special features for environment variables make it possible.

The language setting in the NetMan system is stored in the *NMLg* variable. This makes it possible to define both a global default language for the system and different default languages for individual user profiles and users. A setting for a user *profile* takes precedence over the global setting, and for a *user* over the user profile setting.

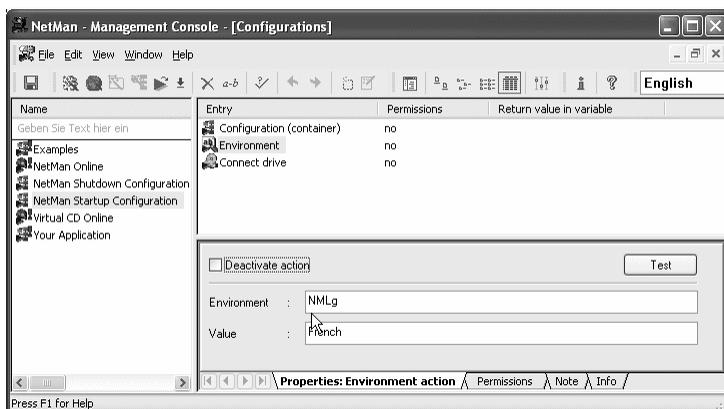
The default language setting when you install NetMan is English; this is stored in the *NMLg* environment variable.



A user can overwrite this setting by selecting another language from the list in the toolbar. The NetMan Client determines its language-dependent information internally from this dynamic variable. This forms the basis of the NetMan Client’s flexibility with regard to language.

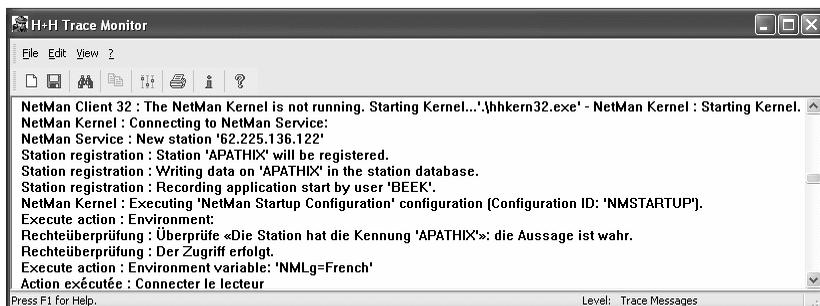
To set the language separately for a given station or stations, first add an Environment

action to the Startup configuration, to set the value for this variable directly in the NetMan environment:



Then you can assign exclusive “execute” rights to this action so that it is only executed for the station in question. Better yet, you can create a separate Startup configuration for the workstation(s) in question so that the Environment action does not have to be processed by every station that uses NetMan.

The result of this action is shown in the Trace Monitor as follows:



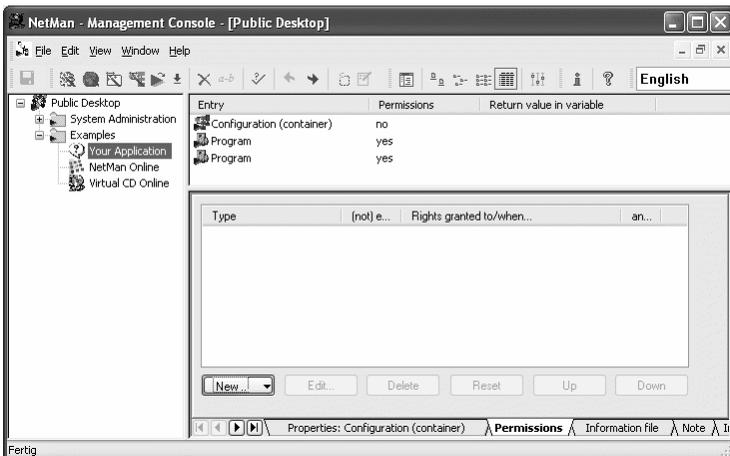
## Note

*This mechanism can be used in many areas of your NetMan system. On the **ADVANCED** dialog page of the NetMan Settings, you can define environment variables that are valid globally and can be modified for profiles or for individual stations or users. In your*

*NetMan configurations, you can add actions that give users the option of changing the content of variables dynamically. You can use variables to define access rights to folders or applications, or to control the processing of NetMan configurations, by assigning “execute” rights that are based on the contents of environment variables. This means you have quite a broad range of possibilities for customizing your system. To return to the example of the Language Module, the `NMLgAvailable` variable lets you define whether the optional languages are available for user selection.*

If you have a multi-lingual application for which the language setting can be defined by a command line argument, INI file entry or Registry entry, you can use the `NMLg` variable to pass the active language setting to the application.

If separate versions of the application are installed for each language, you can add separate Program actions for each version to your NetMan configuration, and define ‘execute’ permissions to each action based on the content of the language variable:



## 4. Language Controls in the HTML Framework

Language control in the framework of HTML consists in opening different HTML documents for different languages. The HTML tag

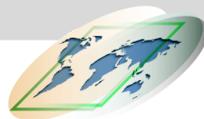
```
<!-- @NM_LANGUAGE = "French" -->
```

in the header of an HTML document tells HTML View to read the data for NetMan configurations and desktops from the French-language databases, and to use the templates in the French-language subdirectory for status messages (such as messages about license availability).

HTML Wizard lets you store NetMan desktops in different documents dependent on the installed Client language. NetMan configurations are inserted in the active Client language.



**H+H NetMan® XP HTML View Module**





# Table of Contents

---

Introduction .....	1
Contents of This Manual .....	1
Performance Features .....	1
HTML View in LANs (without Terminal Servers) .....	2
HTML View in Intranets and the Internet (with Terminal Servers) .....	2
Advantages of Using NetMan HTML View to Generate HTML Pages .....	4
Requirements .....	4
Installing NetMan HTML View .....	7
NetMan and Apache Server on One Computer .....	8
NetMan and Apache Server on Separate Computers .....	9
Initial Testing of Program Viability .....	11
Adapting the Apache Server for HTML View .....	12
Directory Structure in NetMan HTML View .....	13
Authentication Services Directory .....	13
Virtual WWW Directories .....	14
How HTML View Works .....	15
Embedding an Expanded Desktop .....	16
Embedding a Nested Desktop .....	18
Embedding Individual NetMan Configurations .....	20
Selecting the Language .....	21
Embedding HTML Code .....	22
The First Steps .....	23
Troubleshooting .....	33
Restarting HTML View .....	33
Errors .....	33
Apache HTTP Server .....	33
Client Browser .....	33
Terminal Server Access .....	34
HTML View Settings: .....	34
Diagnostic Tools .....	35
Windows NT Event Viewer .....	35
NetMan Trace Monitor .....	35
Log Files .....	35



Designing HTML Pages .....	37
Templates for Generating Desktop Structures .....	37
Templates for Application Starts .....	41
Placeholders in Templates .....	41
Using Style Sheets .....	42
Concluding Remarks .....	43
HTML View Settings .....	45
Directories in HTML View .....	45
Defining URLs in HTML View .....	47
Filter Configuration .....	48
Application Start .....	49
Application Start on a Client Machine (NetMan Start Files) .....	50
Application Start on a Terminal Server (ICA Start Files) .....	52
Virtually Published Applications .....	53
Browser Rules .....	59
Send ICA Launch Link .....	62
Embed Netscape Plug-in .....	63
Embed Java Applet .....	63
Insert Link to Netscape Plug-in .....	64
Insert Link to Java Applet .....	64
Browser Actions .....	64
Operating Systems .....	66
H+H Authentication Services .....	69
'Authentication' Dialog Page .....	70
'IP range / Host name' Dialog Page .....	73
LDAP Interface .....	75
Adapting the Settings in HTML View: Practical Examples .....	77
From 'NMSamples' to Your Own URL .....	77
NetMan HTML Files on a Different Server .....	89

# 1. Introduction

## *Contents of This Manual*

- Chapter 1, “Introduction”, gives you an overview of the HTML View module, including its performance features and system requirements
- Chapter 2, “Installing NetMan HTML View” tells you how to install NetMan HTML View in your system
- Chapter 3, “How HTML View Works”, describes the options available for presenting NetMan configurations in HTML pages
- Chapter 4, “The First Steps”, uses a practical example to show you the minimum configuration necessary for generating HTML pages with HTML View
- Chapter 5, “Troubleshooting”, can help you find the problem if anything goes wrong
- Chapter 6, “Designing HTML Pages” describes the HTML templates provided with NetMan and tells you how to modify them for your own requirements.
- Chapter 7, “HTML View Settings” provides a detailed description of all configuration options available in HTML View. This includes settings for client and browser type, access methods and permissions, file paths, gateway accounts, and the different types of user login. This chapter is a handy reference guide for program configuration.
- Chapter 8, “H+H Authentication Services”, describes how you can use HTML View to provide information to a limited range of users, or IP addresses or host names, and includes a description of the different types on user logon.
- Chapter 9, “Adapting the Settings in HTML View: Practical Examples”, shows you how you can adapt HTML View settings for your particular requirements and preferences.

## *Performance Features*

NetMan HTML View lets you give your users HTML-based access to NetMan container and hyperlink configurations. With this module, a Web browser, rather than Windows-based NetMan Client, is the interface when users access your NetMan applications and hyperlinks.

The integration of NetMan configurations and hyperlinks in HTML pages conforms to your settings in the NetMan Management Console; the same access permissions you configure in NetMan, for example, are effective in HTML

View. Container or hyperlink configurations for which a given user has no permission are not displayed when that user is logged on. HTML

View generates HTML pages dynamically; if you change a NetMan configuration in Management Console, the new settings are effective immediately in the user interface, without your having to edit or create a new HTML page.

When a *hyperlink configuration* is activated, HTML View points the browser to the URL defined for that configuration. When you activate a *container configuration*, on



the other hand, which can contain a Windows application call or a complex sequence of Windows-based commands, one of two processes is started:

- Configuration started on a terminal server with Citrix MetaFrame:  
When the configuration is activated, the application in question starts on a terminal server: a multi-user Windows NT Server running the Citrix MetaFrame software. Terminal servers communicate with clients using, for example, the ICA protocol from Citrix. Access is attained through a Web client, which accesses the terminal server and displays this access on the client screen. Only keystrokes, mouse clicks and screen updates are sent over the network. The tremendous advantage of this technique is that the applications called are not dependent on any configuration of the client machine. In other words, if a particular application runs in a terminal server session started by one network station, you know it will work for any station because the program logic is processed on the terminal server.
- Configuration started on the client machine (e.g., in a LAN):  
With this method, NetMan and the applications it manages run on the client machine. When a user clicks on a link to a NetMan container configuration in an HTML page, the browser starts the application on the local workstation. With this method, the client machine must meet all requirements for operation of the application in the LAN. This may seem obvious, but we point it out here because many people associate a browser interface with the Internet or an intranet, which is often associated with unlimited access. That would come very close to the effect of the method described above, where the only requirement on the local machine is a one-time installation of the ICA components; or when the Java Client is used, no local components are required at all.

If you are reading this introduction to find out whether HTML View would be useful in your network, the brief description above of the main HTML View functions may not be enough for you to base a decision on. For this reason, we address some of the more frequently asked questions concerning this module in the next sections.

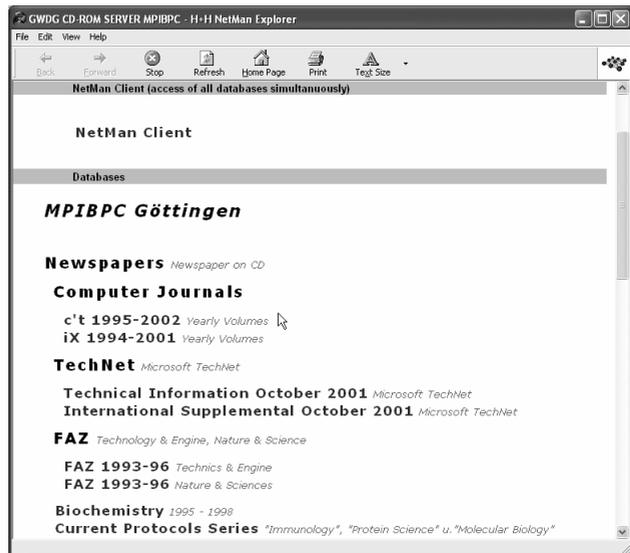
### ***HTML View in LANs (without Terminal Servers)***

In a LAN environment, as opposed to a terminal server environment, HTML View has hardly any technical advantages to offer. Some of our customers use HTML View in their LANs, however, because they generally prefer browser interfaces, and HTML View makes it easy to integrate NetMan in their prevailing desktop format.

### ***HTML View in Intranets and the Internet (with Terminal Servers)***

The *NetMan Client*, as opposed to the NetMan HTML View, can also be called from a browser page (using the ICA WebClient). It can also be called through Microsoft Terminal Services or the Citrix Program Neighborhood. What advantages does the NetMan *HTML View* offer over these methods?

- The information and applications provided through NetMan are listed over HTTP; you do not have to start a terminal server session just to see what is available. Furthermore, if a NetMan configuration is deactivated, or if all licenses for a given application are in use, this information is also provided over HTTP.
- Hyperlinks are executed directly. This means the presentation of hyperlinks and Windows applications is independent of the technical implementation of the corresponding data requests, which can range for example from on-line services to the start of a CD-ROM-based database on a terminal server. By contrast, activating a hyperlink configuration using the NetMan Client in a terminal server session is a complex and resource-intensive operation.
- With HTML View, each NetMan configuration is automatically assigned a URL and can be presented as a link in the Internet. When you create a new NetMan configuration, it is automatically available with all the access restrictions and permissions you have defined, under a URL generated with the following syntax:  
`<webserver>/NetManBin/NMWebClit.dll&tConfigId=<ConfigID>.`  
You can integrate these URLs in your own or higher-level database systems and search engines.
- Although HTML View, unlike the NetMan Client, opens a new terminal server session every time another NetMan configuration is started, in practice this occurs only rarely. Where this is likely to happen, the NetMan Client can be offered parallel to HTML View, and the end user can decide which technique is most suitable.





## Advantages of Using NetMan HTML View to Generate HTML Pages

If you want a browser-based interface to the applications on your terminal server, you can design it yourself. So what advantages does HTML View offer?

- If you use the NetMan system, HTML View is designed and expanded automatically. You do not need to have knowledge of HTML, nor perform additional configuration measures. The NetMan Management Console in the background is your HTML editor, which you can use to design complex Web pages for your intranet and the Internet.
- The display of applications and hyperlinks can be made dependent on user, station, group or profile membership, network group membership, and even IP address and host name. Furthermore, NetMan supports Active Directory Services and LDAP.
- Deactivated configurations and licenses in use are reported directly over HTTP.
- The concept of published applications developed by Citrix is essential for terminal server security. The NetMan HTML View makes the most of this concept by channeling all NetMan applications through a single published application. Thus you do not have to publish your applications individually to attain the same level of security provided by the Citrix method. The more user applications you have, the more work you can save and the more configuration errors you can prevent (such as incorrect or failed publication) using this mechanism.
- Published applications cannot be accessed through ICA files that are created by an end user or started from a browser cache. Access is permitted only through HTML View. (The default setting for expiration of ICA files is 120 seconds.)
- You can send ICA start files that are optimized for various client browsers, platforms or operating systems (e.g., seamless windows for Windows platforms, terminal windows for others).

## Requirements

The following components and conditions are required for working with the NetMan HTML View:

- Windows NT 4.0 Server with Service Pack 3 or later
- Apache HTTP Server v2.0.39 or later
- NetMan installed on an NT Server (not necessarily the same server that HTML View is installed on; see the Chapter “Installing NetMan HTML View”)
- *For access through Novell’s NetWare Directory Services (NDS), a NetWare Client from Novell (not from Microsoft) must be installed on the same NT server that the Apache server and HTML View are installed on*
- To provide access to applications through a terminal server over the ICA protocol, Citrix MetaFrame 1.8 or later must be installed, and the NetMan Terminal Server module must be licensed

- Client stations must have standard Web browsers
- To use the ICA protocol, client stations must have the ICA Web Client

For optimum configuration and use of HTML View, you need a basic understanding of the following:

- Installation and configuration of an Apache HTTP server
- Windows NT administration
- Microsoft Terminal Server and Citrix MetaFrame and their various Web clients
- How a Web browser works
- The HTML templates provided as samples are fully functional. If you wish to create your own templates, you need to have a working knowledge of HTML.



## 2. Installing NetMan HTML View

HTML View is technically an extension for the Apache HTTP server (Windows-based) and is integrated into an existing Apache installation by a separate setup program. This chapter describes the installation of HTML View.

If you selected the “HTML View Module” when installing the NetMan Base Module, the files required for installation of HTML View are copied to

```
%NMHome%\System\HTML-View\Setup
```

during the NetMan setup. See the Chapter “Installing NetMan”, in the “Base Module” manual for a detailed description of this procedure, as well as for instructions on installing and licensing NetMan modules subsequent to your initial NetMan installation.

After performing installation as described in this chapter and then restarting your Apache server, you will be ready to integrate your NetMan configurations as dynamic HTML pages and activate them as hyperlinks, using HTML View.

For your first test, you can use the following URL: `<server>/nmsamples/default.htm`. The `/nmsamples` directory is an Apache alias (virtual directory) automatically entered in `nmview.conf` by HTML View setup, and points to the “...\Example” directory. It contains a number of HTML templates that you can use for testing.

When you are ready to configure your own HTML format templates, just copy this directory and edit the files in the new directory as needed. The procedures are described in detail in the Chapter “Adapting the Settings in HTML View: Practical Examples”. Exactly how you install HTML View depends on where you install it. It must be installed on the same machine on which you run your Apache server, in any case. The question is, whether or not Apache runs on the same machine that your NetMan Base module is installed on. The two installation methods,

NetMan root directory and Apache server installed on one computer, and

NetMan root directory and Apache server installed on separate computers

are described in detail below.

There are a number of factors that may influence your decision between these two options; for example:

Although you might want the improved performance afforded by installing NetMan on a terminal server, you may not want to run the Web server on the same terminal server.

Perhaps your Apache server is already installed on a different computer than the one your NetMan root directory is on.

**Note**

*HTML View and Apache must be installed on the same server. Make sure you call the setup program for HTML View on the console of the same server on which your Apache server is installed.*

### ***NetMan and Apache Server on One Computer***

Following the initial installation of NetMan, the `%NMHome%\System\HTML-View\Setup` directory contains the setup program for installing HTML View Apache server extensions on the machine that runs your Apache server.

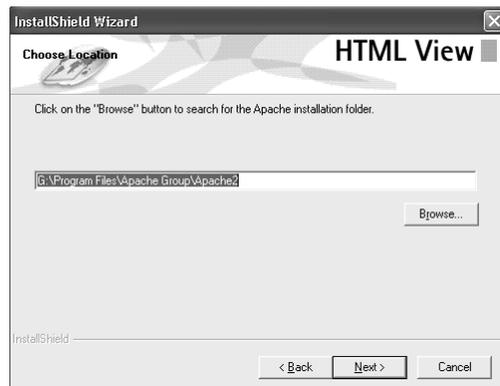
This setup program unpacks the HTML View files, writes the required Registry entries, creates the virtual directories required by HTML Client in the Apache server, and updates system files as necessary. You can start the setup by double-clicking on SETUP.EXE. You might have to shut down all NetMan programs, as well as the Internet Explorer, on the installation machine so that system files can be updated.

When you start the installation program, the first dialog prompts you to select the Setup language.



Select the language in which you want to run the Setup.

After the Welcome window and the licensing agreement, you are prompted to enter the path to the Apache server:



Once the Apache installation (in the required version) is found, the NetMan root directory found by Setup is displayed. In this case, this is a local path. Please check the path and correct it if necessary.



After the Apache and NetMan directories are entered and confirmed, the Setup data is summarized. Confirm the summary to start the installation.



#### Note

*Files are installed in subdirectories of the Apache installation, ..\HH\HTML-View and ..\HH\Common.*

If another program is in use during installation, system DLL files cannot be installed (overwritten). In this case, close down the other program(s) and select **RETRY**.

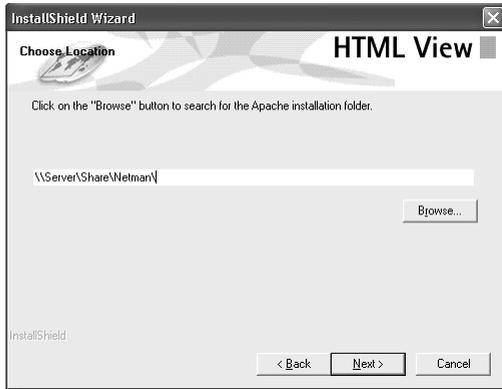
### **NetMan and Apache Server on Separate Computers**

Log in on the console of the server where Apache is installed, and on which you wish to install HTML View.

