Help for UNIX System Administrators

New Lay-flat Binding

Managing NFS *and* NIS

A NUTSHELL HANDBOOK

Hal Stern

O'Reilly & Associates, Inc.

Managing NFS and NIS
by Hal Stern

Editor: Mike Loukides

Printing History:

June 1991: First Edition.

April 1992: Minor corrections.

This book is printed on acid-free paper with 50% recycled content, 10-15% post-consumer waste. O'Reilly & Associates is committed to using paper with the highest recycled content available consistent with high quality.

ISBN: 0-937175-75-7 [6/94]

Preface

Ten years ago, most computer centers had a few large computers shared by several hundred users. The "computing environment" was usually a room containing dozens of terminals. All users worked in the same place, with one set of disks, one user account information file, and one view of all resources. Local area networks have made terminal rooms much less common. Today, a "computing environment" almost always refers to distributed computing, where users have workstations on their desks and shared resources are provided by special-purpose systems such as file, compute, and print servers. Each workstation requires redundant configuration files, including user information, network host addresses, and local and shared remote filesystem information.

A mechanism to provide consistent access to all files and configuration information ensures that all users have access to the "right" machines, and that once they have logged in they will see a set of files that is both familiar and complete. This consistency must be provided in a way that is transparent to the users, that is, a user should not know that a filesystem is located on a remote fileserver. The transparent view of resources must be consistent across all machines and also consistent with the way things work in a non-networked environment. In a networked computing environment, it's usually up to the system administrator to

manage both the machines on the network (including centralized servers) as well as the network itself. Managing the network means ensuring that the network is transparent to users rather than an impediment to their work.

The Network File System (NFS) and the Network Information Service (NIS)* provide mechanisms for solving "consistent and transparent" access problems. The NFS and NIS protocols were developed by Sun Microsystems and are now licensed to about 300 vendors and universities. NIS centralizes commonly replicated configuration files, such as the password file, on a single host. It eliminates duplicate copies of user and system information and allows the system administrator to make changes from one place. NFS makes remote filesystems appear to be local, as if they were on disks attached to the local host. With NFS, all machines can share a single set of files, eliminating duplicate copies of files on different machines in the network. Using NFS and NIS together greatly simplifies the management of various combinations of machines, users, and filesystems.

NFS provides network and filesystem transparency because it hides the actual, physical location of the filesystem. A user's files could be on a local disk, on a shared disk on a fileserver, or even on a machine located across a wide-area network. As a user, you're most content when you see the same files on all machines. Just having the files available, though, doesn't mean that you can access them if your user information isn't correct. Missing or inconsistent user and group information will break UNIX file permission checking. This is where NIS complements NFS, by adding consistency to the information used to build and describe the shared filesystems. A user can sit down in front of any workstation in his or her group that is running NIS and be reasonably assured that he or she can log in, find his or her home directory, and access tools such as compilers, window systems, and publishing packages. In addition to making life easier for the users, NFS and NIS simplify the tasks of system administrators, by centralizing the management of configuration information and of disk resources.

NFS can be used to create very complex filesystems, taking components from many different servers on the network. It is possible to overwhelm users by providing "everything everywhere," so simplicity should rule network design. Just as a database programmer constructs views of a database to present only the relevant fields to an application, the user community should see a logical collection of files, user account information, and system services from each viewpoint in the computing environment. Simplicity often satisfies the largest number of users, and it makes the system administrator's job easier.

*NIS was formerly called the "Yellow Pages". While many commands and directory names retain the *yp* prefix, the formal name of the set of services has been changed to avoid conflicting with registered trademarks.

Who This Book is For

This book is of interest to system administrators and network managers who are installing or planning new NFS and NIS networks, or debugging and tuning existing networks and servers. It is also aimed at the network user who will be building tools and applications that use NIS services, or who are interested in the mechanics that hold the network together.

We'll assume that you are familiar with the basics of UNIX system administration and TCP/IP networking. Terms that are commonly misused or particular to a discussion will be defined as needed. Where appropriate, an explanation of a low-level phenomenon such as Ethernet congestion will be provided if it is important to a more general discussion such as NFS performance on a congested network. Models for these phenomena will be drawn from everyday examples rather than their more rigorous mathematical and statistical roots.