Call the HTML View Setup over a network connection from the `%NMHome%\System\HTML-View\Setup` directory, if possible using UNC syntax (e.g., `[\\Servername\Share\NetMan\System\HTML-View\Setup]`). The NetMan installation must be shared on the other server for access by the HTML Setup.

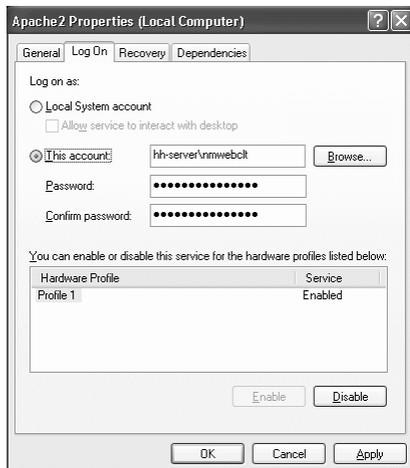
The Setup itself is similar to that described above, under “NetMan and Apache Server on One Computer”.

In this case, the NetMan root directory detected by the HTML View Setup is on a different server. Please check the path and correct it if necessary.



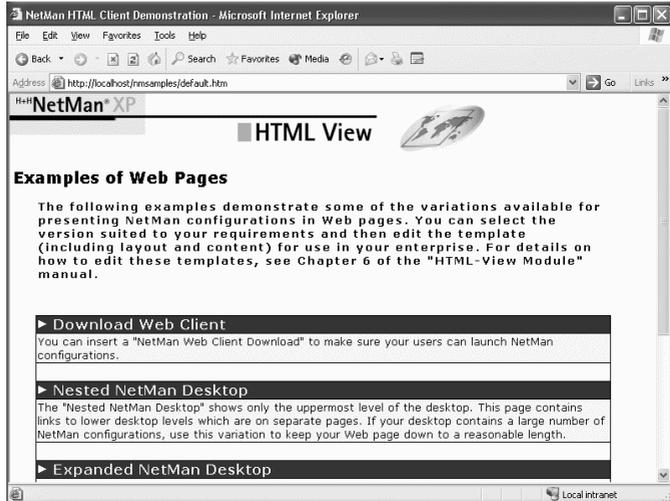
The Apache service usually runs under a system account. Since this system account has no privileges in the network, you need to set up a user account for the Apache service with privileges in the NetMan directory, so that the HTML View program can access the NetMan Base installation. This is a “normal” (non-administrative) NetMan user account, with privileges as described in the Base Module manual, under “NetMan Directories and Network Rights”. Additionally, this account must have ‘write’ privileges in %NMHome%\Prot\NMProt\*.\*, because HTML View adds a record to a log file every time a hyperlink is activated.

Enter this user account in the Apache service properties:

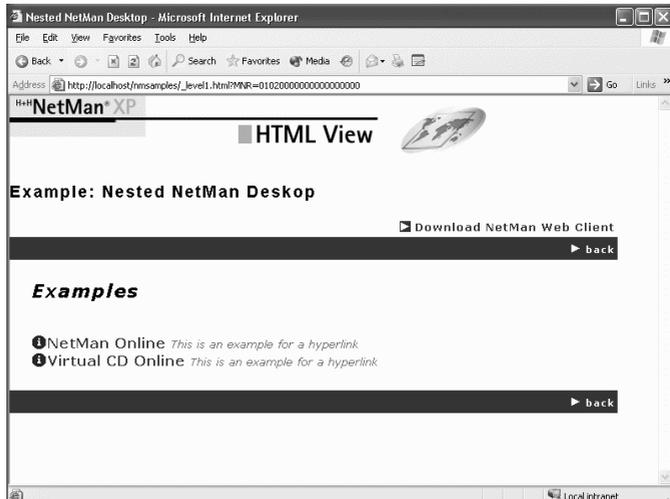


## Initial Testing of Program Viability

After successfully completing the installation and restarting the Apache service, you should be able to open the following sample pages at "http://Servername/nmsamples/default.htm":



...and access the standard NetMan desktop configured here for this purpose:



You can also activate a hyperlink:



The link to a NetMan application, however, has not yet been configured:



That procedure is demonstrated in detail in the Chapter “The First Steps”.

### ***Adapting the Apache Server for HTML View***

Only one minor modification is required in your Apache installation for running NetMan HTML View: The configuration defaults for HTML View (*NMView.conf*) and its authentication module (*HHAuth.conf*) are loaded in the Apache’s *httpd.conf* configuration file. These files are stored in a new folder created in the Apache installation, ../HH.

## Directory Structure in NetMan HTML View

HTML View has several subdirectories in the installation directory (<Apache-Installation>\HH\HTML\_View\):

Directory	Purpose
<code>_download</code>	For downloading the ICA Web Clients that HTML View uses to access your terminal server. A link lets the end user download and install Web clients manually. This directory is where your Web clients, such as the ICA Netscape plug-in or the ICA Java Client, should be stored. The directory contains Citrix Client Setups, modified by H+H, which you can adapt as needed.
<code>_images</code>	Contains the graphics used by HTML View to display its sample HTML pages.
<code>Example</code>	Contains the sample HTML pages.
<code>bin</code>	Contains the executable components of HTML View
<code>default.htf</code>	This is the standard template for presentation of the NetMan desktop and of NetMan container and hyperlink configurations.
<code>myformat.htf</code>	This is a copy of the “default.htf” directory. You can use it to store templates you have modified. This directory is not overwritten when you update the program.
<code>launch</code>	Contains templates used by HTML View to create configuration files for accessing applications.
<code>NetManBin</code>	HTML View uses this directory when generating links
<code>setup.vpa</code>	Contains the setup for a service that is required only if you use virtually published applications.
<code>WithCategories</code>	This is an example template for presentation of the NetMan desktop and of NetMan container and hyperlink configurations with special symbols for NetMan categories.

## Authentication Services Directory

Because the H+H Authentication Services can also be used by other H+H products (such as HAN), this service is installed in a separate directory:

<Apache-Installation>\HH\Common.

## Virtual WWW Directories

The virtual directories required for operating HTML View are defined in `NMView.conf`.

The virtual WWW directory `/nminfo` refers to the directory that contains the information files which you can assign to NetMan configuration links.



### Note

If you have installed HTML View and the NetMan Base Module on two different machines, this directory is not on a local path. The network path to the NetMan installation is entered in UNC syntax.

The `/_download` and `/_images` virtual directories point to the `_download` and `_images` directories in the HTML View directory structure (see “Directory Structure in NetMan HTML View” above).

The `/NetManBin` virtual directory points to `<Apache-Installation>\HH\HTML_View\NetManBin`, which is empty. For details on the purpose of this directory, see Section 0, Defining URLs in HTML View“.

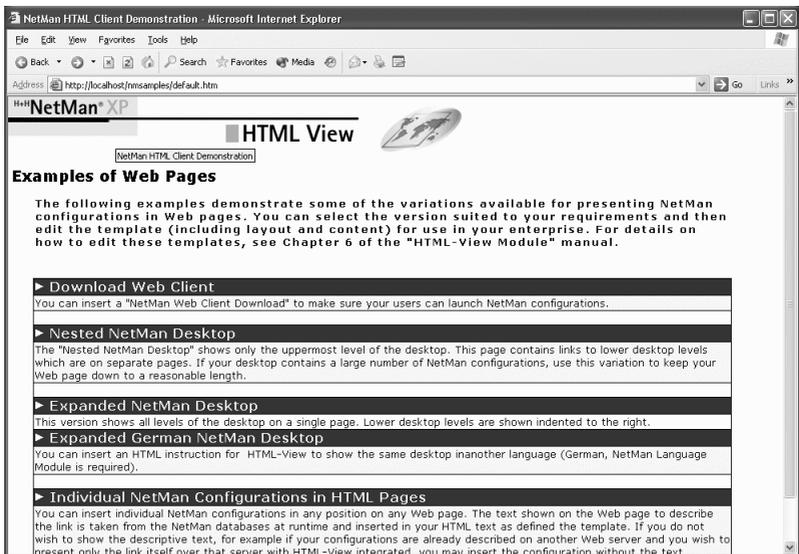
**`/NMSamples` contains the sample pages which you can open in a browser as soon as NetMan HTML View is installed.**

### 3. How HTML View Works

The NetMan HTML View offers you a number of different possibilities for the presentation of applications in an HTML page. The application links seen by the user point to the NetMan configurations that you define in your NetMan Management Console.

The HTML View module comes with a number of fully functional sample HTML pages, stored in the *example* subdirectory. The HTML View Setup creates an Apache alias called */nmexample* that points to this directory (see also “Directory Structure in NetMan HTML View” above).

When you point your browser to `<server>/nmsamples/default.htm`, the following page should be displayed:



These sample files can help you to complete the basic configuration of your HTML View quickly and easily. First, take a look at each of the sample HTML pages, and select the format that comes closest to your own preferences.

To get an idea of how you can configure your own pages with HTML View, it may help to understand how NetMan desktops and configurations are embedded in HTML pages. The procedure is fairly simple:

NetMan HTML View uses HTML comments to embed NetMan configurations and desktops; these comments must be written with a specific syntax. When a browser accesses

a page that has this type of comment in it, the comments are interpreted and, if necessary, modified by HTML View before the page is opened.

Which directories and files HTML View should scan for comments must be defined in the HTML View Settings (see “Filter Configuration” in this manual). Files not so defined are passed to the client’s browser without alteration by HTML View.

The following sections of this chapter describe the options available for embedding NetMan desktops and configurations in HTML pages.

The layout of inserted entries is based on templates in HTML View which are stored under names that correspond to the desktop components (title, subtitle, application, hyperlink) they represent. These templates have placeholders that are filled by texts describing each component (name and description), as entered in the NetMan Management Console, when the page is opened. For details on changing templates, refer to the Chapter “Designing HTML Pages”.

### ***Embedding an Expanded Desktop***

The HTML comment `<!-- @NM_DESKTOP_COMPLETE -->` embeds a complete NetMan desktop in a single HTML page. This is what is meant by the term “expanded desktop”.

Exactly which of your NetMan desktops is embedded is defined elsewhere in the HTML View Settings (see “Directories in HTML View”).

#### ***Example:***

```
<html>
..
<!-- @NM_DESKTOP_COMPLETE -->
..
</html>
```

#### ***Result:***



If a different desktop is assigned to a given user or to the profile that this user belongs to, then that desktop is loaded, rather than the default as shown here, when the user in question logs in.

To load a desktop other than the default desktop in an HTML page, simply include the desired desktop file name in the HTML comment:

```
<html>
..
<!-- @NM_DESKTOP_COMPLETE = "myDesktop.mnu" -->
..
</html>
```

**Note**

*If a different desktop is assigned to a given NetMan user, or to the profile that this user belongs to, then that desktop is loaded instead.*

The length of the HTML page generated is determined by the number of entries in the desktop.

**Tip**

*This type of presentation is advisable only if the specified desktop does not contain a large number of NetMan application calls, so that all of the main information can be seen at a glance.*

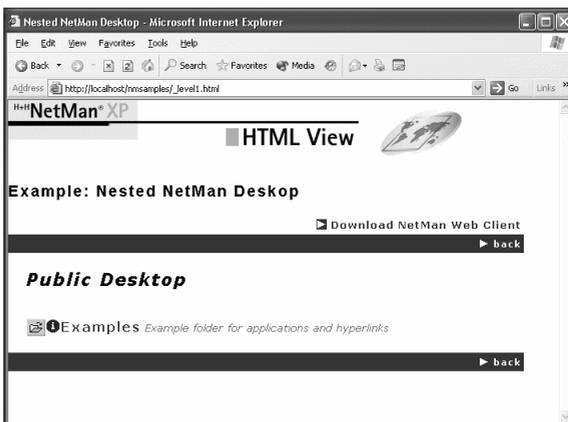
## Embedding a Nested Desktop

The HTML comment `<!-- @NM_DESKTOP_SINGLE_LEVEL -->` also embeds the default NetMan desktop defined in the HTML View Settings; in this case, however, each level of the desktop's directory structure is on a separate page. Hyperlinks lead to lower desktop levels.

### Example:

```
<html>
..<!-- @NM_DESKTOP_SINGLE_LEVEL -->
..
</html>
```

### Result:



The difference between this option and an expanded desktop becomes clear when a very full desktop is displayed. Since it displays only the highest level of the desktop hierarchy, the nested desktop allows a clear overview.



### Tip

*This option is especially useful, for example, if the desktop you want to present contains a large number of configurations in a highly structured thematic organization.*

**Note**

Generally, users click on the browser's "Back" button to return to the next higher desktop level. HTML View, however, gives you the option of integrating your own "Back" link, by adding the @NM\_BACK placeholder. The advantage of this option is that you can add a URL to the hypertext reference, which the client browser is then pointed to if the user is already at the highest desktop level when he or she clicks on this "Back" link. You could, for example, insert the URL of your own home page, as we have done in this example:

Example:

```
<html>
..
<a href="@NM_BACK=http://www.hh-zfrk.com" >
<font face="Arial" size="4"><strong>zurück</strong></font>
</a>
<!-- @NM_DESKTOP_SINGLE_LEVEL -->
..
</html>
```

**Tip**

If you wish to provide your users with a browser in "kiosk" mode (a browser with no navigational tools) and have decided to present the nested desktop, this "Back" link is essential to allow your users to navigate through the desktop.

## Embedding Individual NetMan Configurations

Unlike the two placeholders described above, which embed entire desktops, the @NM\_CONFIGURATION marker embeds an individual NetMan configuration in an HTML page. You can embed only those configurations that you have already defined in your NetMan Management Console.

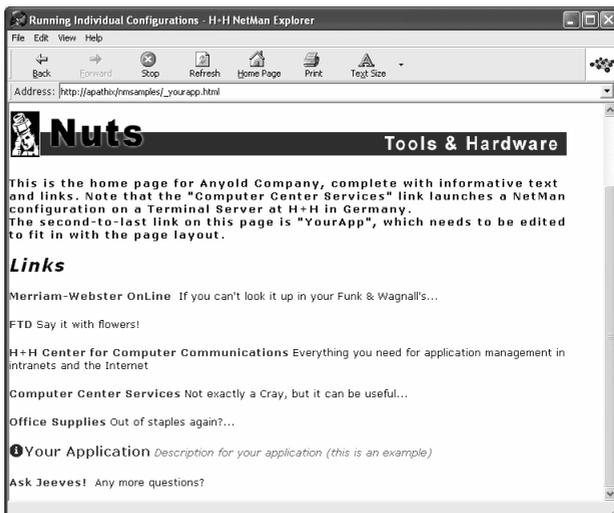
You can insert an @NM\_CONFIGURATION marker at any position in an HTML page; HTML View replaces this with the relevant hyperlink data before the page is passed to the browser. You can also edit HTML templates to adapt the display of these links to specific page layouts.

### Example:

```
<html>
..
<!--@NM_CONFIGURATION="YourApp"-->
..
</html>
```

The entry (in quotation marks) specifying the @NM\_CONFIGURATION must correspond to a valid configuration ID in the NetMan Management Console.

### Example:



## Selecting the Language

If you have the NetMan Language Module, you can use the HTML `@NM_LANGUAGE` placeholder to define the language used in the information files that describe your applications to users. This makes it easy, for example, to design multilingual pages. When generating the HTML page, NetMan HTML View refers to the language-dependent texts in the NetMan Management Console.

### Example:

```
<html>
..
English Desktop
<p>
<!-- @NM_LANGUAGE="ENGLISH" -->
<!-- @NM_DESKTOP_SINGLE_LEVEL -->
<p>
German Display
<p>
<!-- @NM_LANGUAGE="GERMAN" -->
<!--@NM_CONFIGURATION="VLB"-->
<p>
English Display
<p>
<!-- @NM_LANGUAGE="ENGLISH" -->
<!--@NM_CONFIGURATION="VLB"-->
..
</html>
```

If a particular language is assigned to a given user or user's profile, HTML View automatically uses the specified language when the user in question is logged in, without having the language declared in the HTML page.

The language modes currently supported by NetMan are:

- English
- German
- French



## Embedding HTML Code

With the HTML marker `@NM_INCLUDE`, you can insert your own HTML code in your HTML pages. This makes it easy to use HTML formatting in all of your HTML pages. You can embed only the HTML sequences that exist in the `include.htm` file in the selected template directory (see Chapter “Directories in HTML View”).

For example, say you have a Java script that you would like to embed in all your HTML templates. (As a matter of fact, the templates in *Default.htf* use a number of Java scripts.)

*In the example below, HTML View uses an include instruction to integrate the JavaScript for a link to an info file:*

```
<html>
..
<!--@NM_INCLUDE-->
..
<!--@NM_CONFIGURATION="YourApp"-->
..
</html>
```

Excerpt from the *include.htm* file:

```
<script language="JavaScript">
<!--
function newWindow(szUrl)
{
window.open(szUrl,'Remote','titlebar=no,scrollbars=yes,resizable=yes,width=640,height=480');
}
-->
</script>
```

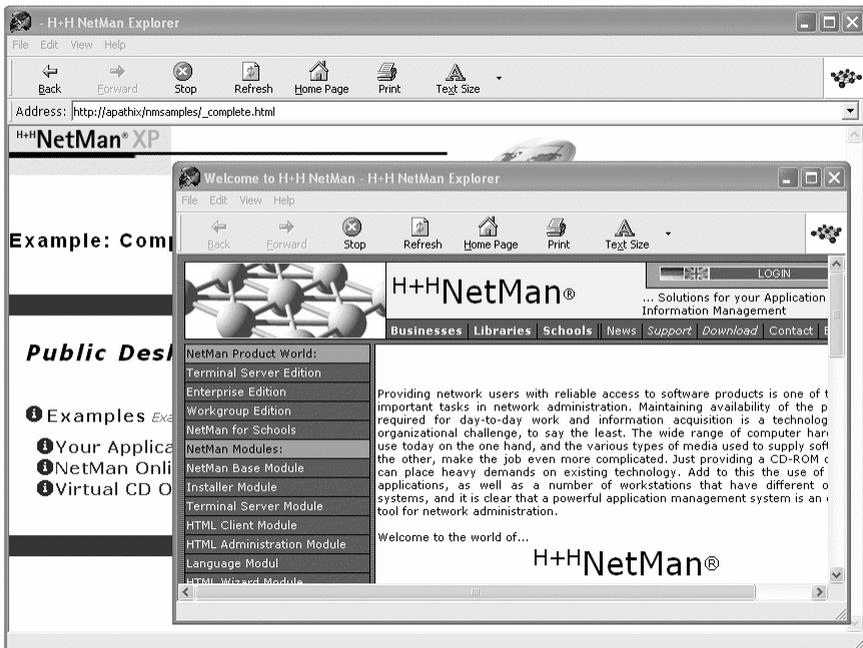
Resulting HTML file:

```
<html>
..
<script language="JavaScript">
<!--
function newWindow(szUrl)
{
window.open(szUrl,'Remote','titlebar=no,scrollbars=yes,resizable=yes,width=640,height=480');
}
-->
</script>
..
<!--@NM_CONFIGURATION="YourApp"-->
..
</html>
```

## 4. The First Steps

The following chapter guides you through the first steps of HTML View configuration, and an example of a commonly used initial configuration is given. The settings described may not address your particular requirements, but this example gives you an in depth look at the process logic of the HTML View Settings program, which will help you in configuring your own settings. The Chapter “HTML View Settings”, on the other hand, provides only an overview of the available options for your reference.

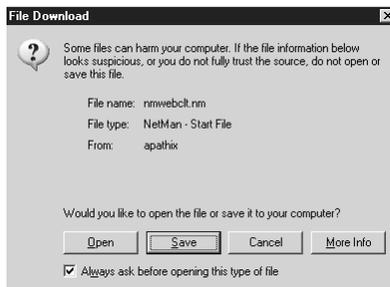
Following HTML View setup on your Apache server (see “Installing NetMan HTML View”), you can load the sample pages described above in your browser (see the Chapter “Designing HTML Pages”, for details on creating your own HTML pages). In this example, we open the “Public Desktop” page in a new browser window, without changing any of the default HTML View Settings, and click on the hyperlink configuration, “NetMan On-line”. HTML View “knows” exactly what to do, because HTML documents use a standardized syntax, and HTML View has a format template for opening the page:



Clicking on “Your Application”, however, a container configuration link, will in all probability open the “File download” dialog:

This is because the default HTML View Settings are configured to use *NetMan start files*, but the Mime type, *application/x-netman*, is not yet registered in the browser.

Link the Mime type by running the NetMan command line program, (%NMHome%\bin\nmcmd32.exe), if you wish to use NetMan start files. To do this, select OPEN THIS FILE FROM ITS CURRENT LOCATION and open NMCmd32.exe in the NetMan ‘Bin’ directory. Now the Windows Calculator should open, concluding this example.



#### Note

*For more information on registering \*.nm files, read the discussion of Mime types in “Application Start on a Client Machine (NetMan Start Files)”.*

This workstation is now configured for opening NetMan application in NetMan HTML View.

Because you usually have links that open applications with system requirements not addressed here, such as drive environments, etc., you will probably need to modify other HTML View settings before all your links are viable. For details on this access method, refer to Section “Application Start on a Client Machine (NetMan Start Files)”.

In most cases, HTML View is used to start applications on a terminal server. Changing this setting is the first step in the next phase of our configuration example, which also includes a number of other commonly used installation parameters.

The objective is a **highly effective, low-maintenance installation** that provides access to NetMan-managed applications for a large number of users and enables the following functionalities:

- Applications started from NetMan’s Public Desktop run on a **MetaFrame server**, using the HTML template files included with the HTML View installation.
- *Applications can be accessed by anonymous users.*

- Access is permitted based on the client's IP address or host name, which can indicate the **location** of the client machine. Furthermore, location-specific log data on access is collected and evaluated by NetMan's statistics program.
- Access is platform-independent.

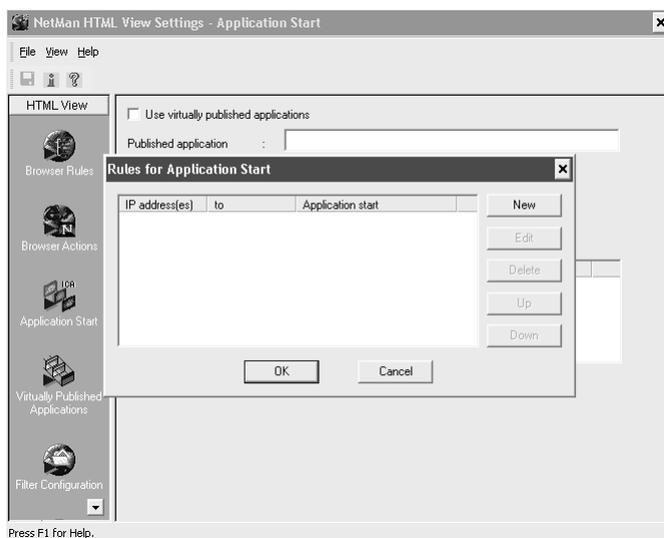
This example is based on the following installation parameters:

- *The NetMan Base Module, the Apache server and HTML View are all installed on one Citrix MetaFrame server.*
- HTML View has been installed as described in the Chapter "Installing NetMan HTML View".

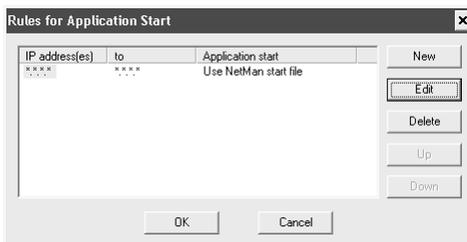
The configurations required include settings in the following software components:

- NetMan HTML View
- MetaFrame server
- Apache HTTP server

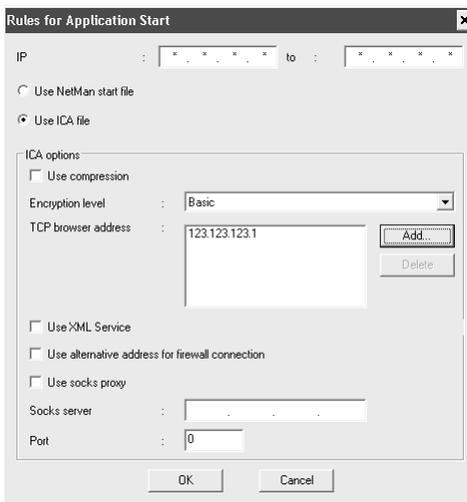
HTML View is configured through the HTML View Settings program, which you can open by running *NMWebcfg.exe* from the 'Bin' directory of your HTML View installation. We shall now modify the default settings so that ICA files, rather than NetMan start files, are sent to client browsers when application links are activated. To do this, open the APPLICATION START dialog in the HTML View Settings and click on RULES under IP-DEPENDENT OPTIONS:



At this point, the list is empty; no rules have been defined for the application start. Click on **NEW** to add a new rule; this opens the **RULES OF APPLICATION START** dialog (see below). An explicit rule specifying the use of NetMan start files, for example, might look like this in the list:



We will now add a rule for sending ICA start files:



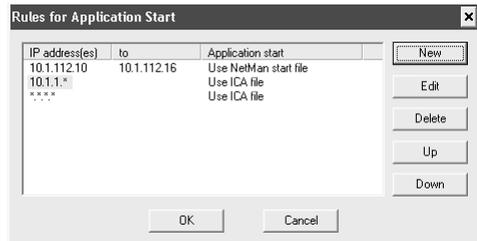
## Note

*This manual does not go into detail concerning ICA-specific configuration options. The dialogs are generally adapted to those used in the Citrix Program Neighborhood and are described in the relevant Citrix manuals.*

The rules are scanned sequentially, which permits flexible configuration and combination of IP-address dependent rules.

### Example: List of IP-address Dependent Rules

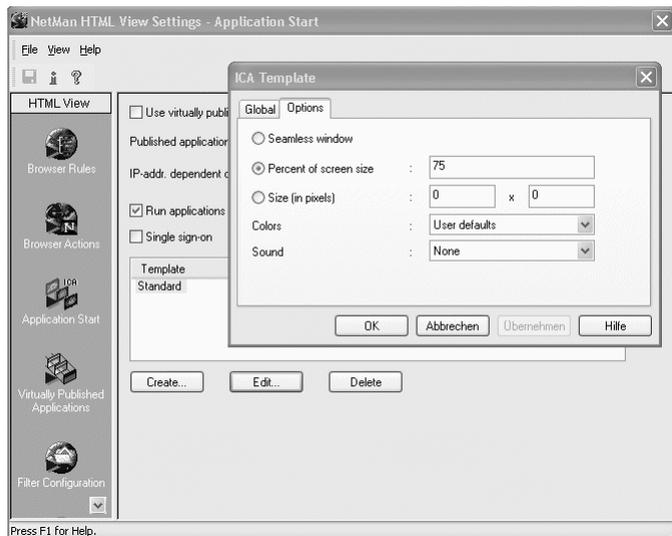
In this example, we want to use NetMan start files for some of the network stations, while other stations should be sent ICA files. Furthermore, the MetaFrame server has an internal IP address, which means a distinction must be made between ICA access from within and from outside a firewall. The solution could look something like this:



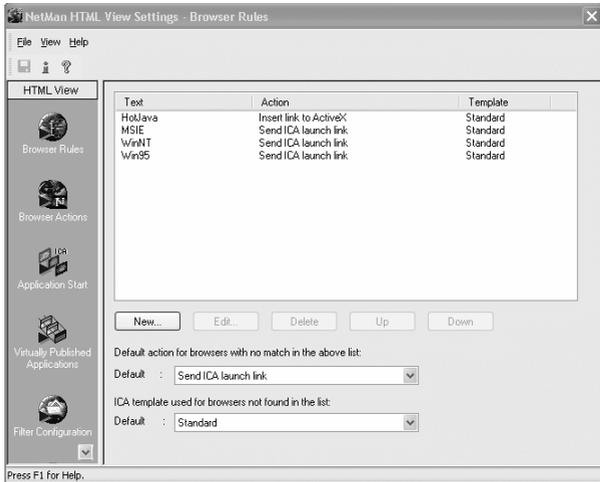
When defining the last rule, we selected the USE ALTERNATE ADDRESS FOR FIREWALL CONNECTIONS option in the RULES OF APPLICATION START dialog.

No more than one rule is required here, however, as the object is simply to have ICA files sent rather than NetMan start files, and access is location-independent.

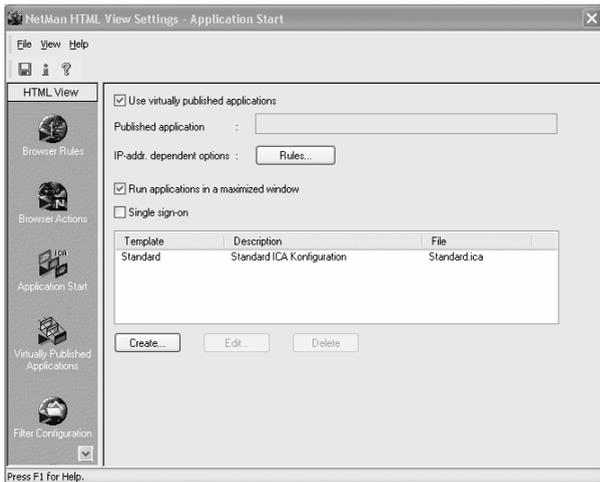
Once this application start rule is formulated, the *Standard.ica* ICA template is automatically the template for all start routines,



because this is the default setting in the HTML View browser rules:

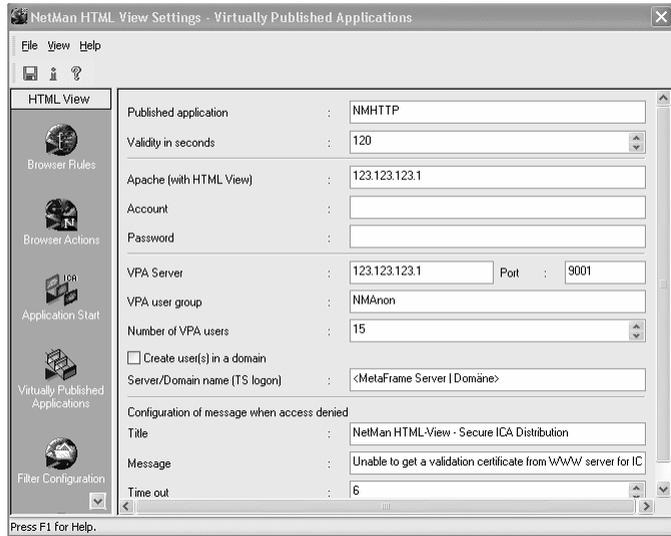


Getting back to the application start: we now select the USE VIRTUALLY PUBLISHED APPLICATIONS option, to enable access for anonymous users while keeping administration tasks to a minimum:



The procedure for setting up virtual application publishing is described under “Virtually Published Applications”. That section described in detail how and where to install the required VPA service and define the published application.

In the current example, the virtually published application is configured as shown here:



These settings have the following effects:

1. When an application start is activated, an ICA file is generated according to IP- and browser-dependent rules. This ICA file is sent to the client machine and starts *NMHTTP*, the virtually published application. HTML View determines an available user name from the *NMAnon* group and requests a new password from the *VPA server* (on a terminal server).
2. The client machine passes the ICA file to the Citrix WebClient, which starts a session on the MetaFrame server. The *NMHTTP.exe* program asks the Apache HTTP server which application the user wishes to start.
3. HTML View passes the start parameters to the program in question. If the user is unknown to HTML View, or if 120 seconds have elapsed since the ICA file was sent, the error message configured under MESSAGE WHEN ACCESS DENIED is output.

If you test an application start now, you will see that these settings are sufficient to trigger the start of the application in question on the MetaFrame server.

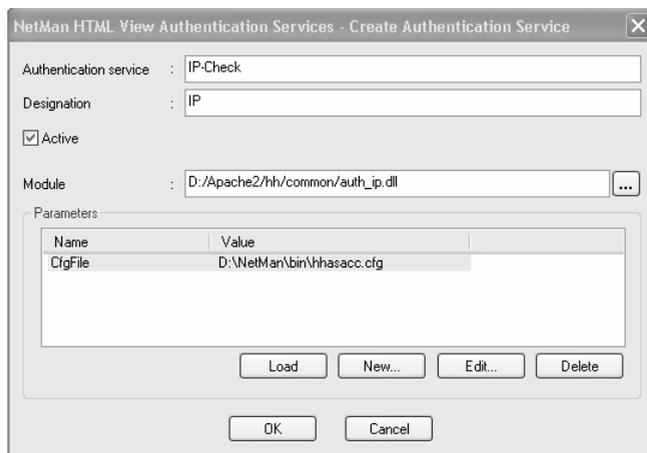
Two of the desired functionalities listed have not yet been specifically addressed:

- platform-independent access, and
- location-specific permissions.

Platform-independent access is provided through the minimum requirements defined in *Standard.ica*, which is defined here as the default template for ICA configuration. (Platform-independence can be further optimized under *Browser Rules*.)

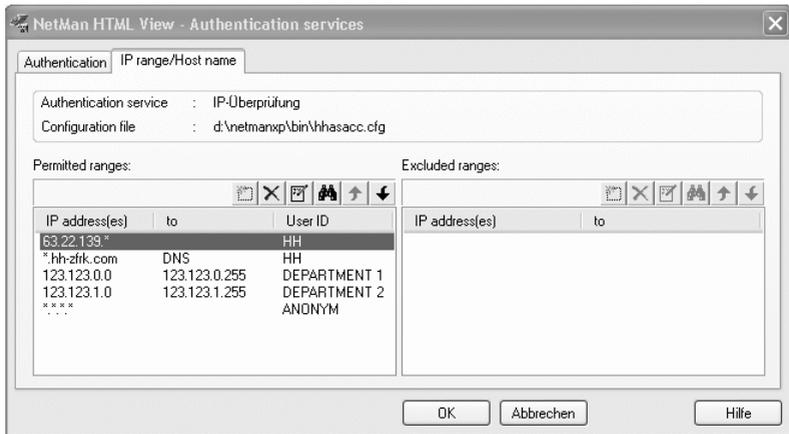
Location-specific permissions can be defined—including for anonymous users—in the H+H authentication service **IP Address Assessment**. NetMan user IDs can be generated based on client IP address and/or host name and used as the basis for 'execute' permissions. These user accounts can also belong to user groups, and applications can be made available based on group membership. All you have to do is configure and activate the helper program that checks IP addresses, as follows.

- Start *NMAuthset.exe* from the *HTML View* working directory (*.\HTML-View\Bin*).
- Activate the **USE AUTHENTICATION** setting.
- Click on to open the **CREATE AUTHENTICATION SERVICE** dialog, and configure the settings shown here:



Click on **LOAD** to view the parameter name. Click on **EDIT** and enter the full name of the file (yet to be written) in which the IP addresses to be permitted are stored together with their corresponding NetMan user names.

In this example, the configuration file defined for the Terminal Server Access Control program is used. This file might already contain entries, if it has been used to control access through the NetMan Client or the HTML View's predecessor, NetMan HTML Client.



### Tip

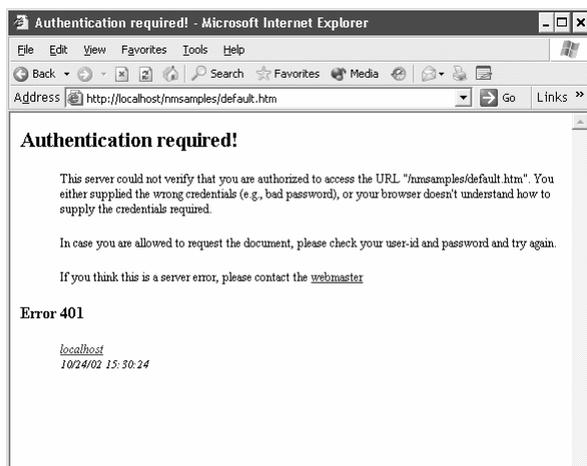
*If you wish to provide access through the NetMan Client as well, it is best to enter the hhasacc.cfg file here; otherwise, you would have to make the same configurations in two files.*

The configuration pictured above has the following effects:

- Clients accessing the system from the 'H+H' domain are logged under the NetMan user name "H+H".
- A client with the IP address 123.123.0.111 is logged on to NetMan as "DEPARTMENT1".
- If the IP address of the client is not one of the two specified addresses, nor within the range configured for "H+H", the client is logged on as "ANO-NYM".



If the last entry in the list shown here was not configured, the client described above would be denied access, and a message dialog is opened. If no other authentication service is defined, they cannot log on, as no basis for logon is available. In that case, the standard Apache error message is shown:



If another authentication service is configured and the user has an account that meets the requirements it defines, the user is logged on to NetMan under the account entered.

The authentication module used for this mechanism is described in detail in the Chapter “H+H Authentication Services”.



#### Note

*At first glance, this example may seem fairly complicated. Keep in mind, though, that these are one-time configuration steps. Once completed, you have at your disposal an extremely powerful mechanism that provides seamless integration of software components from different manufacturers. From now on, every configuration you add in your NetMan Management Console is automatically assigned a URL that you can publish as widely as you choose without lifting a finger, simply by on the basis of the location-specific access permissions that you configure.*

## 5. Troubleshooting

HTML View is a complex component of the H+H NetMan system, insofar as it provides interfaces to a number of software components from other manufacturers, including browsers, MS Terminal Services, Citrix MetaFrame, plug-ins, and so on. If you have any problems with the NetMan HTML View module, keep in mind that they may not be caused by HTML View directly, but may stem from configurations in other components that interact with HTML View.

### *Restarting HTML View*

Some of the configuration settings you modify are not effective until you have shut down and then restarted HTML View. To do this, shut down your Apache server and reboot it. This reloads HTML View, which then loads the new configuration settings.

### *Errors*

If you do have any problems, checking the following factors may help you locate or eliminate possible error sources:

#### *Apache HTTP Server*

- Does the account used by the Apache service have rights to the resources it requires for serving the HTML pages? Does this account have the required permissions in NetMan databases?
- Did you restart the Apache service after making alterations in HTML View settings or authentication services?
- Did you copy the NetMan HTML View filter DLL (*nmwebclnt.dll*) to the Apache HTTP Server? (must be performed manually only if you have Apache HTTP 4.0 or later)

#### *Client Browser*

Are the required client components (such as Netscape plug-ins or the ICA WebClient) installed and registered?

One known problem in this context is the following:

Background:

- You use the MS Internet Explorer.
- Only the ICA WebClient is installed on the client machine.

The Internet Explorer does not call the ICA WebClient as an external application, but rather asks the user how this object is to be handled - this means the Internet Explorer does not have a reference to the ICA WebClient.

This indicates an error in the ICA WebClient setup. There are two ways to solve this problem:

1. Directly after installing the ICA WebClient, open the Windows Explorer (not the Internet Explorer) on the workstation in question. Select VIEW / FOLDER OPTIONS and go to the FILE TYPES dialog page. Under REGISTERED FILE TYPES, select *Citrix ICA Connection* and click on EDIT.... Then click on OK without changing any settings.
2. Use the Registry editor to open the Registry on the workstation where you have installed the ICA WebClient, and make the following changes:
  - Add a new subkey, called application/x-ica, to the *HKEY\_CLASSES\_ROOT\MIME\Database\Content Type* key.
  - In the new subkey, add a *string* entry called *Extension*.
  - Enter the value *.ica* for the new entry.

After these configuration steps, you can use the ICA WebClient with the NetMan HTML View and the Microsoft Internet Explorer.



#### Tip

*Use the NetMan WebClient to avoid these problems.*

### **Terminal Server Access**

- Have you created the required published applications?
- Is the terminal server physically accessible to the end user?

### **HTML View Settings:**

- Have you defined the desired HTML pages in the HTML View filter configuration ?
- Have you created the required gateway accounts with passwords? Gateway accounts are user accounts that enable access to components on other servers. These can include NetMan components on various servers (for example, if the HTML View and NetMan Base installations are on different servers) or user databases on NT or NetWare servers.
- Do the gateway accounts have the required access rights?
- Have you selected a valid NetMan desktop?
- If you work with virtually published applications: is the VPA service correctly installed?
- Have you double-checked which operating systems are excluded from access?

*This list does not claim to be complete, but it should give you an overview of the main sources of errors.*

*The following section provides suggestions on localizing and diagnosing errors in your system.*

## **Diagnostic Tools**

### **Windows NT Event Viewer**

HTML View outputs error messages and warnings to the Windows NT Event Viewer. Such events would be recorded in the application log; check there for help in locating problems.