This book focuses on the way NFS and NIS work, and how to use them to solve common problems in a distributed computing environment. Use this book in conjunction with your vendor's documentation, since utilities and their options will vary by implementation. This book explains what the configuration files and utilities do, and how their options affect performance and system administration issues. By walking through the steps comprising a complex operation, or by detailing each step in the debugging process, we hope to shed light on techniques for effective management of distributed computing environments. There are very few absolute constraints or thresholds that are universally applicable, so we refrain from stating them. This book should help you to determine the fair utilization and performance constraints for your network.

Versions

This book is based on the SunOS 4.1 implementations of NFS and NIS. When used without a version number, "SunOS" refers to the SunOS 4.1 UNIX operating system and its derivatives. In this sense, "SunOS" refers to the operating system for SPARC-based machines. NFS- and NIS-related tools did not change significantly between SunOS 4.0 and SunOS 4.1, but functions that are specific to SunOS 4.1 will be noted. Some of the NFS discussions apply to SunOS 3.5 and earlier releases, but SunOS 4.0 introduced many new features: diskless client support using NFS, new exporting rules, the automounter, and a new virtual memory management system.

The discussion of PC/NFS is based on PC/NFS 3.5. Any feature that is particular to PC/NFS 3.5 or behaves differently in earlier releases is noted.

Organization

This book is divided into two sections. The first nine chapters contain explanations of the implementation and operation of NFS and NIS. Chapters 10 through 14 cover advanced administrative and debugging techniques, performance analysis, and tuning. Building on the introductory material, the second section of the book delves into low-level details such as the effects of network partitioning hardware and the various steps in a remote procedure call. The material in this section is directly applicable to the ongoing maintenance and debugging of a network.

Here's the chapter-by-chapter breakdown:

- Chapter 1, *Networking Fundamentals*, provides an introduction to the underlying network protocols and services used by NFS and NIS.
- Chapter 2, *Network Information Service Operation*, discusses the architecture of NIS and its operation on both NIS servers and NIS clients. The focus is on how to set up NIS and its implementation features that affect network planning and initial configuration.
- Chapter 3, *System Management Using the Network Information Service*, discusses operational aspects of NIS that are important to network administrators. This chapter explores common NIS administration techniques, including map management, setting up multiple NIS domains, and using NIS with domain name services.
- Chapter 4, *Building Applications with NIS*, shows how to use NIS as a distributed database system, creating your own NIS maps to manage a telephone list and to grant restricted access to specific services. It contains several examples ranging from shell scripts to C programs that use the NIS client programming library.
- Chapter 5, *System Administration and the Network File System*, covers basic NFS operations such as mounting and exporting filesystems, and using symbolic links within NFS.
- Chapter 6, *Network File System Design and Operation*, explains the architecture of NFS and the underlying virtual file system. It also discusses implementation details that affect performance, such as file attribute and data caching.

- Chapter 7, *Diskless Clients*, is all about diskless clients. It also presents debugging techniques for clients that fail to boot successfully.
- Chapter 8, *Network Security*, explores network security. Issues such as restricting access to hosts and filesystems form the basis for this chapter. We'll also go into how to make NFS more secure, including a discussion of setting up Sun's Secure NFS.
- Chapter 9, *Centralizing Mail Services with NFS and NIS*, covers the impact of NFS on electronic mail services and suggests ways to consolidate mail delivery functions in a network. Tips for handling user aliases, wide-area mailing lists and mail forwarding are included.
- Chapter 10, *Diagnostic and Administrative Tools*, describes the administrative and diagnostic tools that are applied to the network and its systems as a whole. This chapter concentrates on the network and on interactions between hosts on the network, instead of the per-machine issues presented in earlier chapters. Tools and techniques are described for analyzing each layer in the protocol stack, from the Ethernet to the NFS and NIS applications.
- Chapter 11, *Debugging Network Problems*, is a collection of debugging stories, and shows how the tools described in the previous chapter are applied in some real-world situations.
- Chapter 12, *Performance Analysis and Tuning*, covers performance tuning and analysis of machines and the network. It discusses network partitioning using bridges and routers, with a focus on NFS performance optimization. Server parameter tuning, NFS mount parameter adjustments, and client-side issues are explored in detail in this chapter as well.
- Chapter 13, *The Automounter*, discusses the automounter, a powerful but sometimes confusing tool that integrates NIS administrative techniques and NFS filesystem management.
- Chapter 14, *PC/NFS*, covers PC/NFS, a client-side implementation of NFS for DOS machines.
- Appendix A, *Transmission Line Theory*, contains a low-level description of transmission line theory, the basis for Ethernet termination and many other physical constraints of the Ethernet.
- Appendix B, *IP Packet Routing*, explains how IP packets are forwarded to other networks. It is additional background information for discussions of performance and network configuration.