### **NetMan Trace Monitor**

The Trace Monitor can be used to view status and error messages from HTML View. Start the Trace Monitor on the same computer on which HTML View runs to track internal processes in HTML View. Refer to the on-line Help in the Trace Monitor for more information. If your HTML View is installed on a terminal server, you cannot use the Trace Monitor; instead, you must record events in a log file. The procedure for this is described below.

### **Log Files**

You can record the process and error messages output by HTML View in a log file, if desired. On a terminal server, this is the only way to track HTML View events. Proceed as follows to activate this function:

1. Change your current directory to the HH common directory (<Apache directory>\HH\common)
2. Edit the *hhmes32.cfg* file so it contains the following settings:

```
[Settings]
TraceToFile=1
TraceFile=c:\apache2\hh\html-view\bin\trace.txt
```

After you restart your Apache service, its internal processes are recorded in the file named in the 'TraceFile=' line in the 'hhmes32.cfg' file.

We recommend deactivating this function again ("TraceToFile=0" and restart the Apache service) once you have completing troubleshooting, as it tends to slow down the HTML View program, and because the log file can become very large in a short space of time.



## 6. Designing HTML Pages

The HTML pages that are supplied with your HTML View are generated on the basis of formatting templates. A distinction is made between two types of templates:

- Templates for the HTML layout of the pages to be generated (see “Templates for Generating Desktop Structures” below), and
- Templates for configuring application starts (see “Templates for Application Starts” below).
- These templates are described in detail in the following; the descriptions assume a basic understanding of HTML code.

### ***Templates for Generating Desktop Structures***

The templates used by HTML View to present NetMan desktops are stored in subdirectories that end in “.htf”. In the HTML View Settings, you can define which directory is used in generating your HTML pages (see “Directories in HTML View” in this manual).

The templates contain placeholders (variables) which the HTML View replaces dynamically—directly before passing the page to the client browser— with the text data in the corresponding desktop elements. For example, the “subtitle.htm” template contains the placeholder “@NM\_DESCRIPTION; this is replaced by the brief description of the application in question, defined in your NetMan Management Console as part of the container configuration.

With the default settings, *Default.htf* is the default template directory. To create your own templates, either modify the templates in *Myformat.htf* (a copy of the default) or create a new directory. We strongly advise against modifying *Default.htf* for use as your templates, as this directory may be overwritten during future program updates.

#### ***The main templates for HTML formatting are:***

<b>Title.htm</b>	Shows the highest desktop level. The name of the desktop is entered here
<b>SubTitle.htm</b>	Gives the title of a subgroup of entries in an expanded (“complete”) desktop
<b>MLink.htm</b>	Gives the title of a subgroup of entries in a nested desktop
<b>Link.htm</b>	Link to a container configuration to be started directly or on a terminal server
<b>WebLink.htm</b>	Link to a hyperlink configuration

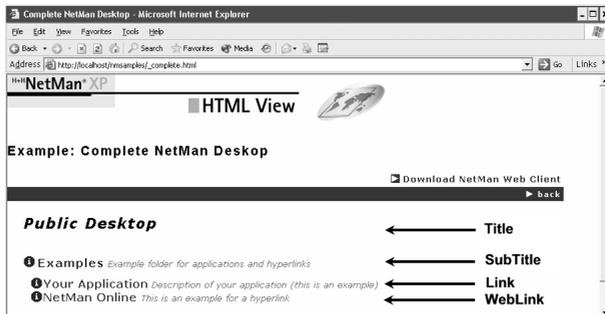


## Note

Each template should be so structured that it can be used as an independent component in an HTML document. For example, *SubTitle.htm*, in the *Default.htf* directory, is written as follows:

```
<table border="0" WIDTH="100%" cellspacing="0">
  <tr>
    <td colspan=2>
      <P ID="subtitle">
        @NM_INFO_LINK@NM_PROMPT
      <span ID="descr">
        @NM_DESCRIPTION
      </span>
      </p>
    </td>
  </tr>
  <tr>
    <td>
      <table border="0" WIDTH="100%" cellspacing="0" cellpadding="10">
        <tr>
          <td>@NM_CONTENTS</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

Thus the display of an entire desktop is created from a chain of templates. In each template, the HTML View inserts text data for the individual desktop elements corresponding to the placeholders found.



If you remove a placeholder, the corresponding information does not appear when the page is opened in a browser. For example, if you delete @NM\_DESCRIPTION, the brief description following the application name is not shown.

Some templates are used only when certain types of actions are configured for the application start:

<b>Embed.htm</b>	Used when the application is inserted in the HTML page by a Netscape plug-in or a Java applet (see “Embed Netscape Plug-in” and “Embed Java Applet” in this manual)
<b>Embed2.htm</b>	Used when the application is inserted by a Netscape plug-in or a Java applet in an HTML page referred to by a hyperlink (see “Embed Netscape Plug-in” and “Embed Java Applet” in this manual)

There are other templates as well that are used only in certain events:

<b>Info.htm</b>	Link to the information file configured for a given application call or desktop. If you do not use information files, references to Info.htm can be removed from the following templates: SubTitle.htm, Link.htm, Mlink.htm and WebLink.htm.
<b>Inactive.htm</b>	Appended as a text constant to a desktop entry if the configuration in question has been deactivated in the NetMan Management Console.
<b>Noaccess.htm</b>	Appended as a text constant to a desktop entry if the client operating system is not permitted access (see “Directories in HTML View” in this manual). This template is used if the application was to have been embedded directly in the HTML page.
<b>Noaccess2.htm</b>	Opened as a separate page if the client operating system is not permitted access (see “Operating Systems” in this manual). This template is used if the application was to have been opened in a separate page.
<b>NoLic.htm</b>	This page is shown if no license is available for the NetMan configuration activated. The user can go to the license queue from here.
<b>NoRight.htm</b>	This page is shown if the user has no rights to execute the NetMan configuration activated. This may happen if the user tries to load the URL of the configuration directly.



The following is an example of the page opened by HTML View when there is no license available for a requested application:



You can define the texts, illustrations and layout of the templates used when access is denied based on IP address. The templates included with the HTML View program serve as examples.

When editing the HTML templates used when no license is available for a requested application, however, please observe the following:



#### Note

*The templates included with the program for this event were created using the Windows editor, as this editor allows creation of non-redundant, clearly structured HTML components. If you use an HTML editor, be sure to check whether the editor automatically adds HTML code that would preclude the template from functioning as an independent component. For example, an HTML editor might append an </html> tag, which you cannot see, when you save a template. Some editors are generally unsuitable for processing format templates; for example, MS Word increases the volume of source code 20-fold when you save an HTML document.*

## Templates for Application Starts

HTML View comes with link templates that enable it to create syntactically correct configuration files (such as ICA and NM files). These files are used with all layouts; in other words, they are not assigned to a certain layout directory, but are valid globally.

Link templates are located in the directory defined in the HTML View Settings (see “Directories in HTML View”). The default directory is `..\HTML-View\Launch`.

You can define whether the application started in HTML View runs on the client PC (started by a NetMan start file) or on the terminal server (started by an ICA file), based on client IP address. To do this, open `HTML VIEW SETTINGS / APPLICATION START` and select `IP-ADDR. DEPENDENT OPTIONS`.

The NetMan start file uses the *Start.nm* template.

**Start.nm**      Template for a NetMan start file

With ICA start files, you can define the form of access on the basis of the client browser type (see “Browser Rules”). Specified “actions” are assigned to each browser type registered. An action generally defines the start file to be used for a given browser. These start files often contain window coordinates, usually with the default settings 801 x 601. This is the only setting that you might need to change in these files.

<b>Plugin_man.txt</b>	Embedded Netscape plug-in for manual start
<b>Java_man.txt</b>	Embedded Java applet for manual start
<b>Plugin_aut.txt</b>	Embedded Netscape plug-in for automatic start
<b>Java_aut.txt</b>	Embedded Java applet for automatic start

You can also assign a template to each browser type, with settings for window size, display resolution, colors and sound (see also “Templates for ICA Files” in Chapter 0), for access on MetaFrame terminal servers. These templates are generated according to your settings under `HTML VIEW SETTINGS / APPLICATION START / TEMPLATES`. HTML View comes with an ICA start file (to get you started), called `Standard.ica`.

## Placeholders in Templates

The templates contain placeholders which used as markers by the HTML View, and replaced by texts before an HTML page is sent to a Web browser. The texts are taken from various sources, such as databases. The list below describes the placeholders in detail.

<b>@NM_LAUNCH</b>	Link to a start file
<b>@NM_PROMPT</b>	Name of a configuration
<b>@NM_DESCRIPTION</b>	Description of a configuration
<b>@NM_INFO</b>	Link to an information file
<b>@NM_INACTIVE</b>	Message output when a requested configuration has been deactivated



@NM_CATEGORY	Category of a configuration
@NM_URLCATEGORY	Category of a configuration, with spaces replaced by underline characters (“_”). This allows the use of the text within the URL.
@NM_EMBED_APP	Action assigned to the client browser according to your browser rules

## Using Style Sheets

Because page formatting with the HTML View is distributed over a number of independent HTML text components, it can be difficult to change a format. This is why the default formatting files in Default.htm now use style sheets.

Below is an excerpt from the “\_nm.css” style sheet, which you can use to change fonts and font sizes. This style sheet is used, for example, by the *SubTitle.htm* excerpted above.

```
#title {  
    font-size: 16pt;  
    font-weight: bold;  
    letter-spacing: 2px;  
    font-style : oblique;  
}
```

```
#subtitle {  
    font-size: 12pt;  
    font-weight: bold;  
    letter-spacing: 2px;  
}
```

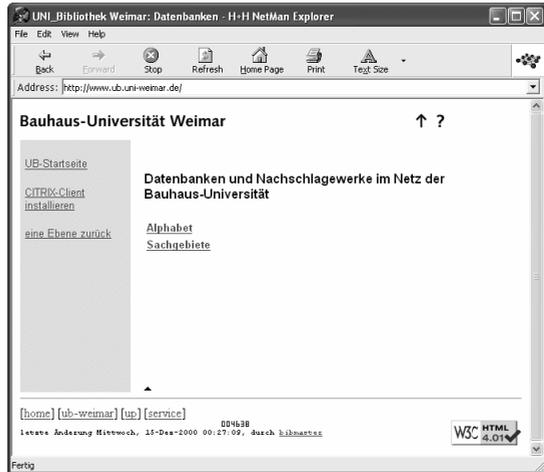
```
#descr{  
    font-size : 10pt;  
    font-weight: 500;  
    letter-spacing: 0px;  
    font-style: italic;  
    color: #606060;  
}
```

```
#link{  
    font-size: 12pt;  
    font-weight: bold;  
    letter-spacing: 2px;  
}
```

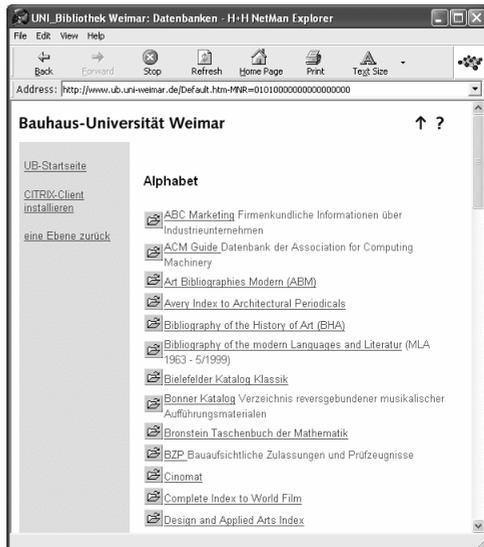
## Concluding Remarks

Customizing formats with the HTML View is even easier to do than it is to describe here. If you already have experience in Web design, you can probably integrate the basic HTML View functions in your Web without even reading this manual, as most of the modifications involve standard, as opposed to NetMan-specific, operations.

We have seen a wide variety of original NetMan HTML page designs created by our customers. Here is just one example:

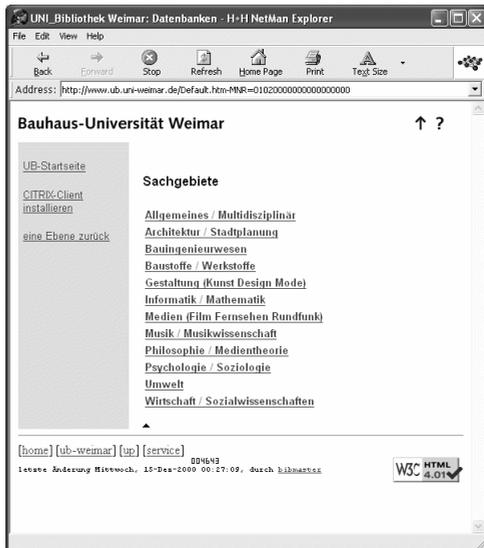


You can hardly tell that HTML View is even used here, thanks to the nested desktop format. The desktop has two lower levels, organizing their spectrum of information by alphabet in the one folder:

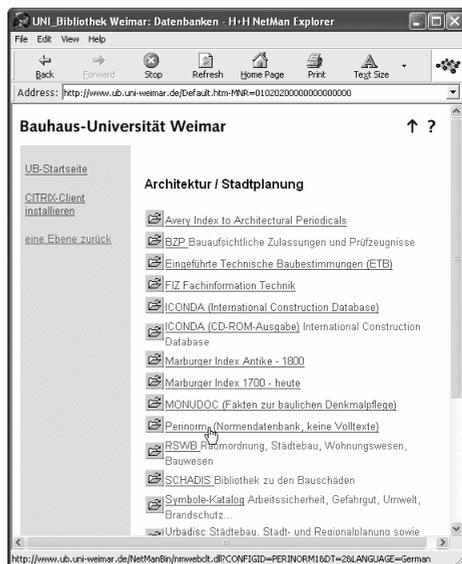




and by topic in the other:



Under “Architecture / City Planning” we find a range of information that contains both hyperlinks and applications, with the latter running on a terminal server:



Thus NetMan is quite invisible to the user, and operates “only” as information and application management software in the background. In the foreground are the presentations and information from the institution deploying NetMan.

## 7. HTML View Settings

To start the NetMan HTML View Settings program, call the *nmwebcfg.exe* program from the HTML View working directory (..\HTML-View\Bin) on the console of the server where HTML View is installed.

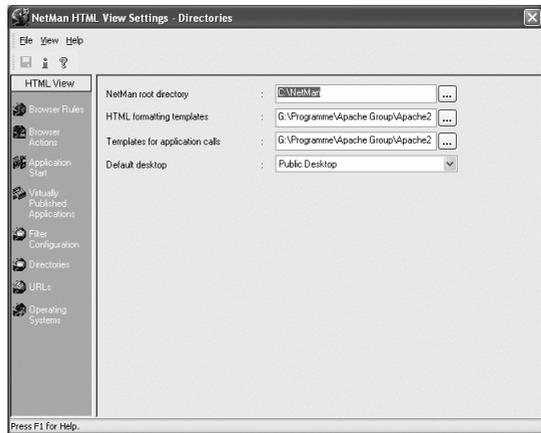
All parameters for HTML View are configured here. In some cases (indicated below), modifications in these settings are not effective until you restart HTML View (and any client Web browser running while the changes were made). To restart and re-initialize HTML View, shut down and restart the Apache HTTP server.

The HTML View Settings program has a number of categories, which are described in detail below. These descriptions provide a handy reference guide to the range of options available for settings, complementing the practical introduction to the use of HTML View given in the preceding chapters.

To save changes in settings, either click on the “diskette” icon or select FILE/SAVE.

### Directories in HTML View

The directories required for operating HTML View are defined in the DIRECTORIES category. Modifications in these settings are effective only after HTML View is restarted (i.e., by restarting the Apache service).



The NETMAN ROOT DIRECTORY is entered in the first field. This tells HTML View where to find the NetMan databases and configurations that you wish to present in HTML View. Among other things, this is where HTML View reads the desktop and configuration data that you present in HTML pages.

What you enter here depends on whether the NetMan HTML View and the NetMan root directory are on the same computer or not (see “Installing NetMan HTML View”).



- NetMan root directory and HTML View on the same computer:  
Enter the local path of the NetMan root directory (for example, d:\hh\netman).
- NetMan root directory and HTML View on two different computers:  
Enter the UNC path to the NetMan root directory (for example, \\server\share\hh\netman) in the NETMAN ROOT DIRECTORY field. To enable HTML View to access this network resource, the Apache service has to run under a user account with privileges in the NetMan databases. Ideally this account should be located on the same server that NetMan is installed on. Enter the account name and the password in the Apache Service properties.

In the HTML LAYOUT TEMPLATES field, enter the HTML template directory for NetMan desktop and configuration layout. This directory must have the extension *.HTF* and must contain the desired HTML template files. For detailed information about HTML template files, see “Templates for Generating Desktop Structures”.

In addition to HTML pages, HTML View also generates the start files that make it possible to launch configurations from HTML pages. You can define these launch events based on client browser type (see “Templates for Application Starts”). HTML View refers to a second type of template file to generate these start files. The directory containing these templates is specified in the TEMPLATES FOR APPLICATION STARTS field in the dialog box shown above. For more information about these templates, see “Templates for Application Starts”.

In the DEFAULT DESKTOP field, enter the default desktop for HTML pages. You can choose from the NetMan desktops contained in this list.

**Tip**

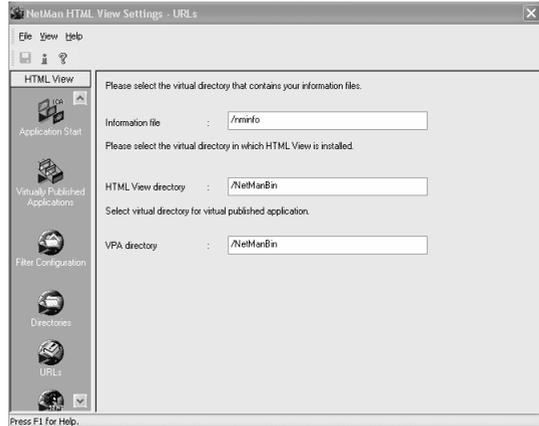
*You can configure different desktops for the standard NetMan Client and HTML View. For example, you can define a desktop for HTML View that shows only the NetMan configurations that you wish to present in your HTML pages, rather than all of the configurations available in the NetMan Client. Enter this desktop in the DEFAULT DESKTOP field described above.*

**Note**

*If a different desktop is assigned to a given NetMan user or station, or to the profile that the user or station belongs to, then the desktop specification in the HTML View Settings or in the HTML page is ignored.*

## Defining URLs in HTML View

NetMan HTML View requires virtual WWW directories located on your Apache HTTP server. These directories are created automatically when you install HTML View. For more information, see “Virtual WWW Directories”.



In the **INFORMATION FILE DIRECTORY** field, enter the virtual directory defined for NetMan information files. HTML View refers to this URL to present information files. The standard name for this directory is *nminfo*. This is a subdirectory of the NetMan Base installation, not of the HTML View installation.

The **HTML VIEW DIRECTORY** and **VPA DIRECTORY** are similar to working directories. HTML View uses these directories to generate your links. These are virtual directory that are accessed, and that do not have to contain program files. The default directory is *NetManBin*. This directory name was chosen for reasons of compatibility with the older HTML Client module, the predecessor to HTML View. In the earlier version, *NetManBin* pointed to the HTML Client installation directory (<HTML-Client-Installation>bin). With HTML View, it is no longer necessary to have the program files in the directory used to generate HTML pages. As mentioned in “Virtual WWW Directories”, the virtual *NetManBin* directory is empty.

Unless there is a specific reason for doing otherwise, confirm the default virtual directory, *NetManBin*, in the **HTML VIEW DIRECTORY** and **VPA DIRECTORY** fields.



### Note

*If you enter a different directory in **VPA DIRECTORY**, PLEASE NOTE THE FOLLOWING: The user account that starts NMHTTP.EXE on a terminal must have access to the virtual directory*

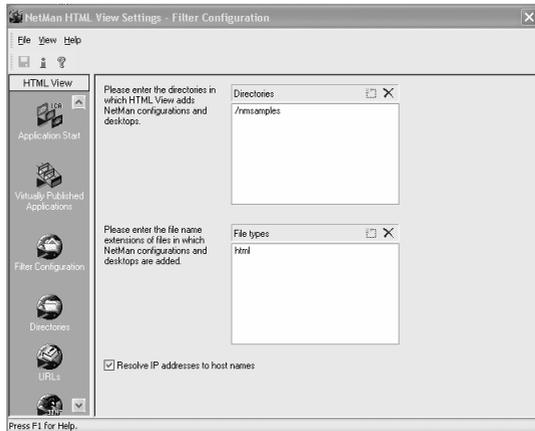
you enter here. If you have configured a system logon for this directory in H+H Authentication Services, you need to a user account in the VIRTUALLY PUBLISHED APPLICATIONS category, under APACHE (WITH HTML VIEW) / ACCOUNT that exists in the database for one of the defined authentication services.

## Filter Configuration

The Chapter, “How HTML View Works”, explains how HTML View inserts NetMan configurations and desktops in HTML files. Technically, HTML View is an extension for the Apache HTTP Server that filters data before it is output to client browsers.

In the FILTER CONFIGURATION category, you can define which files are checked for the placeholders that HTML View fills in. Every time a user accesses one of your HTML View pages from a Web browser, HTML View checks the corresponding files for HTML markers (described under “How HTML View Works”) before the page is sent to the client, and modifies them as needed. Files stored on your Web server that are not included in the Filter Configuration definition are not processed by HTML View.

In the DIRECTORIES and FILE TYPES boxes, click on the “New” icon (square) to add an element, and on the red “X” to delete the selected element.



The first field, DIRECTORIES, defines which directories HTML View checks for HTML placeholders. These are virtual directories in the Apache HTTP server on which your HTML View is installed.

In the second field, FILE TYPES, enter the file name extension(s) of the file type(s) that HTML View is to process. Files of this type are processed only if found in one of the directories entered in the DIRECTORIES field.

**Note**

The extension *.htm* is handled internally the same as *.htm\** would be, and includes files with the extension *.html*, for example. This is why we specified only the *.html* extension in the default settings. Of the files included with your HTML View installation, those which do not require processing before being opened in a browser have the extension *.htm* and are thus excluded from HTML View processing, which spares system resources.

Enable or disable the RESOLVE IP ADDRESSES TO HOST NAMES option to turn host name resolution on or off. If you do not require host name resolution in your environment, we recommend turning this option off (deselecting it) because the attempts by HTML View to resolve host names where this is not possible can hinder performance.

HTML View requires name resolution if access permissions are assigned according to host names.

**Application Start**

In the APPLICATION START category, you define how HTML View reacts when a NetMan container configuration is activated, triggering a sequence of actions. There are basically two possibilities:

1. The actions run locally, on the client machine. This has a number of implications:
  - The client must be running a 32-bit Windows operating system.
  - The client must be able to access NetMan, as well as any network resources required by the application started.
  - There must be no drive conflicts between NetMan, the application started, and the configurations in the client machine.
2. The actions run remotely, on a terminal server with Citrix MetaFrame.

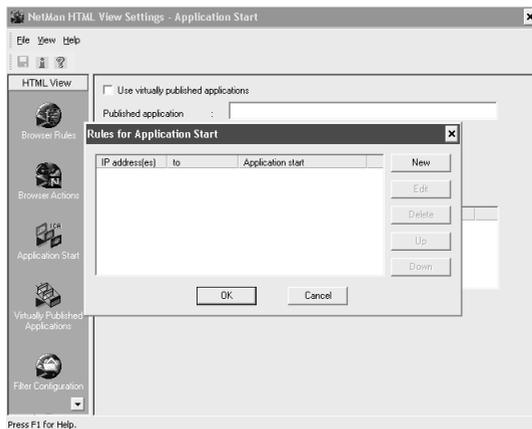
In the former instance, HTML View is used in a LAN and replaces the NetMan Client as user interface. *NetMan start files* are sent to client browsers, and run the requested NetMan configuration using the NetMan command line program.

In the latter case, certain types of file must be transferred (plug-ins, applets, ICA files) that start NetMan configurations on *MetaFrame servers*.

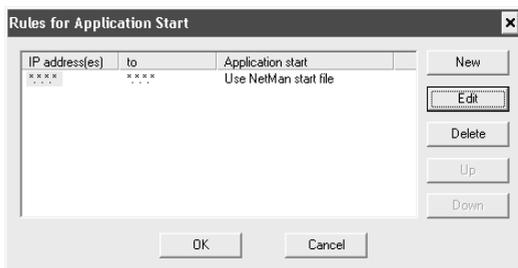
With the default HTML View settings, *NetMan start files* are sent. For details on address-dependent options, see “The First Steps”.

## Application Start on a Client Machine (NetMan Start Files)

With the default HTML View settings, *NetMan start files* are sent. The RULES FOR APPLICATION START box, opened in HTML VIEW SETTINGS / APPLICATION START / IP-ADDR. DEPENDENT OPTIONS by clicking on RULES, is empty.



The configuration that specifies the default use of NetMan start files is as follows:



With this technique, the application selected in the Web browser is started on the client computer. The NetMan Kernel operates in the background to perform any runtime recording and license monitoring functions configured for the application in question.

When this type of link is activated, HTML View creates a file with the extension *.nm* and sends it through the Web server to the browser together with the MIME type *application/x-netman*.

If you are already familiar with the term "*MIME type*", you might want to skip the following discussion.



## Discussion

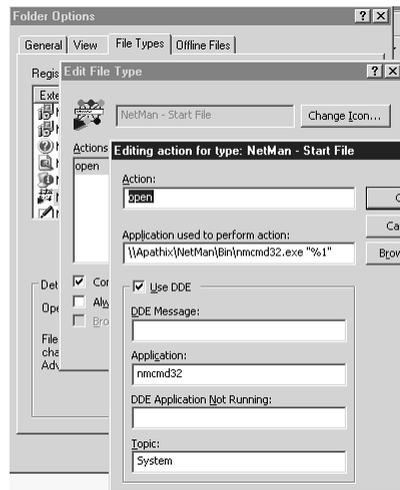
The “MIME type” of a file tells the browser what program to open the file with. MIME types are registered in the browser and are associated with specified file name extensions. For example, when a standard HTML page is sent, the accompanying MIME type is `text/html`. The program associated with this MIME type is the browser software, so the file is opened and processed by the browser.

The MIME type sent with an `.nm` file is `application/x-netman`. Once this MIME type has been registered in the browser, the browser “knows” that, when it receives a file with the `.nm` extension, it must start the NetMan command line program, `NMcmd32.exe` (see “How HTML View Works”), and pass the incoming file to that program. `NMcmd32.exe` interprets the file, starts the NetMan kernel in the background and presents the desired application on screen.

The content of NetMan start files follows this pattern:

```
[Config]
ConfigurationID=<Configuration ID>
```

The command that calls the NetMan command line program, `NMcmd32.exe`, is registered in your browser automatically the first time you call `NMcmd32.exe` from the ‘Bin’ subdirectory in the NetMan root directory. Once it has been registered, all files with the `.nm` extension received by the browser are opened with the `NMcmd32.exe` program. The Internet Explorer and Netscape Navigator both support this automatic registration mechanism. In other browsers, the MIME type must be registered manually as shown on the right:



## Note

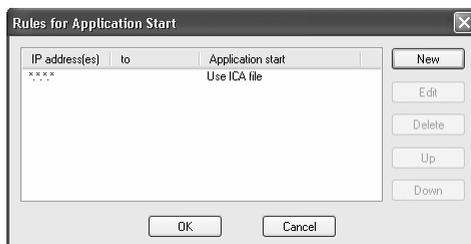
*With this option, all of the program logic runs on the client workstation. NetMan and the*

application launched must be able to run in parallel on the client workstation. This means that any components required by these programs on the local machine, such as INI files, DLL files or Registry entries, must be present. The user must be able to log on to the network and must have any access rights that may be required. All of these requirements are exactly the same when the standard NetMan Client interface (as opposed to NetMan HTML View) is used. The browser serves only as a different user interface.

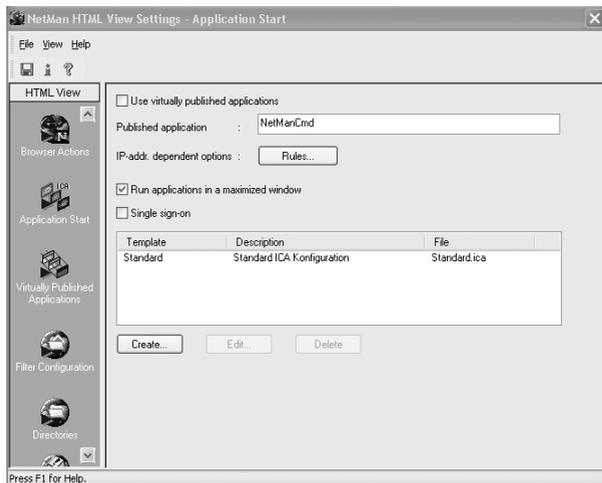
### Application Start on a Terminal Server (ICA Start Files)

With this technique, the application is started remotely on a MetaFrame server. This does not require a Windows platform, nor is it necessary to modify any settings in the client machine specifically for the application in question. The only client-side requirement is that the ICA Client is installed.

To enable application launches on a terminal server, enter **Use ICA file** under RULES FOR APPLICATION START (in HTML VIEW SETTINGS / APPLICATION START / IP-ADDR. DEPENDENT OPTIONS).



If the USE VIRTUALLY PUBLISHED APPLICATIONS option is disabled, you need to enter the name of a published application defined in Citrix that contains the NetMan command line program call with a placeholder for command line arguments taken from ICA files. In our example, this published application is called "NetManCmd".

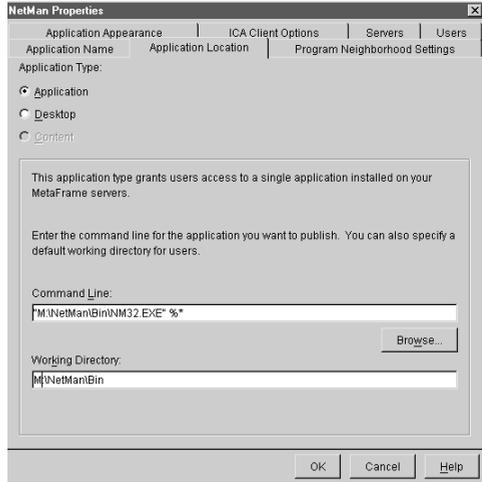


The concept of application publishing was developed by Citrix. For instructions on publishing an application, please refer to the relevant Citrix documentation.

In the PUBLISHED APPLICATION field, enter the name of the *published application* defined in the Citrix management console that starts the NetMan command line program. The command line might look like this:

```
"M:\NetMan\BinNMCmd32.exe" %*
```

The command line variable „%\*“ *must* be appended to the program call. The quotation marks are also required:



## Virtually Published Applications

Virtually published applications are published by the NetMan HTML View and represent an extension of the Citrix concept of published applications, in that they serve to increase security.

This feature enables an elegant presentation of NetMan configurations through HTML View. The use of virtually published applications offers the following advantages:

- You need publish only the NMHTTP.exe program. This program provides access to all of your NetMan configurations.
- ICA files sent by HTML View are marked as “expired”, and remain for only a brief period in the client’s browser cache.
- ICA files created on the client machine are discarded as invalid.



### Note

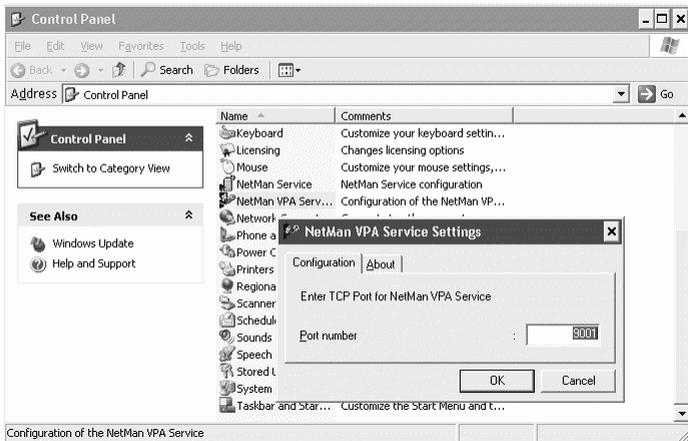
*This mechanism improves day-to-day control of your overall system, because clients cannot launch applications that they cannot see in their Web browser.*

There are 3 configuration steps required before you can use virtually published applications:

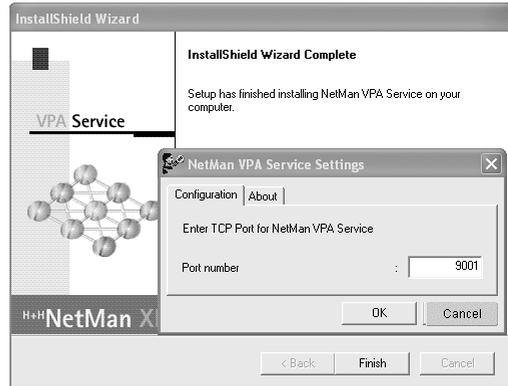
1. Install the service for virtually published applications (VPA service) on
  - your MetaFrame server (without load balancing), or
  - on any server in your domain (with load balancing).
  - The setup program for the VPA service is stored under `..\HTML-View\Setup.vpa` in your NetMan root directory.
  - In most cases, you can use the default port configured:



In case of conflict with other software components, you can change the VPA port in the Windows settings:



The user account entered for the VPA service must have permission to create local users on the MetaFrame server (without load balancing) or in your domain (with load balancing). We recommend creating a special user account for this purpose, such as the “nmvpa” account in the example below:



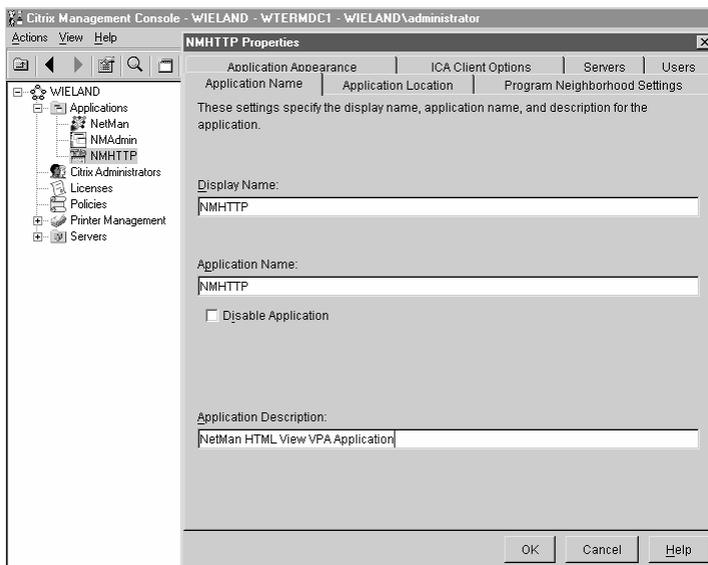
The VPA service creates NetMan user accounts on the fly as needed, with an internal password. If you wish to assign a user profile or other properties, you need to configure this manually.



#### Note

*The VPA user is created automatically the first time you run the VPA service. This means you can try a test launch first, before defining user properties.*

- With the virtually published applications mechanism, NetMan applications are “channeled” through *one* “real” published application, called “*NMHTTP*”. Enter this name when you create the published application:



- The *NMHTTP* application must be created *for explicit users* and must be executable by a user group called *NMAAnon*. The *NMAAnon* group must be created manually.

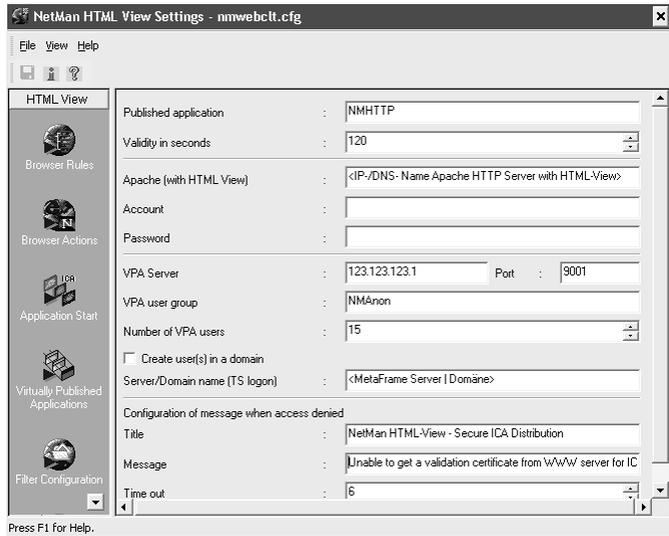


### Note

*The NMAAnon group must be created manually before you can enter it here.*

- Enter the full name of *NMHTTP.exe* as the command line, with its ‘Bin’ directory as the working directory.

- Enter these settings in the VIRTUALLY PUBLISHED APPLICATIONS dialog in the HTML View Settings program. Some fields are already filled in for you:



The following settings can or must be added:

### ***Published application***

Enter the name of the “real” published application on your MetaFrame server or in your server farm.

### ***Term of validity (in seconds)***

This determines how long an ICA file sent by your Apache HTTP Server is valid. Once this time period has elapsed, the file is discarded as invalid. With this setting you can prevent users from starting applications on your Terminal Server through “handmade” ICA files. The *NMHTTP.exe* program on your terminal server accepts only ICA files created and sent by HTML View.

### ***Apache (with HTML View)***

Enter either the IP address or the host name of the server where your Apache HTTP Server and HTML View are installed.

### ***Account***

If the HTML View virtual directories are not configured for authentication, you need to leave this field /and the next one) blank. The same applies if a “Check



IP Address “ authentication service is configured to grant the terminal server access privileges in the virtual directory specified under URLs / VPA DIRECTORY. Otherwise, enter an account under which the *NMHTTP.exe* program can log on to this virtual directory, to check whether the access is permitted and which program to start. This user account must exist in the database of an activated authentication service.

For detailed information on authentication, please see “H+H Authentication Services”.

### ***Password***

Enter the password assigned to the account entered directly above.

### ***VPA server***

Enter the name of the server on which the NetMan VPA service is installed.

### ***VPA port***

Enter the port over which your Apache HTTP communicates with the VPA service. The default is port 9001. If you change this value here, make sure you change it in the Windows Settings>Control Panel under NetMan VPA Service as well.

### ***VPA user group***

The VPA service creates users that belong to this group and have permission to call the published application that publishes NetMan applications. This user group is either a local group on a MetaFrame server (without load balancing) or a domain group (with load balancing).

### ***Number of VPA users***

The number of required NT users created by the VPA service. The number should be based on the number of MetaFrame and NetMan licenses available.

### ***Create domain users***

This is where you define whether the VPA service should create local or domain users. Local users should be created if you use only one terminal server. If you work with multiple terminal servers and load balancing, it is essential that you select the “Create domain users” option.

### ***Server or domain name (TS logon)***

Enter the name of your single MetaFrame server (if the VPA service creates local users) or your domain (if you selected CREATE DOMAIN USERS).

**Title**

Enter a title for the message box displayed when the ICA file is invalid. An ICA file is invalid if the server entered under **APACHE (WITH HTML VIEW)** does not confirm that it sent the file in question, within the period entered under **TERM OF VALIDITY (IN SECONDS)**, to the user logging in.

**Message**

Enter the text of the message displayed in the event described above, under **TITLE**.

**Timeout**

Enter the period of time, in seconds, for which the message (see above) is displayed. When this period has elapsed, the user's session is ended automatically.

**Browser Rules**

The browser rules define how HTML View reacts to a given Web browser type when the client activates a link to an application on a MetaFrame server.

**Note**

*With the default configuration, HTML View is configured to launch applications using NetMan start files. Before the settings you configure under **BROWSER RULES** can have any effect on a given workstation, you need to configure a rule for application start called **Use ICA file**, under **APPLICATION START / IP-ADDR. DEPENDENT OPTIONS**, for the station's IP address.*

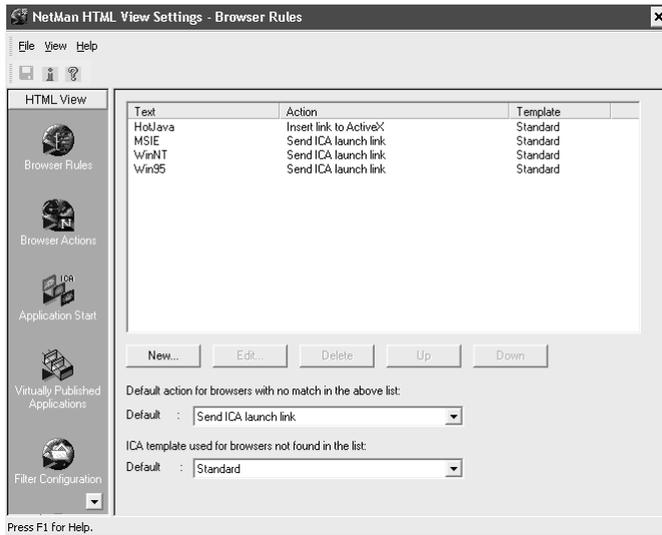
The **BROWSER RULES** category lets you define which Citrix WebClient (for access to the Citrix server over wfica32.exe or a Netscape plug-in, for example) is used, as well as – using templates – the appearance of the user interface (window size, resolution, colors sound).

**Note**

*The **BROWSER RULES** are closely associated with the elements configured under **APPLICATION START**. For example, the **ICA TEMPLATES** that you can assign in the **BROWSER RULES** are defined in the **APPLICATION START** settings category.*

HTML View recognizes browsers by their *userAgent property*, which identifies the browser version and type, as well as the operating system that the browser is running under.

Your initial HTML Client installation includes a few rules as examples:



HTML View checks these rules sequentially when a browser requests an application start. The text in the first column from the left is compared with the text in the client *userAgent* property; if a match is found, the action associated with the corresponding browser rule is carried out, using the designated ICA template.

The default ICA template included in the HTML View installation is compatible with Windows, Macintosh, and Unix platforms.



### Tip

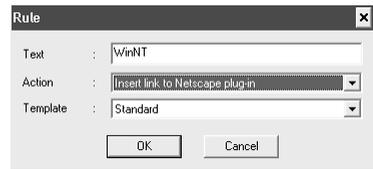
*Because the *userAgent* property provides an indication of the client platform, you can use these browser rules for platform-specific optimization of ICA access (for example, by selecting the “Seamless window” option for Windows platforms).*

**Example**

The userAgent property of the Internet Explorer always contains the string “MSIE”. With the settings shown here, an *JCA launch link* is sent, using the default template (“Standard”), whenever HTML View is accessed from an Internet Explorer regardless of client operating system.

You can add or modify rules; in the example below, an operating-system specific rule is added.

Click on **NEW** to define a new rule. New rules are added at the end of the list. Click on **EDIT** to modify an existing rule.



Use the **UP** and **DOWN** buttons to change the order of the list. You can define more than one rule for a given string. The first match found, working from top to bottom of the list, is run. This makes it easy to test new rules, for example, by briefly switching rule positions.

Click on **DELETE** to remove a rule from the list.

**Note**

*Unfortunately, browser manufacturers have not agreed on a uniform convention for userAgent properties. For example, the userAgent property for the Netscape Navigator version 4.05 includes the following string: Mozilla/4.05 [en] (WinNT; I), while the Internet Explorer 5.0 has the string: Mozilla/4.0 (compatible; MSIE 5.0; Windows NT). The two strings are identical up through the first 11 characters; furthermore, they have different ways of designating operating systems (“WinNT” as compared to “Windows NT”).*

The action you specify under **DEFAULT RULE FOR BROWSERS...** is executed when the browser’s userAgent property does not match any of the rules.

The following actions are defined in HTML View, all of which refer to MetaFrame server access:

- Send ICA launch link
- Embed Netscape plug-in
- Embed Java applet
- Insert link to Netscape plug-in
- Insert link to Java applet

***Each action is described in detail in the following.***



#### Note

*For more information about the various forms of terminal server links, see the “Terminal Server Module” manual. When a user activates a link to a terminal server, both NetMan and the application are launched in a terminal server session, and not on the client computer. The task of the NetMan HTML View is to generate HTML pages dynamically with links to terminal servers.*

*Due to design problems in the ICA ActiveX controls, it is not possible at present to use them with HTML View. You can, however, assign the ICA Launch Link and Java Applet actions for the Microsoft Internet Explorer. Furthermore, with the Microsoft Internet Explorer version 3.02 or later, you can use the ICA Netscape plug-in. This plug-in must be manually integrated, however, because the Netscape plug-in installation program does not support the Internet Explorer.*

### **Send ICA Launch Link**

When a user selects an application from a page generated by HTML View, an ICA file is sent from the Web server to the ICA Web Client on the client machine. The mechanisms used here are explained in detail under “Application Launching” in the “Terminal Server Module” manual.

## Embed Netscape Plug-in

When this action is activated, HTML View generates an HTML page with a Netscape plug-in embedded directly in the page. The emphasis here is on the term “directly”.



### Note

*Direct embedding means that wherever HTML View inserts a reference to a NetMan application call, the HTML code for the Netscape plug-in is embedded in that position. Thus you may have an HTML page with a large number of embedded plug-ins. If each plug-in has an automatic connection function, this could result in several terminal server sessions being opened simultaneously when this HTML page is opened. To avoid this, our “Embed Netscape plug-in” action embeds a plug-in for manual connection to the terminal server.*



### Tip

*This option is useful if your NetMan configurations are spread out over a number of HTML pages (such as in the nested desktop), or when you create HTML pages that have only one application per page. We do not recommend using direct embedding with a fully expanded desktop.*

## Embed Java Applet

The distinctive feature of the Java computer language is that it is platform-independent, which means that it can run in any browser that has a Java Virtual Machine (JVM) version 1.1 or later. Unfortunately, access over the ICA Java applet is fraught with difficulties and disadvantages that have their origin in the Java language itself (see “Embedded Java Applets” in the “Terminal Server Module” manual). With this option, the Java applet is embedded directly in the HTML page. Thus this action has the same potential drawbacks as mentioned above concerning the directly embedded Netscape plug-ins.

## Insert Link to Netscape Plug-in

This action is similar to the one described above in Section 0, under “Embed Netscape Plug-in”. The only difference is the way the embedded application is displayed on the HTML page. When the application link is activated, a new HTML page is generated and displayed. The Netscape plug-in with the link to the terminal server is embedded in the new page and started automatically.



### Note

We recommend using this option when you present a fully expanded NetMan desktop as an HTML page.

## Insert Link to Java Applet

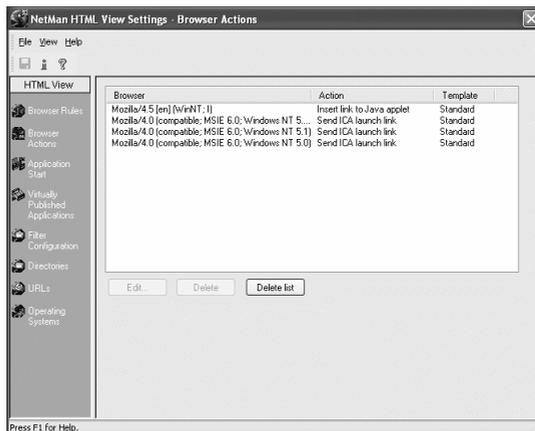
This action is similar to the one described above under “Embed Java Applet”, with the exception that in this case, the Java applet with the link to the application is embedded in a new page.

## Browser Actions

The BROWSER ACTIONS dialog automatically lists the userAgent properties of all browser types that access HTML View through the ICA Client. When an unknown type of browser accesses HTML View, its userAgent property is added to this list, together with the assigned action. You can edit these entries as desired (see also “Application Start” in C).

This list is generated automatically, providing you with a constant overview of the browser types accessing your system.

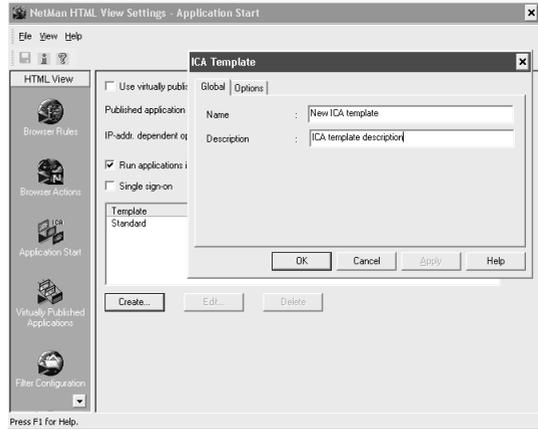
When you add a new rule or change an existing one—or even just to test whether old browser types are still accessing your HTML View—you may want to delete the entire list and let a new list be generated.



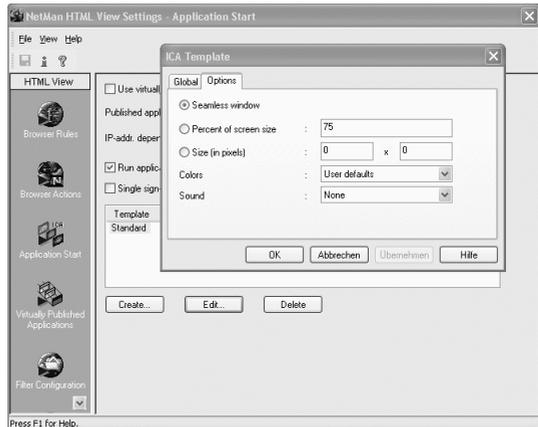
### Templates for ICA Files

In the HTML View Settings program, under APPLICATION START, you can configure ICA template files and assign them according to client platform and browser type.

You can create, edit and delete ICA templates:



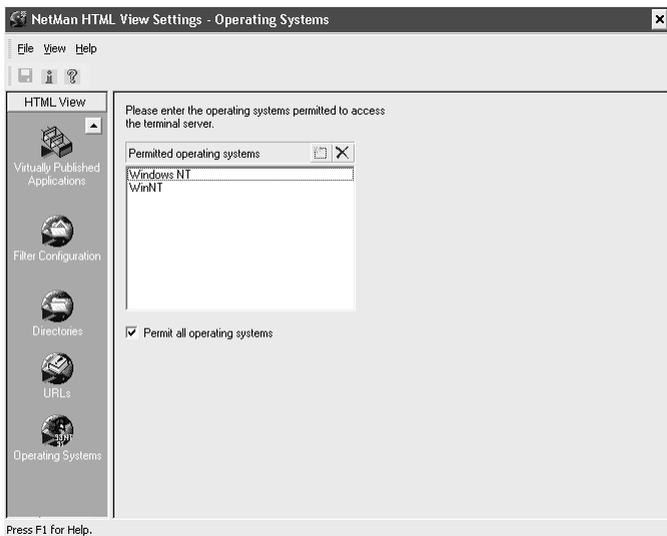
For descriptions of ICA-specific settings, please refer to the relevant Citrix documentation.



## Operating Systems

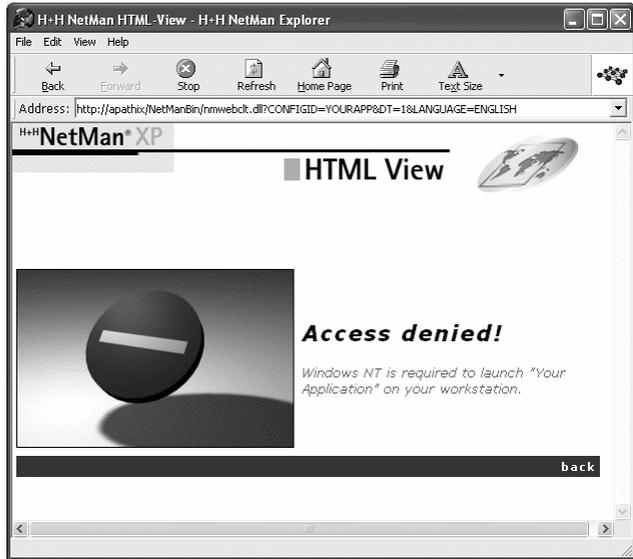
With HTML View, you can specify the client operating systems that are allowed to access your terminal server. For example, if you are running MetaFrame from Citrix on a Windows NT 2000 Terminal Server, Microsoft's licensing specifications require every client that accesses the server to have a valid Windows NT 2000 license.

The NetMan HTML View determines the client's operating system from the browser's userAgent property (see "Browser Actions" above).



Select the OPERATING SYSTEMS category of the HTML View Settings to enter the operating systems you wish to permit. In the PERMITTED OPERATING SYSTEMS field, enter the segment of the userAgent string that designates the client operating system.

When a client attempts access with an operating system that is not permitted, access is refused and an error message is output. The HTML template files *noaccess.htm* and *noaccess2.htm* contain the default error messages; you can edit these if desired. For example, if you enter Windows NT 5.0 so that only Windows 2000 is allowed, you can change the error message text accordingly.



You can switch off the function that checks for client operating systems by selecting the PERMIT ALL OPERATING SYSTEMS option.



#### Note

*The operating system designation in the userAgent property can vary from one browser type to the next. For example, in the Microsoft Internet Explorer property the string Windows NT (the NT version is added in browser versions 5.0 and later) designates the operating system, while the Netscape Navigator uses WinNT to designate the same system. To see the various designations in use, check the list in the BROWSER ACTIONS category; this list registers all the browser types that have accessed the system to date.*



## 8. H+H Authentication Services

If you want to restrict access to the information you provide in HTML View by limiting it to a specified set of user names, IP addresses or host names, you need to define and activate the H+H Authentication Services.

You can configure access restrictions at two levels:

- On the lower level is restriction on the basis of *IP address / host name*. When you use the IP-Authentication Service, all stations within a specified range of addresses are permitted access, with no logon required.
- Additionally, you can integrate other Authentication Services that entail *user logon*.

If only the IP-assessment service is active, stations that are outside the defined IP address/host name range cannot access the system. If only user logon services are active, all users are prompted to log on. When controls are active on both levels, users are prompted to log on only if they request access from a machine outside the defined IP address/host name range.

The following control modules are available:

- IP/host name assessment
- NetMan user login
- NT user login
- Active Directory Services (ADS)
- LDAP over Netscape/Microsoft
- PICA
- SISIS

H+H Authentication Services are implemented in Windows over modules in the Apache HTTP server. With the default settings, the authentication you define with the 'NM-Authset' configuration program applies to all HTML virtual directories created during installation. In particular, the required commands are inserted in the <Apache-Installation>\HH\HTML\_View\bin\NMView.conf file for all virtual directories that contain the markers:

```
### Start-Authentication  
### End-Authentication
```



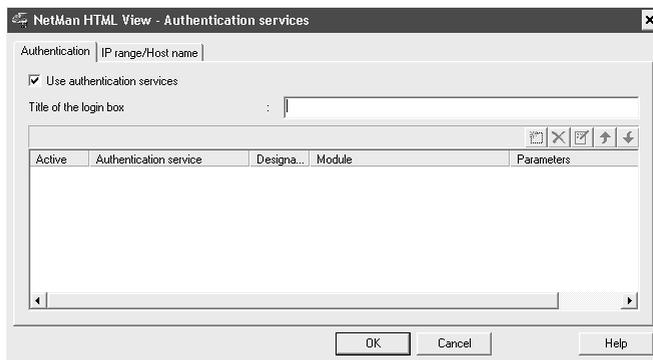
### Tip

*If you wish to define separate access conditions for individual URLs, you can do this by editing the virtual directories in the Apache configuration files. The commands written by the H+H Authentication Services provide examples.*

The configuration program for H+H Authentication Services is stored under  
 <Apache-Installation>\HH\HTML\_View\bin\nmauthset.exe.

### 'Authentication' Dialog Page

This first option on the AUTHENTICATION dialog page lets you turn the H+H Authentication Services feature on and off.



You can also enter a text for the user logon dialog box. With the settings used in the example above, the logon box looks like this:

Below these two fields is a table showing the defined Authentication Services.

The entries in this table are processed in the order in which they appear in this table, from top to bottom (with one exception; see below). Users are logged on with the first applicable set of conditions found. Click on  to add a new entry. To change the position of an entry, select it and click on  or .



Regardless of the entry positions in this list, however, IP/host name assessment services are processed first, and thus take precedence over other services defined here.

The columns in this table have the following functions:

**Active:**

Lets you activate the service.



**Note**

*Please note that an active state indicated here applies only to the program for which you called the H+H Authentication Services configuration program. H+H Authentication Services can be used by more than one H+H program; at present, for example, both NetMan HTML View and Hidden Automatic Navigator (HAN) use this feature. The list here shows the services defined in all H+H programs for which you provide these Authentication Services. The 'activation' checkmark, however, applies only to the program for which the service in question was defined; in this case, for NetMan HTML View.*

*If you disable an entry in this list—for example, for test purposes—that entry is automatically moved to the bottom of the list. When you reactivate the entry, it is **not** automatically returned to its previous position. (Use the arrow buttons as described above to move the entry.)*

**Authentication Service:**

You can enter a name of your choice for the service. If you have more than one H+H program (or think you might in the future) using different sets of access conditions, it is a good idea to enter a name that indicates which program the service is defined for, such as "IP assessment - HTML View" or "IP assessment - HAN".

**Designation:**

Enter a brief code for unique identification of the service. This ID is used to address the service.

**Module:**

Select the DLL file for the desired Authentication Service. The Authentication Services provided by H+H are stored under <Apache-Installation>\ HH\ Common, and are listed in the table below:

Authentication Service	Module
IP/host name assessment	auth_ip.dll
NT logon	authnt.dll
ADS logon	authads.dll
LDAP logon	authldap.dll
NetMan logon	auth_netman.dll
PICA logon	auth_pica.dll
SISIS	auth_odbc.dll

**Parameters:**

Each logon module requires specific parameters, which are configured in the CREATE and EDIT dialog box (click on to edit an existing service). Parameters defined for the H+H logon module—which must conform to a special syntax—can be loaded by clicking on the LOAD button. The example below shows a ready-to-use authentication service for NT user logon:

NetMan HTML View Authentication Services - Create Authentication Service

Authentication service : NT-User Registration (HTML-View)

Designation : HTML\_VIEW\_DOMAIN1

Active

Module : G:/Program Files/Apache Group/Apache2/hh/common/authnt.dll ...

Name	Value
DefaultDomain	DOMAIN1

Load New... Edit... Delete

OK Cancel

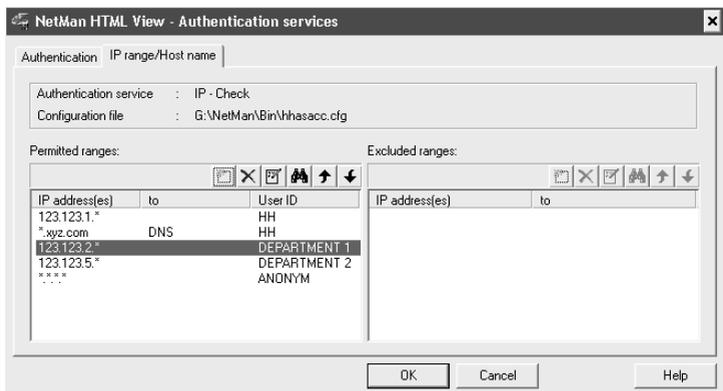
The following table shows the parameters valid for each of the logon modules:

Module	Parameter	Description
auth_ip.dll	CFGFile	CFG file containing lists of permitted and/or excluded IP addresses.
authnt.dll	DefaultDomain	Default domains, applied when the user does not enter a domain at logon.
authldap.dll	CFGFile	Full path to %NMHome%\bin\hhenv.cfg. The LDAP interface is configured in the NetMan Settings.
authads.dll	Object	ADS object for logon
auth_netman.dll	NMHome	%NMHome% in the NetMan directory

With the exception of IP/Host name assessment and LDAP logon, the Authentication Services are ready to run once you enter values for the required parameters. The parameters for IP/host name assessment and LDAP logon are defined in configuration files, which are described in detail in the following.

### *'IP range / Host name' Dialog Page*

The file named on the AUTHENTICATION page for IP address/host name assessment is configured on the IP RANGE / HOST NAME dialog page. You cannot activate this page for input unless an IP address/host name assessment service is selected on the AUTHENTICATION page.



The name of the selected service and the file assigned to it are shown at the top of this page.



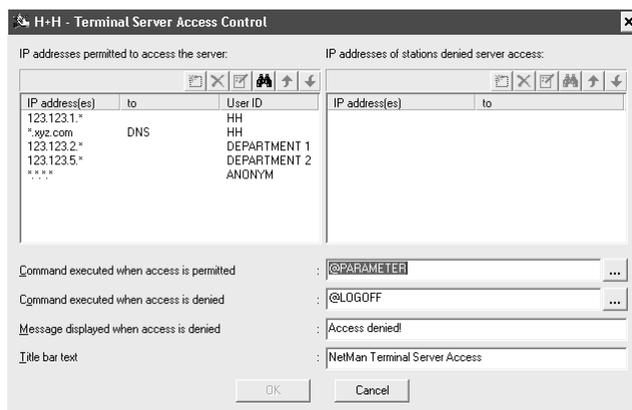
Below this are two lists, showing permitted and excluded addresses or host names. The authentication service detects the client IP address and compares it to:

- The list of permitted host names and IP addresses (including ranges of addresses), and
- The list of excluded host names and IP addresses.

Each list is scanned sequentially. The first match found for the client in question is applied, which means the order in which addresses/names are listed is significant.

If a NetMan user ID is assigned to the applied entry, the user is logged on to your system under that user name; the same ID forms the basis for evaluation of permissions and identification in log files.

If you use NetMan in a Terminal Server environment, you may already have recognized this as a point of interface between H+H Authentication Service for IP address/host name assessment and the Terminal Server Access Control program, a component of the Terminal Server module. The configuration program for Terminal Server Access Control, called from the NetMan Client, looks like this:



This program defines the %NMHome%\bin\hhasacc.cfg file

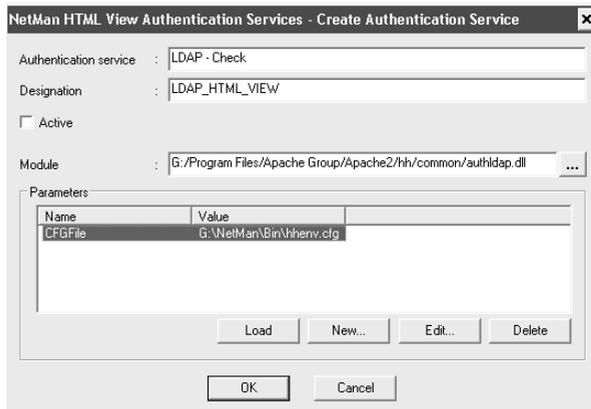


### Note

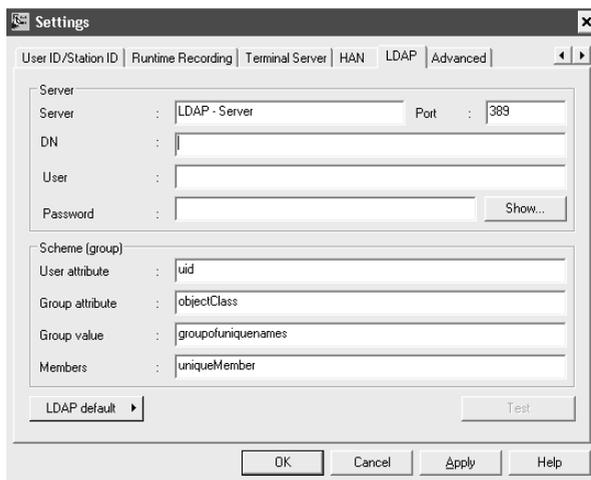
*If you wish to provide both NetMan HTML View and the NetMan Client for your terminal server users, and wish to use the same set of controls for both interfaces, enter the full path and file name for 'hhasacc.cfg' on the AUTHENTICATION page. Please also refer to the example given in the Chapter "The First Steps".*

## LDAP Interface

The parameter required for LDAP logon authentication is the `hhenv.cfg` file, with the full name of the NetMan installation path it is stored on, including the resolved value for `%NMHome%` (fixed value):



This file (`<%NMHome%>\bin\hhenv.cfg`) is configured in the NetMan Settings program:





Proceed as follows to configure the LDAP interface:

- Enter data on the LDAP server you wish to use:
  - Server = Name of the LDAP server
  - DN = Distinguished name of the directory in which users are stored
  - User = User name for LDAP server logon
  - Password = Password for LDAP server logon
- Click on LDAP default to choose one of two group schemes:
  - Microsoft LDAP server, or
  - Netscape LDAP server
- Double-click on a selected server to insert the corresponding attributes:
  - User attribute = This value is used to depict the user name in the corresponding user DN.
  - Group attribute = Name of the attribute
  - Group value = Indicates whether this is a group or not.
  - Member = Attribute in which the members are defined

In designing the LDAP interface, we decided to have the settings configured in the NetMan Settings so that LDAP rights to NetMan configurations can be assigned in the NetMan Management Console. We had to keep in mind that HTML View and the NetMan basic installation might be located on different machines.

## 9. Adapting the Settings in HTML View: Practical Examples

### From 'NMSamples' to Your Own URL

The <server>/nmsamples/ directory is available as a fully viable URL once you have installed HTML View. This is an Apache alias (virtual directory), defined in the file called <Apache-Installation> \HH\HTML\_View\bin\NMView.conf, that points to the <Apache-Installation> \HH\LDAP directory:



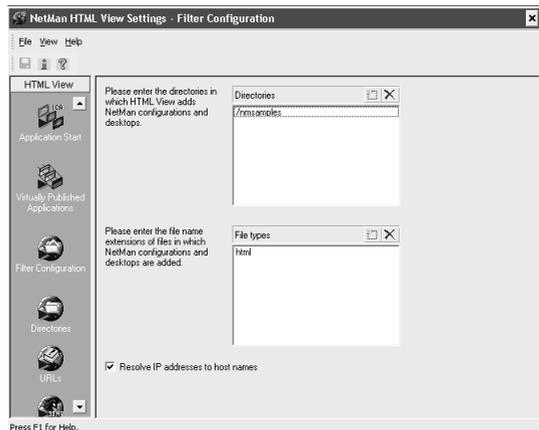
```
nmview.conf - Notepad
File Edit Format View Help
Alias /nmsamples/ "G:/Program Files/Apache Group/Apache2/HH/HTML-View/example/"
<Directory "G:/Program Files/Apache Group/Apache2/HH/HTML-View/example">
  DirectoryIndex default.htm
  ### Start-Authentication
  AuthType Basic
  AuthName "Information service provide by HH"
  HHAuthEnable on
  HHAuthProvider
  HHAuthDefaultProvider
  require valid-user
  ### End-Authentication
  options followSymLinks
  AllowOverride None
</Directory>
```



#### Note

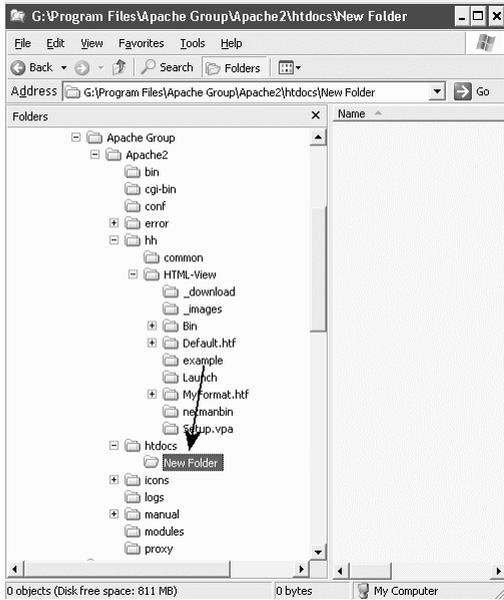
*The comments ### Start-Authentication and ### End-Authentication indicate that this virtual directory is subject to processing by +H Authentication Services.*

In the default configuration of the HTML View Settings, under FILTER CONFIGURATION, the \NMSamples virtual directory is entered as a directory to be processed by HTML View, which interprets HTML comments (see also the Chapter “How HTML View Works”):



In the following steps, we will copy these sample files and store them under another name.

First, we copy the <Apache-Installation> \HH\HTML-View\Example directory to the standard folder for HTML documents on the Apache HTTP server, and rename it "H+H Information System".



Then we copy the section in NMView.conf that applies to the \NMSamples virtual directory and modify it so that the folder ..\H+H Information System as available under the alias \infoservice:

```
nmview.conf - Notepad
File Edit Format View Help
Address G:\Program Files\Apache Group\Apache2\htdocs\New Folder
Folders
  Apache Group
    Apache2
      bin
      cgi-bin
      conf
      error
      hh
      common
      HTML-View
      _download
      _images
      Bin
      Default.htf
      example
      Launch
      MyFormat.htf
      netmanbin
      setup.vpa
      htdocs
        New Folder
      icons
      logs
      manual
      modules
      proxy
0 objects (Disk free space: 811 MB) 0 bytes My Computer

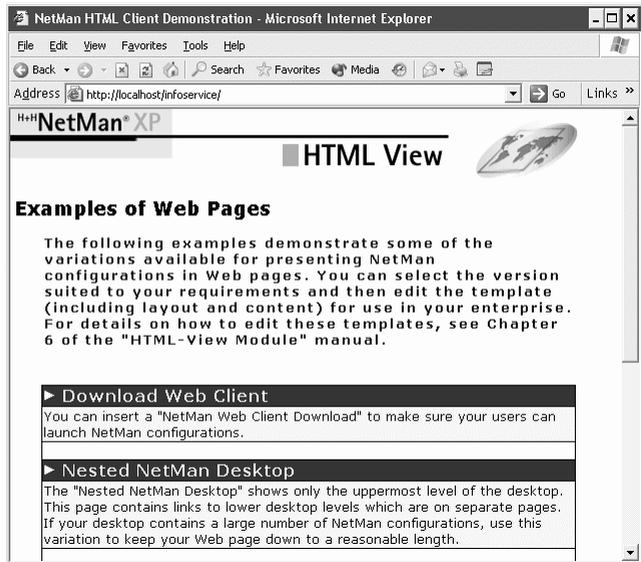
nmview.conf - Notepad
File Edit Format View Help
Alias /msamples/ "G:/Program Files/Apache Group/Apache2/hh/html-view/example/"
<directory "G:/Program Files/Apache Group/Apache2/hh/html-view/example">
  DirectoryIndex default.htm
  ## Start-Authentication
  AuthType Basic
  AuthName "InformationService provide by H+H"
  HHAAuthenable on
  HHAAuthProvider
  HHAAuthDefaultProvider
  require valid-user
  ## End-Authentication
  options FollowSymLinks
  AllowOverride None
</directory>
Alias /InformationService/ "G:/Program Files/Apache Group/Apache2/htdocs/New Folder/"
<directory "G:/Program Files/Apache Group/Apache2/htdocs/New Folder/">
  DirectoryIndex default.htm
  ## Start-Authentication
  AuthType Basic
  AuthName "InformationService provide by H+H"
  HHAAuthenable on
  HHAAuthProvider
  HHAAuthDefaultProvider
  require valid-user
  ## End-Authentication
  options FollowSymLinks
  AllowOverride None
</directory>
```



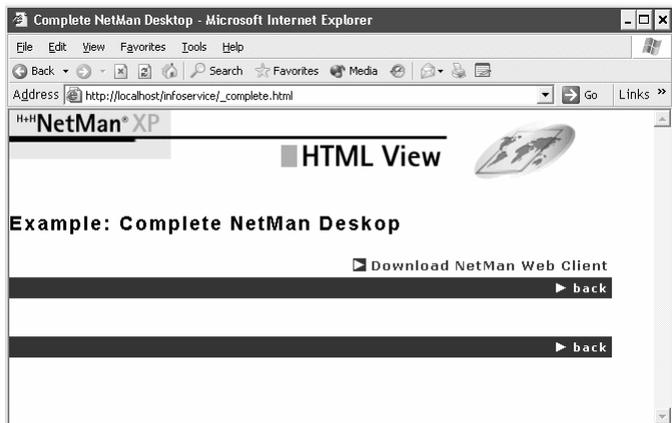
## Note

*Virtual directory names (aliases) are case-sensitive.*

Following a reboot of the Apache HTTP server, the URL `<server> / infoservice/` is available:

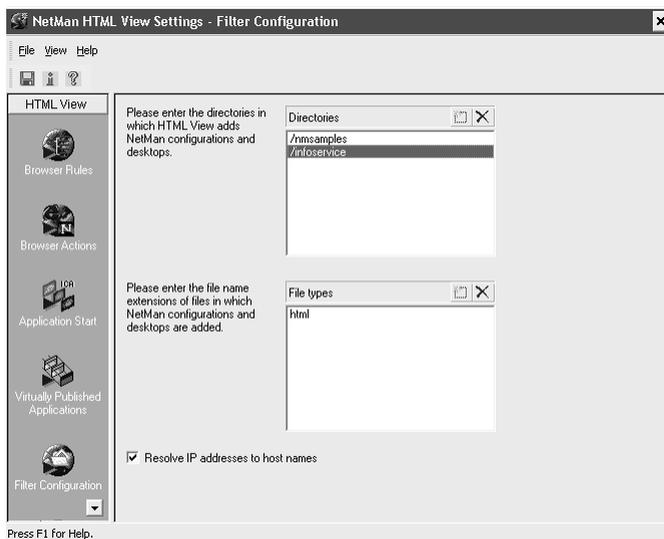


When we open one of the sample desktops, however, the page looks fairly empty:

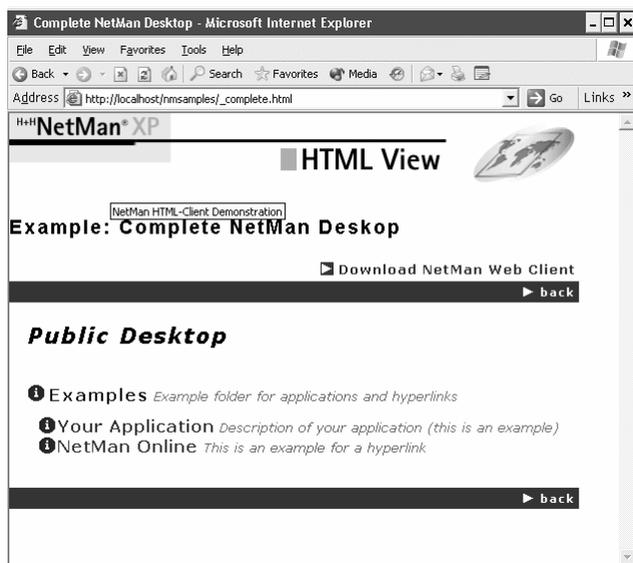


This is because the HTML comments in the sample files have not been interpreted by HTML View. Why not? Because the new virtual directory has not been entered in the HTML View Settings, under FILTER CONFIGURATION.

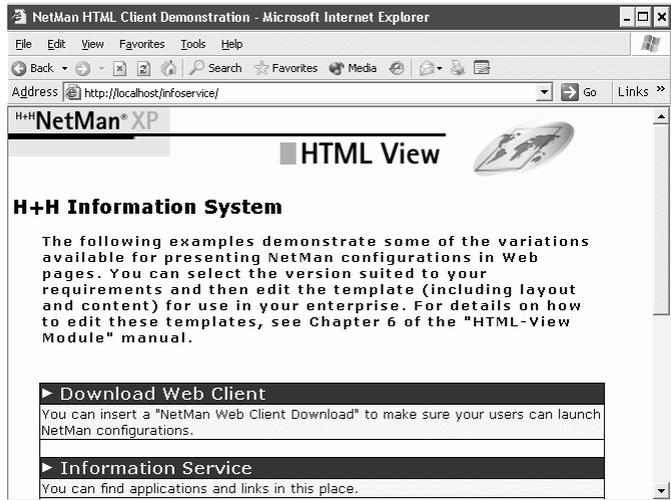
So we will add the new alias now:



Now, the desktop is shown as it was meant to appear:



Having looked at the various options, we decide to use the 'expanded desktop' format for our own enterprise. There are a number of ways to do this. One very simple method would be to simply delete all files that do not apply (i.e., files that define other desktop variants) from the default.htm and default.html directory, and then edit the remaining texts as needed. As in the original sample, the resulting page contains a link to the expanded desktop:

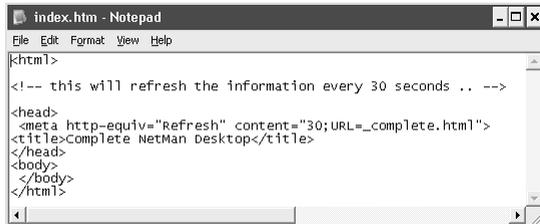


#### Note

*You will note that we left in the download link for the NetMan Web Client.*

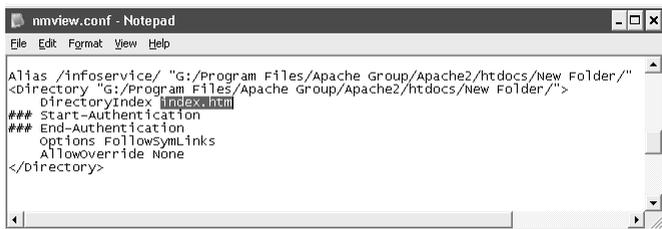
If we want to have the the expanded desktop opened as soon as the client browser arrives at our <server>/infoservice/ location, the procedure is a little more complicated. This is because HTML View cannot filter files for which no file name is specified. Here, however, is technique for getting around this complication:

First, create 2 files - we'll call them Index.htm and Index.html - with the following content:



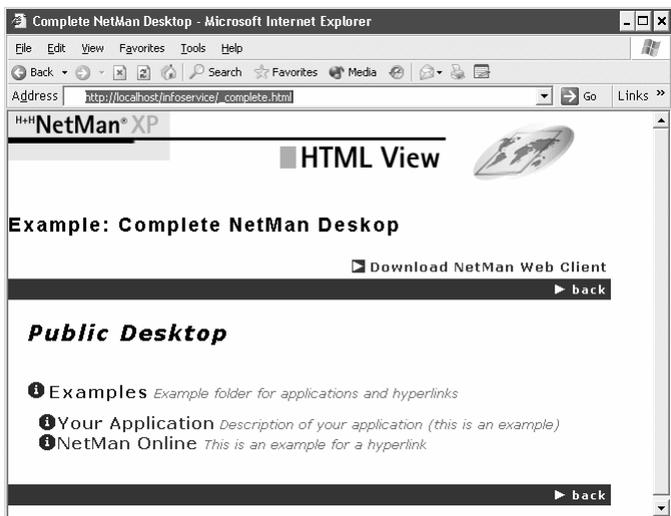
```
index.htm - Notepad
File Edit Format View Help
<html>
<!-- this will refresh the information every 30 seconds .. -->
<head>
<meta http-equiv="refresh" content="30;URL=_complete.html">
</head>
<title>Complete NetMan Desktop</title>
<body>
</body>
</html>
```

Then change the default file specified in NMView.conf:



```
nmview.conf - Notepad
File Edit Format View Help
Alias /infoservice/ "G:/Program Files/Apache Group/Apache2/htdocs/New Folder/"
<Directory "G:/Program Files/Apache Group/Apache2/htdocs/New Folder/">
  DirectoryIndex index.htm
### Start-Authentication
### End-Authentication
Options FollowSymLinks
AllowOverride None
</Directory>
```

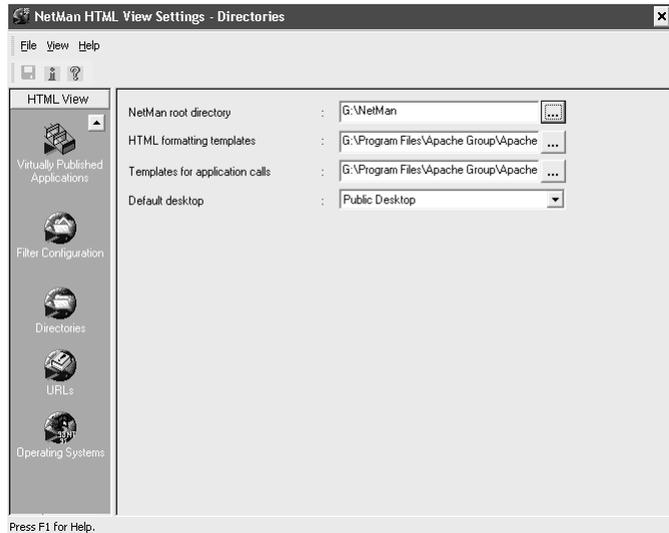
The URL now opens the expanded desktop:



The index.htm file calls the page containing the comment that loads the desired NetMan Desktop.

In the next example, we want to modify the pages generated by HTML View so that they fit in with the layout used in our own enterprise.

According to the default configuration of the DIRECTORIES category in the HTML View Settings, the templates for generating HTML pages are stored in <Apache-Installation> \HH\HTML\_View\Default.htf (see "Templates for Generating Desktop Structures"):

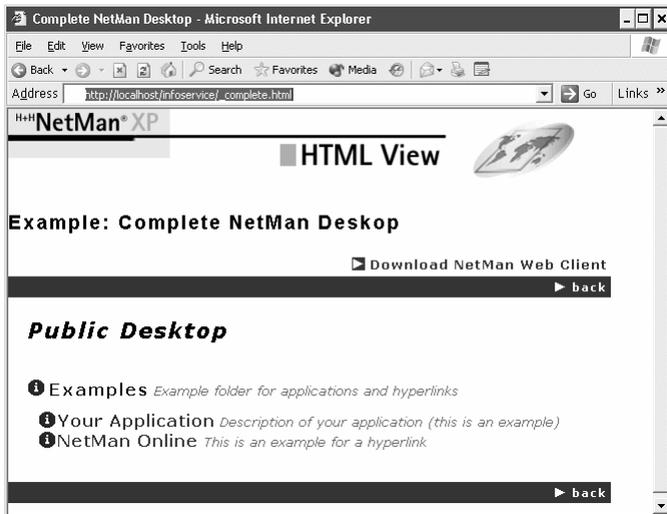


We do not want to make our modifications here, however, as a software update might overwrite this directory. As seen above, we could simply copy and rename the directory.

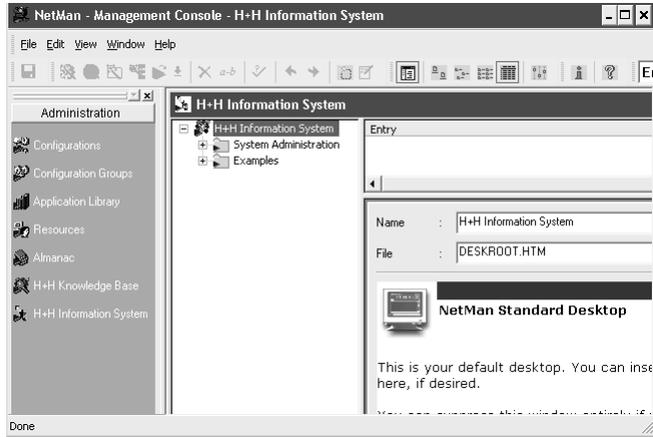
In this case, however, we decide to use the existing `..\MyFormat.HTF` directory (supplied for this purpose), so we enter this as our 'formatting templates' directory.



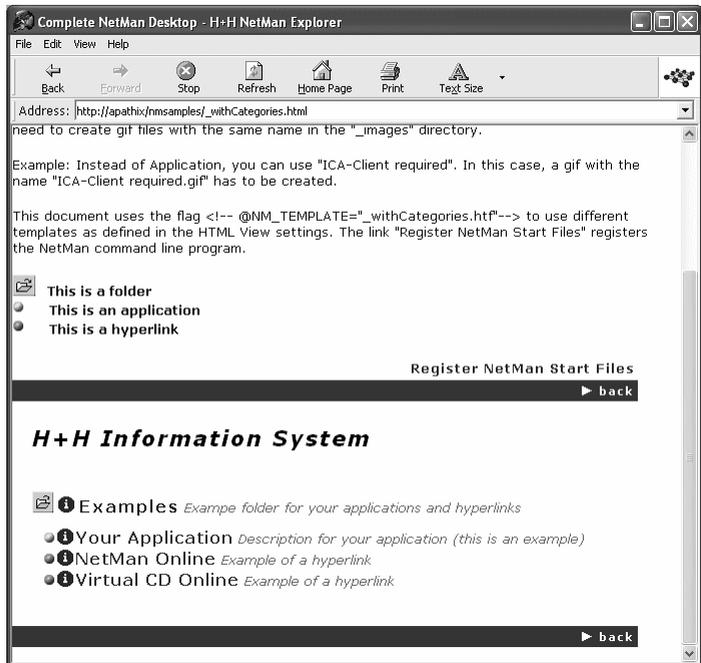
When the browser is pointed to our URL, the page is opened as usual:



Now we rename the Public Desktop in our Management Console:

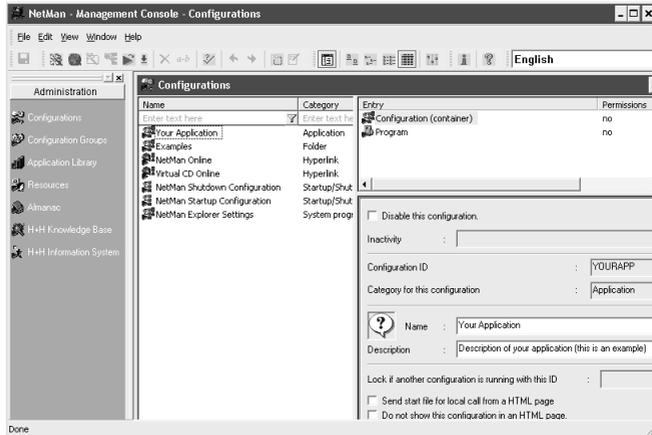


And the page looks like this:



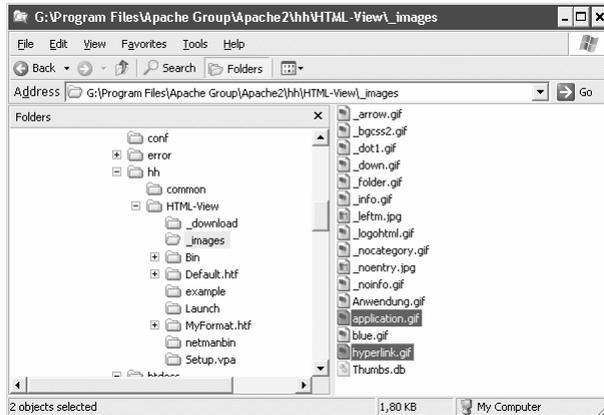
Looking over the information files, we decide that none of them are relevant for our purposes. We want to take out the info file links and add pictures that indicate whether a given link launches an application, or points to another HTML page.

The NetMan Management Console shows which category each NetMan configuration belongs to ("Folder", "Application" or "Hyperlink"):



We address the *Category* property of a NetMan configuration using the `@NM_CATEGORY` placeholder in our HTML page.

Then we copy the image files for the icons to `\_images` in our HTML View installation.



The next step is to modify the Hyperlink.htm and Link.htm templates. We use the files stored in `\MyFormat.htf`.

First we change the relevant tag in Hyperlink.html from...

```
link.htm - Notepad
File Edit Format View Help
<table border="0" cellspacing="0">
<tr>
<td>@NM_INFO_LINK</td>
<td width="100%">
<p ID="link">
<a href="@NM_LAUNCH"
onmouseover="start('@NM_PROMPT launched by H+H NetMan');return true;"
onmouseout="stop();">
@NM_PROMPT
</a>
<span ID="descr">@NM_DESCRIPTION</span></p>
</td>
</tr>
</table>
```

to...

```
link.htm - Notepad
File Edit Format View Help
<table border="0" cellspacing="0">
<tr>
<td>
&nbsp;
</td>
<td width="100%">
<p ID="link">
<a href="@NM_LAUNCH"
onmouseover="start('@NM_PROMPT launched by H+H NetMan');return true;"
onmouseout="stop();">
@NM_PROMPT
</a>
<span ID="descr">@NM_DESCRIPTION</span></p>
</td>
</tr>
```

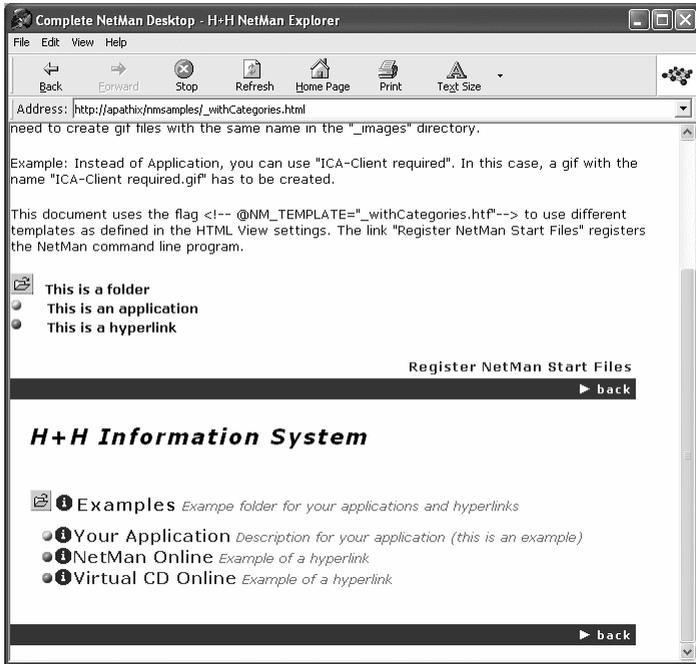
And in Weblink.htm, from...

```
weblink.htm - Notepad
File Edit Format View Help
<table border="0" cellspacing="0">
<tr>
<td>@NM_INFO_LINK</td>
<td width="100%">
<p ID="link">
<a href="javascript:newwindow('@NM_LAUNCH')"
onmouseover="start('@NM_PROMPT launched by H+H NetMan');return true;"
onmouseout="stop();">
@NM_PROMPT
</a>
<span ID="descr">@NM_DESCRIPTION</span></p>
</td>
</tr>
</table>
```

to:

```
weblink.htm - Notepad
File Edit Format View Help
<table border="0" cellspacing="0">
<tr>
<td>
&nbsp;</td>
<td width="100%">
<p ID="link">
<a href="javascript:newwindow('@NM_LAUNCH')"
onmouseover="start('@NM_PROMPT launched by H+H NetMan');return true;"
onmouseout="stop();">
@NM_PROMPT
</a>
<span ID="descr">@NM_DESCRIPTION</span></p>
</td>
</tr>
</table>
```

The updated page looks like this:



You can expand on these options, for example by linking the images with @NM\_Launch so that the application or link can be activated by clicking on the picture.

Many of our customers have expressed a wish to use configuration properties to indicate that a given link is available only to certain parties; for example, only when accessed from a workstation within the library or information department, or whether a certain range of information is provided for in-house use only or is freely available. You can do this by creating NetMan configuration categories to suit your organization and then using these category names, or image files with the same name, when designing your HTML page.

This example demonstrates how you can create customized NetMan HTML View pages by making just a few minor changes in existing files.

## NetMan HTML Files on a Different Server

The files used to generate your HTML View pages do not have to be stored on the Apache server that HTML View is installed on. For example, you might prefer to manage these files on a central Web server, with HTML View on another server. In such cases, you need to use the Apache *ProxyPass* command to give HTML View access to the HTML files it needs.

### Example:

If HTML View is installed on a server called “HH-SERVER”, and loads files for “http://www.kunde.de/netman” from http://netman-server.customer.com/nmsamples/, the command is as follows:

```
ProxyPass /nmsamples http: //www.customer.com/netman
```

To set up this function, the directory containing the HTML pages in question (in our example, ..\HTML-View\Example) are copied to a virtual directory called “NetMan” on the central Web server. This way, you can make use of HTML tags (see “How HTML View Works“) and permissions are evaluated as configured in HTML View.



### Note

*The Apache module ‘mod\_proxy’ must be loaded before you can use the ‘ProxyPass’ command.*



## Glossary





# Glossary

## A

### **Account**

Automatic access to an Internet resource through the optional -> HAN module.

### **Action**

An element of a NetMan -> configuration; an individual event or command.

### **Almanac**

An HTML document that provides an overview of NetMan directories, variables, log entry attributes and error messages.

### **Application**

An “application integrated in NetMan” is generally the same thing as a “NetMan configuration.”

### **Application call**

A “NetMan application call” refers to an application that is embedded in NetMan and called in conjunction with a number of user-definable settings, such as -> actions. In this usage it is equivalent to a NetMan -> configuration.

### **Application drive**

Drive designation under which the applications integrated in NetMan are installed; stored in the %NMAppDrive% variable.

### **Application library**

NetMan component for importing -> preconfigured applications into NetMan installations.

### **Application server**

See -> Terminal server.

### **Application server access control**

See -> Terminal server access control.

## B

### **Base Module**

The basic NetMan program installation, without any optional modules.

### **Browser Action**

A list of the browser types that have started applications on terminal servers through HTML View, together with the application start method used in each instance. The application start to be employed is derived from the -> browser rules.

**Browser rules**

Each browser rule defines the application start method for a specific browser type, determined from the browser userAgent property.

**C****Category**

A property of a NetMan -> configuration. You can group configurations into categories to ensure a clear overview of large numbers of NetMan configurations, and to present different categories with different graphics in HTML View.

**Citrix interface**

The Citrix WebClient variant used when a NetMan -> configuration is started from a Web page. The WebClient is provided by Citrix.

**Classroom control**

Central control element when NetMan is used as an -> educational interface. Classroom control lets you supervise the use of program operating elements by students and workgroups.

**Command line call**

This program can be used to execute NetMan -> configurations from the command line without starting the -> NetMan Client.

**Concurrent Users**

A licensing scheme that counts the number of simultaneous user sessions.

**Configuration**

The term "configuration" when used in conjunction with the NetMan software can refer to any of a number of the elements in the upper right-hand pane of the NetMan Client desktop; in this manual the term is generally equivalent to a NetMan -> application call. See also -> Configuration (container) and -> Configuration (hyperlink).

**Configuration (container)**

A user-definable logical unit containing a sequence of -> actions which are processed by the NetMan -> kernel.

**Configuration (hyperlink)**

Contains a configured -> hyperlink.

**D****Desktop**

The structured display of NetMan -> configurations in the NetMan Client or in an HTML page created by the NetMan HTML View or HTML Wizard.

**Desktop entry**

An element in the NetMan -> desktop.

## *E*

### **Educational interface**

Software specifically adapted for use in classroom settings. For optimum use of NetMan as the operating system interface for computer-supported education, in conjunction with the NetMan for Schools module.

### **Environment**

The NetMan environment can be customized through use of various NetMan variables, which are described in detail in the -> Almanac.

### **Error messages**

NetMan error messages are described in the -> Almanac.

## *F*

### **Folder**

Type of -> desktop entry.

### **Folder display**

The upper right-hand pane of the NetMan Client window; NetMan application calls can be launched from icons displayed in NetMan folders.

## *H*

### **HAN account**

See -> HAN; -> Account.

### **HAN module**

Hidden Automatic Navigator (HAN) is an optional NetMan module that lets you enable access to Internet resources for your users while hiding any separate logon required for a given site, as well as precluding an IP address check of the user's computer by the target host.

### **HTML Administration**

An optional NetMan module for the Enterprise Edition. Permits administrative access, status queries and NetMan system monitoring in HTML.

### **HTML template**

A template used by -> HTML View and the -> HTML Wizard to display NetMan -> configurations in HTML documents. The templates included with NetMan can be edited to suit your individual requirements.

### **HTML View**

An optional NetMan module. HTML View runs on an NT-based Apache Web server and can output NetMan desktops and NetMan -> application calls dynamically over HTTP as HTML documents. These HTML pages are provided with links to NetMan configurations in accordance with the user privileges valid for the client in question.

### **HTML View Settings**

Program for configuring NetMan HTML View.

### **HTML Wizard**

HTML Wizard is part of the -> Base module and lets you generate NetMan desktops as HTML documents, or insert links to NetMan configurations in HTML pages. Basically an alternative user interface, that can be used in place of the -> NetMan Client. Unlike HTML View, the HTML Wizard generate static documents; user privileges are not evaluated until a link has been selected.

### **Hyperlink**

URL; on-line access; HTML pages in general.

## **I**

### **Information display**

Lower right-hand pane of the NetMan Client window; presents information in HTML format on the -> applications integrated in NetMan.

### **Installation script**

A script created manually using the NetMan Installer Script Editor, or automatically by the Installer Script Wizard; integrated in NetMan -> configurations by the addition of an -> action.

### **Installer module**

An optional NetMan module for monitoring the client operating system while installing applications. Changes made locally can be distributed transparently to other workstations in the network using an -> installation script. A special service lets you have local components installed under a system account if the user in question does not have sufficient privileges to perform installation.

## **K**

### **Kernel**

The NetMan system process; controls the execution of container -> configurations, regardless of which NetMan interface is used (NetMan Client, HTML View or Wizard, command line call) to activate the configuration.

## **L**

### **Language module**

An optional NetMan module that lets you present the NetMan user interface (NetMan Client, HTML View or HTML Wizard) in different languages for different users. Also permits users to change the language during run time. The languages available are English, German and French.

### **Log attribute**

Event information recorded in addition to the standard entries (user, station, date and time). Log attributes are described in the NetMan -> Almanac.

## *M*

### **MetaFrame**

An add-on from Citrix for the Microsoft Terminal Server. Enables, for example, access to MetaFrame servers from non-Windows platforms such as Macintosh or Unix.

## *N*

### **Named Sites**

A licensing scheme that counts the number of workstations registered in the NetMan system. Each station is registered automatically when it logs on to NetMan. If a license is unused for a period of 40 days, it is released and can be used by another station.

### **NetMan Client**

The NetMan user interface; NetMan -> configurations are displayed here on the NetMan -> desktop.

### **NetMan Explorer**

A Web browser that runs on the MS IE 5 and is integrated in NetMan. Lets you configure user controls in the browser and define “permitted” and “denied” URLs.

### **NetMan for Schools**

An optional NetMan module for use in educational networks. The -> classroom control feature allows flexible adaptation of the user controls in a station profile to meet the requirements in a given educational setting, using a central computer. The User Account Wizard supports maintenance of student and teacher accounts.

### **NetMan Service**

A central NT service that manages information on users, stations, licenses and NetMan -> application calls.

### **NM file**

A file with the two-letter extension “.NM”; used to start NetMan -> configurations from HTML pages.

## *P*

### **Preconfigured applications**

Ready-to-use NetMan -> configurations provided by H+H for import into your NetMan installation, using the -> application library.

### **ProGuard module**

An optional NetMan module that enables process-level control of your network computers.

### **Public desktop**

The default name of the NetMan -> desktop.

**Published application**

Created with Citrix software to access a session on a -> terminal server. This “published application” is required by the -> HTML Wizard and -> HTML View for making a connection.

**R****Root directory**

The NetMan root directory is the target directory entered for NetMan installation. This directory is addressed by NetMan through the %NMHome% variable.

**S****Secure Distribution of ICA Files**

NetMan extension for Citrix servers: ICA files are only valid for a short period and cannot be re-used. This ensures that access is possible only over your own Web server.

**Selection bar**

A pane in some NetMan program windows that offers a number of elements for processing or navigation; located on the left-hand side of the program window.

**Semaphore files**

A mode of -> station monitoring: with this mode selected, files are created that are opened exclusively when a station logs on to NetMan; thus you can tell which stations are using NetMan by the files that are open (see also -> NetMan Service).

**Settings**

The program for configuring basic NetMan parameters.

**Settings dialog**

See -> Settings

**Shutdown configuration**

A -> configuration specified in the NetMan -> Settings; processed when the NetMan software is shut down.

**SnapShot**

A “picture” of the current system state on a workstation, created by the NetMan -> Installer.

**Start file**

A file with the two-letter extension NM; when this file type is used to start a NetMan -> configuration from HTML View or the HTML Wizard, the configuration runs on the client machine rather than a terminal server.

**Startup configuration**

A -> configuration specified in the NetMan -> Settings to be processed when NetMan is started.

### **Station database**

NetMan database in which every station that starts NetMan is automatically registered under the given NetMan -> station ID.

### **Station ID**

A unique designation that identifies a workstation; registered in the -> station database.

### **Station monitoring**

A mechanism used by NetMan to keep track of a station working with the NetMan system.

### **Station profile**

A set of defined preferences; you can assign the same profile to multiple stations, but each station can be assigned only one profile.

### **Structure display**

Left-hand pane of the NetMan Client window; shows the organization of the NetMan folders.

## *T*

### **Terminal server**

The Microsoft Terminal Server provides server sessions for remote Windows clients. Applications launched by the client run on the server and do not require any specific components on the client computer.

### **Terminal server access control**

NetMan extension for terminal servers: Access can be permitted or denied based on IP address or IP address range. You can assign a NetMan -> user ID to an IP address or to a specified range of addresses.

### **Terminal Server module**

An optional NetMan module for using NetMan in terminal server environments.

### **Timeout**

A program that monitors applications started by NetMan and ends them if no input is detected for a defined period of time.

## *U*

### **User database**

NetMan database in which every user that starts NetMan is automatically registered under the given NetMan -> user ID.

### **User ID**

A unique designation that identifies a user; registered in the -> user database.

**User group**

You can group your NetMan users; for example, to simplify the assignment of permissions.

**User profile**

A set of defined preferences. You can assign the same profile to multiple users, but each user can be assigned only one profile.

**V****Variables**

NetMan supports both system and local environment variables. NetMan variables are described in the NetMan -> Almanac.

**Virtually published applications**

A NetMan extension for Citrix servers: A single published program functions as a “Trojan horse” for all NetMan -> configurations.

**VPA Service**

This service creates anonymous NetMan user accounts as needed; optimizes access from HTML documents to the Microsoft Terminal server.

**W****Windows Script Host**

Provided by Microsoft for extending the Windows operating system. The script host enables access to operating system functions over VBScript and JScript. NetMan provides interfaces to its system functions for the script host, which can be used by VBScript and JScript programmers to expand and adapt NetMan features.

**Working directory**

The working directory for NetMan is the \bin subdirectory in the NetMan -> root directory.