- Appendix C, *NFS Problem Diagnosis*, summarizes NFS problem diagnosis using the NFS statistics utility and the error messages printed by clients experiencing NFS failures.
- Appendix D, *NFS Benchmarks*, contains information on NFS traffic generation for benchmarking.

Conventions Used in This Book

Font and format conventions for UNIX commands, utilities, and system calls are:

- Excerpts from script or configuration files will be shown in a fixed-width font:

```
192.9.200.1    bitatron
```

- Sample interactive sessions, showing command line input and corresponding output, will be shown in a fixed-width font, with user-supplied input in bold:

```
% ls
foo bar
```

If the command can be typed by any user, the percent sign (%) will be shown as the prompt. If the command must be executed by the superuser, then the pound sign (#) will be shown as the prompt:

```
# /usr/etc/ypinint -m
```

- Inside of an excerpt from a script, configuration file, or other ASCII file, the pound sign will be used to indicate the beginning of a comment:

```
#
#   Hal's machine
192.9.200.1    bitatron
```

- UNIX commands, command lines, and mount and export options are printed in bold text when they appear in the body of a paragraph. For example, the **ls** command lists files in a directory, and the **timeo** option changes NFS client behavior.
- Hostnames are printed in italics. For example, server *wahoo* contains home directories.
- Filenames are printed in bold, for example, the **/etc/passwd** file.

- NIS map names are printed in italics. The *passwd* map is used with the **/etc/passwd** file.
- System and library calls are printed in italics, with parentheses to indicate that they are C routines. For example, the *gethostent( )* library call locates a hostname in an NIS map.
- Control characters will be shown with a CTRL prefix, for example, CTRL-Z.

Acknowledgments

Writing about NFS and NIS was not easy, because their implementations changed quite rapidly while I was writing. NFS and NIS were introduced in 1985, which makes them old in the technology timescale. While the protocols have not changed, we have seen many new NFS and NIS features such as NFS write accelerators, caching techniques, and specialized NFS server architectures. There are now almost 300 implementations of NFS, and some campus networks have several hundred nodes sharing files and configuration information.

This book would not have been completed without the help of many people. I'd like to thank Brent Callaghan, Chuck Kollars, Neal Nuckolls, and Janice McLaughlin (all of Sun Microsystems); Kevin Sheehan (Kalli Consulting); Vicki Lewolt Schulman (Auspex Systems); and Dave Hitz (H&L Software) for their never ending stream of answers to questions about issues large and small. Bill Melohn (Sun) provided the foundation for the discussion of computer viruses. The discussion of NFS performance tuning and network configuration is based on work done with Peter Galvin and Rick Sabourin at Brown University. Several of the examples of NIS and NFS configuration were taken from a system administrator's guide to NFS and NIS written by Mike Loukides for Multiflow Computer Company.

The finished manuscript was reviewed by: Chuck Kollars, Mike Marotta, Ed Milstein, and Brent Callaghan (Sun); Dave Hitz (H&L Software); Larry Rogers (Princeton University); Vicki Lewolt Schulman (Auspex); Simson Garfinkel (NeXTWorld); and Mike Loukides and Tim O'Reilly (ORA). This book has benefited in many ways from their insights, comments and corrections. The production group of O'Reilly & Associates also deserves my gratitude for applying the

finishing touches to this book. I owe a tremendous thanks to Mike Loukides of O'Reilly and Associates who helped undo four years of liberal arts education and associated writing habits. It is much to Mike's credit that this book does not read like a treatise on Dostoevsky's *Crime and Punishment*†.

† I think I will cause my freshman composition lecturer pain equal to the credit given to Mike, since she assured me that reading and writing about *Crime and Punishment* would prepare me for writing assignments the rest of my life. I have yet to see how, except possibly when I was exploring performance issues.

Table of Contents